

**SUPRESIÓN DE  
INTERFERENCIAS MEDIANTE  
TRANSFORMADAS WAVELET  
EN SISTEMAS DE  
COMUNICACIÓN CON  
ESPECTRO ENSANCHADO POR  
SALTO DE FRECUENCIA.**

Juan José Pérez Solano

Dpto. de Informática  
Universitat de València



Supresión de Interferencias Mediante Transformadas Wavelet  
en Sistemas de Comunicación con Espectro Ensanchado por  
Salto de Frecuencia.

MEMORIA PARA OPTAR AL GRADO DE DOCTOR  
PRESENTADA AL DPTO. DE INFORMÁTICA  
FACULTAD DE CIENCIAS FÍSICAS  
UNIVERSITAT DE VALÈNCIA

Presentada por:  
D. Juan José Pérez Solano

Dirigida por:  
Dr. Santiago Felici i Castell  
Dr. Miguel Ángel Rodríguez Hernández

Abril 2002



A mis padres y hermanos.  
A Sara.



# Índice General

<b>I</b>	<b>Punto de partida y objetivos</b>	<b>1</b>
<b>1</b>	<b>Introducción</b>	<b>3</b>
1.1	Marco de la investigación realizada . . . . .	3
1.2	Aplicación de la investigación realizada . . . . .	5
1.3	Objetivos de la tesis . . . . .	6
1.4	Breve descripción de los contenidos . . . . .	7
<b>II</b>	<b>Revisión de la situación actual: estado del arte</b>	<b>11</b>
<b>2</b>	<b>Representación de señales en el tiempo, frecuencia y escala</b>	<b>15</b>
2.1	Introducción . . . . .	15
2.2	Transformada Wavelet . . . . .	16
2.2.1	Discretización de los parámetros tiempo-escala . . . . .	18
2.2.2	Wavelet frames . . . . .	19
2.3	Multiresolución . . . . .	21
2.3.1	Definición formal . . . . .	21
2.3.2	Subespacios complementarios ortogonales . . . . .	24
2.3.3	Cálculo de los coeficientes Wavelet mediante recursión: Transformada Wavelet Discreta . . . . .	26
2.4	Bancos de filtros . . . . .	27
2.4.1	Reconstrucción perfecta . . . . .	29

2.4.2	Ejemplo de diseño de filtros: caso filtros de Daubechies con 4 coeficientes . . . . .	30
2.5	Transformadas Wavelet con descomposición irregular: Paquetes de Wavelet . . . . .	31
2.6	Transformadas Wavelet Sobremuestreadas . . . . .	34
2.7	Esquemas eficientes para el cálculo de la Transformada Wavelet . . . .	38
2.7.1	Cálculo mediante matriz Polifásica . . . . .	39
2.7.2	Cálculo mediante estructura <i>Lattice</i> . . . . .	40
2.8	Conclusiones . . . . .	43
<b>3</b>	<b>Comunicaciones con modulaciones de Espectro Ensanchado</b>	<b>45</b>
3.1	Introducción . . . . .	45
3.2	Señales moduladas con Espectro Ensanchado . . . . .	46
3.2.1	Espectro Ensanchado en Secuencia Directa . . . . .	47
3.2.2	Espectro Ensanchado con Saltos de Frecuencia . . . . .	48
3.2.3	Otros tipos de modulaciones de Espectro Ensanchado . . . . .	50
3.3	Estudio del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia no-coherente . . . . .	51
3.3.1	Recepción en canales con ruido aditivo blanco gaussiano . . . .	52
3.3.2	Recepción en canales con ruido aditivo gaussiano de banda estrecha	54
3.3.3	Recepción en canales con ruido aditivo senoidal continuo . . . .	55
3.4	Supresión de interferencias en modulaciones de Espectro Ensanchado .	61
3.4.1	Supresión de interferencias en el dominio temporal . . . . .	62
3.4.2	Supresión de interferencias en dominio de la transformada . . . .	66
3.5	Conclusiones . . . . .	71

### **III Análisis, diseño e implementación del receptor con supresión de interferencias para modulaciones de Espectro En-**

sanchado con Saltos de Frecuencia propuesto. 75

**4 Receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias mediante Transformadas Wavelet Sobremuestreadas 79**

4.1	Introducción . . . . .	79
4.2	Diseño del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias . . . . .	80
4.3	Diseño del filtro de supresión de interferencias . . . . .	86
4.3.1	Hipótesis iniciales . . . . .	86
4.3.2	Análisis del algoritmo . . . . .	87
4.3.3	Algoritmo basado en Transformadas Wavelet Sobremuestreadas	89
4.4	Ejemplo de aplicación del algoritmo propuesto: validación . . . . .	101
4.5	Conclusiones . . . . .	102

**5 Implementación del algoritmo de supresión de interferencias mediante estructuras *Lattice* sobre Procesadores Digitales de Señal 105**

5.1	Introducción . . . . .	105
5.2	Estructura <i>Lattice</i> para Transformadas Wavelet Sobremuestreadas . . .	107
5.2.1	Comparación en coste computacional de las estructuras en forma directa y <i>Lattice</i> . . . . .	110
5.3	Estructura <i>Lattice</i> para la implementación del algoritmo de supresión de interferencias propuesto . . . . .	112
5.4	Diseño del algoritmo de supresión de interferencias en Procesador Digital de Señal . . . . .	114
5.4.1	Implementación de la estructura <i>Lattice</i> en Procesador Digital de Señal . . . . .	115
5.4.2	Implementación del resto de subrutinas que componen el algoritmo de supresión . . . . .	120
5.4.3	Análisis del coste computacional del algoritmo de supresión . . .	121
5.5	Conclusiones . . . . .	122

**IV Resultados 125**

**6 Evaluación de las prestaciones del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias 129**

6.1 Introducción . . . . . 129

6.2 Resultados mediante simulación . . . . . 131

6.2.1 Estructura del simulador . . . . . 132

6.2.2 Cálculo de los intervalos y niveles de confianza . . . . . 134

6.2.3 Experimento 1: variación de la frecuencia de la interferencia . . 136

6.2.4 Experimento 2: variación del ruido aditivo blanco gaussiano . . 138

6.2.5 Experimento 3: interferencia multitono . . . . . 139

6.2.6 Experimento 4: comparación con FFH para interferencia multi-  
tono en el caso peor . . . . . 141

6.2.7 Experimento 5: ruido gaussiano de banda estrecha . . . . . 142

6.2.8 Experimento 6: variación del tamaño del bloque y del nivel ini-  
cial del algoritmo . . . . . 143

6.2.9 Experimento 7: variación del número de niveles y el incremento  
del desplazamiento en frecuencia . . . . . 145

6.3 Resultados sobre prototipo . . . . . 147

6.3.1 Simulación con variación del número de niveles y del incremento  
del desplazamiento en frecuencia, sobre el DSP ADSP2065L . . 148

6.4 Conclusiones . . . . . 152

**7 Aplicación del algoritmo de supresión propuesto a la transmisión de datos por el cable de potencia 155**

7.1 Introducción . . . . . 155

7.2 Análisis del canal de comunicación . . . . . 156

7.3 Construcción del emisor y el receptor con modulación FH-SS . . . . . 160

7.3.1 Características de la modulación . . . . . 160

7.3.2 Características del emisor . . . . . 161

7.3.3	Características del receptor . . . . .	162
7.4	Resultados de campo . . . . .	167
7.5	Conclusiones . . . . .	169
<b>V</b>	<b>Conclusiones y trabajo futuro</b>	<b>171</b>
<b>8</b>	<b>Conclusiones y trabajo futuro</b>	<b>173</b>
8.1	Introducción . . . . .	173
8.2	Aportaciones . . . . .	173
8.3	Conclusiones . . . . .	175
8.4	Trabajo futuro . . . . .	177
<b>VI</b>	<b>Anexos</b>	<b>181</b>
<b>A</b>	<b>Arquitectura de control PRIFEM</b>	<b>183</b>
A.1	Introducción a la arquitectura del sistema . . . . .	183
A.1.1	Arquitectura PRIFEM . . . . .	183
A.2	Elementos URA y UCE . . . . .	187
A.2.1	Modulaciones y tramas . . . . .	188
A.2.2	Implementación de la URA . . . . .	192
A.2.3	Implementación de la UCE . . . . .	195
A.3	Estructura de la instalación piloto . . . . .	197
A.3.1	Elementos remotos . . . . .	198
A.3.2	Elementos del cabezal . . . . .	199
<b>B</b>	<b>Características básicas del procesador digital de señal</b>	<b>205</b>
B.1	El DSP <i>ADSP2065L</i> . . . . .	205
<b>C</b>	<b>Programa para Matlab</b>	<b>209</b>

C.1	Programa del simulador FH-SS, sobre Matlab . . . . .	209
C.1.1	Fichero simuladorfh.m . . . . .	209
C.1.2	Algoritmo de filtrado algofil.C . . . . .	211
<b>D</b>	<b>Programa para el DSP ADSP21065L</b>	<b>217</b>
D.1	Funciones en ensamblador para el cálculo de la estructura <i>Lattice</i> . . .	217
D.1.1	Análisis . . . . .	217
D.1.2	Síntesis . . . . .	219
<b>VII</b>	<b>Bibliografía</b>	<b>223</b>

# Índice de Figuras

2.1	<i>División del espacio tiempo/frecuencia para STFT y WT. En el caso de la WT se supone una descomposición diádica. . . . .</i>	18
2.2	<i>Descomposición espectral de subespacios para las técnicas de multiresolución, aplicando descomposición diádica. <math>V_i</math> subespacio de residuos, <math>W_i</math> subespacio de detalles . . . . .</i>	24
2.3	<i>Descomposición con banco de filtros para señales de una dimensión, donde <math>X</math> es la señal de entrada, <math>h_0(n)</math> y <math>h_1(n)</math> son los filtros paso-bajo de análisis y síntesis, <math>g_0(n)</math> y <math>g_1(n)</math> son los filtros paso-alto de análisis y síntesis y <math>\hat{X}</math> es la señal reconstruida. . . . .</i>	28
2.4	<i>Espectro del filtro paso-bajo <math>H(w)</math> y paso-alto <math>G(w)</math> para multiresolución.</i>	28
2.5	<i>Esquema de descomposición de la WPT hasta el tercer nivel de resolución. En la figura <math>h_0</math> es el filtro paso-bajo y <math>g_0</math> es el paso-alto, <math>j</math> es el nivel de resolución y <math>m</math> el índice de los subespacios dentro de un nivel. .</i>	33
2.6	<i>Espectro de los filtros para los cuatro primeros niveles, utilizando los filtros de Daubechies 4 definidos en la sección 2.4.2. De izquierda a derecha y de arriba a abajo se muestra el primer nivel, <math>j = 1</math>, con los filtros <math>h_{0,1}(n)</math> y <math>g_{0,1}(n)</math>, el segundo nivel, <math>j = 2</math>, con <math>h_{0,2}(n)</math> y <math>g_{0,2}(n)</math>, el tercer nivel, <math>j = 3</math>, con <math>h_{0,3}(n)</math> y <math>g_{0,4}(n)</math> y el cuarto nivel, <math>j = 4</math>, con <math>h_{0,4}(n)</math> y <math>g_{0,4}(n)</math> . . . . .</i>	36
2.7	<i>Esquema de subbandas obtenidas en UWPT, con niveles <math>j \in [0, \dots, 4]</math>. .</i>	37
2.8	<i>Espectro de la subbanda <math>W_{4,0}</math>, obtenida tras aplicar <math>h_{0,1}(n), h_{0,2}(n), h_{0,3}(n)</math> y <math>h_{0,4}(n)</math>. . . . .</i>	38
2.9	<i>banco de filtros con implementación Polifásica. Donde <math>X</math> es el vector de entrada y <math>\hat{X}</math> es el vector reconstruido, <math>H_p(z)</math> es la matriz Polifásica de análisis y <math>F_p(z)</math> es la matriz Polifásica de síntesis. . . . .</i>	40

2.10	<i>Estructura Lattice para un banco de filtros, donde <math>K_0, \dots, K_{N-1}</math> son los coeficientes que definen la estructura, <math>P</math> y <math>Q</math> son las salidas de cada una de las secciones y <math>z^{-1}</math> representa los elementos de retraso. . . . .</i>	41
2.11	<i>Estructura Lattice para un banco de filtros ortogonal con reconstrucción perfecta, donde <math>K_0, \dots, K_{N-1}</math> son los coeficientes que definen la estructura, <math>P</math> y <math>Q</math> son las salidas de cada una de las secciones y <math>z^{-1}</math> representa los elementos de retraso. Además, <math>E(z^2)</math> es la matriz definida como combinación de las secciones de la estructura, contenidas en el cuadrado definido en la figura. . . . .</i>	42
2.12	<i>Esquema del banco de filtros obtenido en la implementación Lattice, definido mediante la matriz de análisis <math>E(z)</math> y la de síntesis <math>R(z)</math>. . . .</i>	43
3.1	<i>Esquema del sistema de comunicación utilizando técnicas SS. . . . .</i>	47
3.2	<i>Ejemplo de aplicación mediante multiplicaciones, de la secuencia PN sobre los bits transmitidos en el alfabeto <math>\{+1, -1\}</math>. La primera línea representa los bits transmitidos, la segunda la secuencia PN y la tercera la secuencia de chips. . . . .</i>	48
3.3	<i>Espectro de la señal DS-SS expandida, junto con la señal PSK original. Donde <math>2R</math> y <math>2R_c</math> representan la amplitud del lóbulo principal de la señal antes y después de la expansión. . . . .</i>	49
3.4	<i>Ejemplo de distribución de frecuencias para FH-SS en función del tiempo, los saltos de frecuencia entre chips están dictados por el código PN. El valor <math>T_c</math> corresponde con el periodo de un chip generado por la secuencia PN. . . . .</i>	49
3.5	<i>Esquema del receptor BFSK no-coherente. La señal <math>r(t)</math> se correlaciona en cuadratura con las frecuencias <math>f_0</math> y <math>f_1</math>, posteriormente se aplica un filtro paso-bajo y se combina la energía de ambos correladores de forma no-coherente, para eliminar la dependencia con la fase <math>\phi_s</math>. La variable de decisión resultante con mayor energía, <math> U_1 ^2</math> ó <math> U_2 ^2</math>, indica el bit transmitido. . . . .</i>	52
3.6	<i>Probabilidad de error para BFSK con interferencia en tono simple, donde <math>\alpha</math> es la relación entre la potencia de la señal y la interferencia. . . .</i>	58
3.7	<i>Probabilidad de error para el receptor FFH-SS, no-coherente con combinación lineal de los chips, con interferencia multitono en el caso peor para distintos números de chips <math>L</math>. Esta gráfica ha sido obtenida de la referencia [81]. . . . .</i>	61

3.8	<i>Probabilidad de error para el receptor FFH-SS, no-coherente con limitador de amplitud, con interferencia multitono en el caso peor para distintos números de chips <math>L</math>. Esta gráfica ha sido obtenida de la referencia [83]. . . . .</i>	62
3.9	<i>Esquema del receptor DS con filtro adaptativo para la eliminación de interferencias de banda estrecha. Dicho receptor está formado por un correlador centrado en la frecuencia de la portadora <math>\omega_0</math>, un integrador sobre el periodo de un chip <math>T_c</math>, el filtro adaptativo y por último el correlador y el sumador para comprimir la señal. . . . .</i>	63
3.10	<i>Esquema del receptor FFH-SS con filtrado adaptativo para eliminación de interferencias de banda estrecha. El receptor contiene dos correladores en cuadratura sintonizados a la frecuencia de la portadora, <math>\omega_0</math>, a continuación se sitúa el filtro adaptativo y por último el receptor no-coherente, que combina los <math>L</math> chips en los que se divide un bit en la modulación FFH-SS. . . . .</i>	65
3.11	<i>Proceso de supresión de interferencias con transformada. Los puntos que superan un cierto umbral de energía en el dominio transformado son eliminados. En la figura <math>R_c = 1/T_c</math> representa el periodo de un chip. . . . .</i>	67
3.12	<i>Esquema del receptor DS con módulo de supresión basado en transformada. El bloque que realiza la transformada, a la salida del correlador y el integrador con periodo <math>T_c</math>, se representa por <math>T</math>. A continuación, el bloque transformado se multiplica por el vector <math>\lambda</math> para eliminar los coeficientes cuya energía supere un cierto umbral. Por último, se realiza la transformada inversa <math>T^{-1}</math> y se comprime la señal mediante el correlador y el sumador con la secuencia PN. . . . .</i>	68
3.13	<i>Banco de filtros triádico donde <math>h_0</math> y <math>h_1</math> son los filtros de análisis y síntesis para la subbanda paso-bajo, <math>b_0</math> y <math>b_1</math> son los filtros paso-banda y <math>g_0</math> y <math>g_1</math> los paso-alto. Al ser un filtro triádico la señal se diezma y se sobremuestra por 3. . . . .</i>	71
3.14	<i>Árbol de descomposición irregular en FTA. En el primer nivel de resolución 1, se supone que el banco de filtros triádico concentra en la subbanda paso-banda la interferencia. En el segundo nivel, se supone que la subbanda paso-alto del filtro triádico engloba la interferencia. En el tercer nivel se supone que la subbanda que concentra la energía de la interferencia es la paso-bajo del filtro diádico . . . . .</i>	72
4.1	<i>Diagrama de bloques del receptor FH-SS modificado, con filtro de supresión de interferencias de banda estrecha. . . . .</i>	81

4.2	<i>Desplazamiento en frecuencia y centrado de la señal sobre una subbanda.</i>	90
4.3	<i>Representación del espectro de los filtros de Daubechies con 4 coeficientes para cuatro niveles de resolución <math>j \in [4, \dots, 1]</math>. La figura muestra el ancho de banda de los filtros en diferentes niveles de resolución.</i>	91
4.4	<i>Espectro de los filtros de Daubechies con 4 coeficientes para el nivel <math>j = 4</math>. La gráfica muestra el intervalo de desplazamientos <math>d(j)</math> que se aplican en el caso <math>J = 4</math>.</i>	93
4.5	<i>Representación del espectro de los filtros de Daubechies con 4 coeficientes para cuatro niveles de resolución <math>j \in [4, \dots, 1]</math>. La figura muestra el alineado de las zonas de corte y paso-banda de los filtros en niveles de resolución consecutivos.</i>	97
4.6	<i>Pseudoódigo del algoritmo de análisis. En el algoritmo la función desplazamiento(<math>X,d</math>), aplica el desplazamiento en frecuencia <math>d</math> sobre la señal <math>X</math>. La función conv(<math>X_1, X_2</math>) realiza la convolución de las dos señales. Por último la función energía(<math>W</math>) calcula la energía del vector de entrada, como se definió en la ecuación 4.25, y la función valor de <math>d(J)</math> que maximiza <math> \Delta E[d(J)] </math> devuelve el índice <math>d(J)</math>, con mayor diferencia de energía entre las dos subbandas. El algoritmo distingue entre el primer nivel de resolución <math>J</math> y el resto <math>j \in [J - 1, \dots, 1]</math>.</i>	98
4.7	<i>Ejemplo de descomposición con el algoritmo de supresión de interferencias propuesto. La figura muestra los espectros de la señal y los filtros en cada nivel de resolución, comenzando en <math>J = 4</math>. Los filtros son los de Daubechies 4. También se muestra de forma esquemática las subbandas, que el algoritmo ha elegido en la descomposición y los vectores de entrada y salida en cada nivel.</i>	99
4.8	<i>Pseudoódigo del algoritmo de síntesis. En el algoritmo la función desplazamiento(<math>X,d</math>) aplica el desplazamiento en frecuencia <math>d</math> sobre la señal <math>X</math>. La función conv(<math>X_1, X_2</math>) realiza la convolución de las dos señales. Por último la función cancelar-desfase(<math>X</math>) cancela el desfase introducido por los filtros sobre el vector <math>X</math>. Este desfase depende de los filtros utilizados y del nivel de resolución considerado. Al igual que en el caso de análisis, el algoritmo distingue entre el primer nivel de resolución <math>J</math> y el resto <math>j \in [J - 1, \dots, 1]</math>. El algoritmo aplica directamente las ecuaciones de reconstrucción propuestas en 4.29 y 4.30.</i>	100
4.9	<i>Representación en frecuencia de las subbandas desarrolladas y los filtros de Daubechies 4 aplicados en cada nivel. Las señales representadas son: <math>X_6, X_5 = W_{6,1,D(6)}, X_4 = W_{5,1,D(5)}, X_3 = W_{4,1,D(4)}, X_2 = W_{3,1,D(3)}, X_1 = W_{2,1,D(2)}, X_0 = W_{1,0,D(1)}</math>.</i>	102

4.10	<i>Representación en el tiempo de las subbandas desarrolladas. Las señales representadas son: <math>X_6, X_5 = W_{6,1,D(6)}, X_4 = W_{5,1,D(5)}, X_3 = W_{4,1,D(4)}, X_2 = W_{3,1,D(3)}, X_1 = W_{2,1,D(2)}, X_0 = W_{1,0,D(1)}</math>. . . . .</i>	103
4.11	<i>Árbol de la transformación resultante. Las subbandas elegidas en cada iteración son: <math>X_5 = W_{6,1,D(6)}, X_4 = W_{5,1,D(5)}, X_3 = W_{4,1,D(4)}, X_2 = W_{3,1,D(3)}, X_1 = W_{2,1,D(2)}, X_0 = W_{1,0,D(1)}</math>. . . . .</i>	104
5.1	<i>Estructura Lattice para un banco de filtros sobremuestreado con 4 coeficientes. En la figura <math>X</math> es el vector de entrada, <math>A, K_1</math> y <math>K_2</math> son los coeficientes de la estructura y <math>P_1, Q_1, P_2, Q_2</math> son las salidas de las dos secciones que componen la estructura. Las salidas de la estructura, <math>P_2</math> y <math>Q_2</math>, son equivalentes a <math>H_0</math> y <math>G_0</math>. . . . .</i>	108
5.2	<i>Estructura Lattice de la Transformada Wavelet Sobremuestreada diádica, donde en cada nivel de iteración <math>j</math>, la matriz de la transformación con estructura Lattice <math>T(z)</math> se modifica según <math>T(z^{2^{j-1}})</math>. . . . .</i>	109
5.3	<i>Estructura Lattice de la Transformada Wavelet Sobremuestreada inversa con 4 coeficientes. En la figura las entradas <math>H'_0</math> y <math>G'_0</math> se corresponden con los filtros de síntesis del banco de filtros, <math>A, K_1</math> y <math>K_2</math> representan los coeficientes de la estructura y <math>\hat{X}</math> es la señal reconstruida. . . . .</i>	109
5.4	<i>Ejemplo de aplicación del proceso de análisis en el algoritmo de supresión de interferencias con <math>J=4</math>, mediante estructuras Lattice. Donde la aplicación del algoritmo ha determinado que las subbandas elegidas en cada nivel sean: <math>W_{4,1,D(4)}, W_{3,0,D(3)}, W_{2,1,D(2)}</math> y <math>W_{1,0,D(1)}</math> y los desplazamientos para centrar la interferencia sean: <math>D(4), D(3), D(2)</math> y <math>D(1)</math>. . .</i>	113
5.5	<i>Ejemplo de aplicación del proceso de síntesis en el algoritmo de supresión de interferencias con <math>J = 4</math>, mediante estructuras Lattice. En el proceso de síntesis se combinan las subbandas obtenidas en el análisis y se aplican los desplazamientos: <math>-D(4), -D(3), -D(2)</math> y <math>-D(1)</math>, para reconstruir la señal. . . . .</i>	114
5.6	<i>Bucle del algoritmo de filtrado Lattice para el DSP ADSP2065L. En la figura la columna MULT representa el multiplicador del DSP, la columna SUMA el sumador, las columnas DM y PM los buses de acceso a memoria, la columna derecha los registro internos ocupados (en color claro) y los desocupados (con color oscuro). . . . .</i>	119

6.1	<i>Diagrama de bloques del simulador construido sobre Matlab. En la figura se representan los módulos y las funciones empleadas en su construcción. El primer módulo es el emisor compuesto del generador de bits aleatorio, el generador de secuencia PN y modulador. El segundo módulo es el canal, que incorpora tres tipos de interferencias. El tercer módulo es el receptor que contiene el demodulador, el filtro paso-bajo, el algoritmo de supresión y el sumador no-coherente. El último módulo es el detector de error. . . . .</i>	133
6.2	<i>Gráfica de resultados experimento 1: variación de la frecuencia de la interferencia CWT. La gráfica representa 3 tipos de receptores: sin filtro, con filtro mediante FFT y con filtro mediante UWPT. La interferencia varía su posición en frecuencias, para situarse en las posiciones que van desde la de una de las portadoras, hasta el punto central del canal de la expansión FH. La potencia de la interferencia CWT es 14dB mayor que la de la señal FH-SS y la SNR del AWGN es de 12dB. . . . .</i>	136
6.3	<i>Gráfica de resultados experimento 2: variación del ruido AWGN. En la gráfica <math>E_b/J_0</math>(dB) es el SNR medido en decibelios y se representan 4 tipos de situaciones: receptor sin filtro, receptor con filtro mediante FFT, receptor con filtro mediante UWPT y el receptor sin filtro y sin interferencia CWT. La potencia de la interferencia CWT es 14dB mayor que la potencia de la señal FH-SS. . . . .</i>	138
6.4	<i>Gráfica de resultados experimento 3: variación de la potencia de la interferencia multitono. La gráfica representa 2 tipos de receptores: sin filtro y con filtro mediante UWPT. El SNR con respecto al AWGN es 12dB. En la gráfica <math>\alpha</math> es la relación entre la potencia de la portadora y la interferencia que afecta a un canal. . . . .</i>	140
6.5	<i>Gráfica de resultados experimento 4: comparación con FFH para interferencia multitono en el caso peor. En la gráfica <math>E_b/J_i</math>(dB) representa la relación entre la señal de comunicación y la interferencia, medida en decibelios. Se han considerado 3 tipos de receptores: sin filtro, con filtro mediante UWPT y con modulación FFH-SS. El SNR con respecto al AWGN es 12dB. . . . .</i>	142
6.6	<i>Gráfica de resultados experimento 5: ruido gaussiano de banda estrecha. En la gráfica <math>\mu</math> representa la fracción del canal FH-SS ocupado por la interferencia. Se comparan 2 tipos de receptor: sin filtro de supresión y con filtro mediante UWPT. El valor de <math>J_o</math> es 6dB superior a <math>E_b</math>, para todos los valores de <math>\mu</math>. . . . .</i>	143

6.7	<i>Gráfica de resultados experimento 6: variación del tamaño del bloque y del incremento del desplazamiento en frecuencias. La gráfica representa el receptor con filtro de supresión mediante UWPT, con 5 canales de frecuencia para la expansión FH, con 3 tamaños de bloque de la transformada: 8192, 4096 y 2048. La potencia de la interferencia CWT afecta a un único canal, siendo igual a la señal FH-SS. El SNR es 12dB.</i>	145
6.8	<i>Espectro de la interferencia CWT en la entrada y la salida del algoritmo. En la figura a) se representa la interferencia cuando el bloque de la transformada es 8192, considerando 3 niveles iniciales para el algoritmo UWPT, <math>J \in [8, 9, 10]</math>. En la figura b) se representa la interferencia cuando el bloque de la transformada es 4096, considerando 3 niveles iniciales para el algoritmo UWPT, <math>J \in [8, 9, 10]</math></i>	146
6.9	<i>Gráfica de resultados experimento 7: variación del número de niveles. La gráfica representa el receptor con filtro de supresión mediante UWPT, con 5 canales de frecuencia para la expansión FH y niveles iniciales <math>J \in [6, 7, 8, 9, 10]</math>. La potencia de la interferencia CWT afecta a un único canal y es igual a la señal FH-SS. El SNR con respecto a AWGN es 12dB.</i>	147
6.10	<i>Número de ciclos en la ejecución de la descomposición de análisis para el filtro de supresión mediante UWPT sobre el DSP ADSP2065L. En la gráfica <math>J</math> es el nivel inicial de la descomposición y <math>\Delta d'</math> es el desplazamiento normalizado.</i>	149
6.11	<i>Gráfica de resultados con variación del número de niveles, para el simulador sobre DSP y estructura Lattice. La gráfica representa el receptor con filtro de supresión mediante UWPT, con 5 canales de frecuencia para la expansión FH, para 3 niveles iniciales <math>J</math> distintos. La potencia de la interferencia CWT afecta a un único canal y es igual a la señal FH-SS. No se ha introducido AWGN.</i>	151
7.1	<i>Representación temporal de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento. Esta interferencia es del tipo CWT y está formada por multiples armónicos.</i>	157
7.2	<i>Espectro de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento. En la gráfica se observa que los componentes frecuenciales de la interferencia, la mayoría provenientes del variador y la bomba están concentrados sobre todo entre las frecuencias <math>(3-7)KHz</math>. Aunque también hay armónicos cercanos a frecuencia 0Hz provenientes de la señal de potencia del cable.</i>	158

7.3	<i>Detalle del espectro de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento. En la gráfica se observa los componentes frecuenciales de la interferencia CWT, para frecuencias (3 – 7)KHz, con armónicos situados en posiciones múltiplos de 250Hz. . . . .</i>	159
7.4	<i>Señal de comunicación con modulación FH-SS enviada desde el emisor hacia el receptor, donde se comprueba la variación del canal de frecuencia utilizado en cada símbolo transmitido. . . . .</i>	162
7.5	<i>Representación en frecuencia de la señal de comunicación con modulación FH-SS, enviada desde el emisor hacia el receptor. La señal queda concentrada en el intervalo de frecuencias (6 – 15)Khz, puesto que se han elegido 5 canales para la expansión FH-SS en ese rango de frecuencias.</i>	163
7.6	<i>Módulos que integran el receptor desarrollado dentro del proyecto PRIFEM. El primer módulo es analógico y forma el interface con la línea. El segundo módulo es el conversor analógico/digital, que muestrea la señal. Los correladores y el filtro paso-bajo llevan a banda base la señal. El filtro de supresión elimina las interferencias. El sincronizador determina el instante de comienzo de los símbolos y el último módulo completa la recepción no-coherente. . . . .</i>	164
7.7	<i>Proceso de sincronización realizado en el receptor. Sobre la cabecera formada por 5 bits se aplica la función de coste 7.3, <math>D(j)</math>, con valores de <math>j</math> en un intervalo de periodo 2ms, en total 96 puntos con una frecuencia de muestreo de 48KHz. El valor <math>j</math> que maximiza <math>D(j)</math> es el punto de inicio de la trama. En la figura se representan los dos niveles de los sumatorios, primero el sumatorio de la señal de salida de los correladores y posteriormente el sumatorio de todos los símbolos de la cabecera. . .</i>	167
7.8	<i>Espectro de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento.</i>	169
A.1	<i>Esquema de la arquitectura CIM. . . . .</i>	184
A.2	<i>Esquema de la arquitectura PRIFEM donde se muestran los elementos UC, UDA, UCE y URA. Hay que destacar que los elementos UCE se corresponde con el receptor del capítulo 7 y la URA con el emisor. . . .</i>	188
A.3	<i>Trama de bits enviada desde UCE a URA. . . . .</i>	192
A.4	<i>Trama de bits enviada desde URA a UCE. . . . .</i>	192

A.5	<i>Esquema de la tarjeta URA, con filtros de entrada y salida que se detallan en la sección siguiente, el amplificador de salida y el comaprador de entrada para recuperar la señal cuadrada, el microcontrolador y la fuente de energía, que almacena carga para el disparo de la salida de pulsos.</i>	193
A.6	<i>Fotografía de la tarjeta URA.</i>	194
A.7	<i>Filtro entrada.</i>	194
A.8	<i>Espectro del filtro de entrada.</i>	195
A.9	<i>Filtro salida.</i>	195
A.10	<i>Espectro del filtro de salida.</i>	196
A.11	<i>Esquema de la tarjeta UCE, con filtros de entrada y salida que se detallan en la sección siguiente, el amplificador de salida y el acondicionador de entrada para recuperar la señal y el DSP</i>	197
A.12	<i>Representación en el tiempo de la señal enviada desde la UCE con modulación OOK y codificación Manchester.</i>	198
A.13	<i>Representación en frecuencia de la señal enviada desde la UCE.</i>	199
A.14	<i>Fotografía de un hidrante de la instalación prototipo.</i>	200
A.15	<i>Mapa de situación de los hidrantes y esquema del cableado de la instalación.</i>	201
A.16	<i>Unidad Central compuesta por el PC y la UCE.</i>	202
A.17	<i>Imagen de la pantalla principal del programa PRIFEM UC.</i>	203
A.18	<i>Imagen de la pantalla del modo de pruebas del programa Prifem UC.</i>	203
B.1	<i>Estructura interna del DSP ADSP2065L.</i>	207



# Índice de Tablas

5.1	<i>Número de accesos a memoria para la implementación del proceso de análisis de la UWT con estructura Lattice, por punto del vector de entrada. . . . .</i>	111
5.2	<i>Número de accesos a memoria para la implementación del proceso de análisis de la UWT con estructura directa, por punto del vector de entrada.</i>	111
5.3	<i>Tabla de comparación de la implementación de la UWT con la estructura Lattice y la estructura directa, por punto del vector de entrada, siendo <math>L</math> el número de coeficientes de los filtros y <math>N</math> el número de elementos del vector de entrada. . . . .</i>	112



## Abreviaturas

Siglas y abreviaturas:

- ASIC *Application Specific Integrated Circuits*/ Circuitos Integrados de Aplicación Específica
- AWGN *Additive White Gaussian Noise*/ Ruido Blanco Gaussiano Aditivo
- BER *Bit Error Rate*/ Tasa de Bits Erróneos
- BFSK *Binary-FSK*/ FSK Binaria
- CDMA *Code División Multiple Access*/ Acceso Múltiple por División de Código
- CIM *Computer Integrated Manufacturing*/ Sistemas de Fabricación Integrados por Computador
- CPFSK *Continuos Phase Frequency Shift Keyed*/ FSK de Fase Continua
- CWT *Continuos Wave Tone*/ Onda con Tono Continuo
- DPSK *Differential Phase Shift Keyed*/ PSK Diferencial
- DSP *Digital Signal Processors*/ Procesadores Digitales de Señal
- DS-SS *Direct-Sequence Spread-Spectrum*/ Espectro Ensanchado en Secuencia Directa
- DWT *Discrete Wavelet Transform*/ Transformada Wavelet Discreta
- EPLD *Erasable Programmable Logic Device*/Dispositivo de Lógica Programable Reprogramable
- FH-SS *Frequency-Hopping Spread-Spectrum*/ Espectro Ensanchado con Saltos de Frecuencia
- FFH-SS *Fast-Frequency-Hopping Spread-Spectrum* /Espectro Ensanchado con Saltos Rápidos de Frecuencia
- FPGA *Field Programmable Gate Array*/ Campos de Matrices de Puertas Programables
- FSK *Frequency Shift Keyed*/ Manipulación por Desplazamiento de Frecuencia
- FT *Fourier Transform*/ Transformada de Fourier
- FTA *Adaptive Time-Frequency*/ Adaptativo en Tiempo-Frecuencia

- IDWT *Inverse Discrete Wavelet Transform*/ Transformada Wavelet Discreta Inversa
- OOK *On Off Keying*/ Manipulación por Encendido y Apagado
- PN *Pseudo-Noise*/ Pseudo-Noise
- PSK *Phase Shift Keyed*/ Manipulación por Desplazamiento de Fase
- RISC *Reduced Instruction Set Computer*/ Computadores con Repertorio de Instrucciones Reducido
- SPD *Spectral Power Density*/ Densidad Espectral de Potencia
- SFH-SS *Slow-Frequency-Hopping Spread-Spectrum*/ Espectro Ensanchado con Saltos Lentos de Frecuencia
- SNR *Signal Noise Ratio*/ Relación Señal a Ruido
- SS *Spread-Spectrum*/ Espectro Ensanchado
- STFT *Short Time Fourier Transform*/ Transformada de Fourier Enventanada
- UWT *Undecimated Wavelet Transform*/ Transformada Wavelet Sobremuestreada
- UWPT *Undecimated Wavelet Packet Transform*/ Transformada por Paquetes Wavelet Sobremuestreada
- VLSI *Very Large Scale Integration*/ Muy Alta Escala de Integración
- WPT *Wavelet Packets Transform*/ Transformada por Paquetes de Wavelet
- WT *Wavelet Transform*/ Transformada Wavelet

## Agradecimientos

En primer lugar, agradezco al codirector Dr. Santiago Felici toda su ayuda, comprensión e interés en el trabajo realizado. Ayuda reflejada en las explicaciones sobre temas relacionados con las Transformadas Wavelet, y también en los malos momentos pasados en la instalación de Sagunto, cuando las cosas no han funcionado del todo bien y era importante tener a alguien con el que afrontar las dificultades y exigencias, que plantea un trabajo de esta envergadura. Por último, también tengo que agradecer especialmente su paciencia, sus indicaciones y el tiempo que ha invertido en la corrección de la memoria de la tesis.

Agradezco al codirector Dr. Miguel Ángel Rodríguez en primer lugar su implicación en la realización de la tesis. En segundo lugar su atención, sus aportaciones e indicaciones sobre los algoritmos y los resultados obtenidos, para mí este diálogo ha sido muy importante, porque me ha permitido afianzar el trabajo realizado y encaminarlo de la forma adecuada. También tengo que agradecer sus comentarios sobre la estructura de la memoria y el tiempo que ha dedicado a su corrección.

Agradezco a Dr. Joan Pelechano, decano de la Facultad de Física, y a Dr. Gregorio Martín, director del *Institut de Robòtica* de la *Universitat de València*, haber iniciado la línea de trabajo mediante la gestión y la coordinación de los proyectos que han dado lugar a la tesis.

También he de reconocer la colaboración de todos las personas que han pasado por el grupo del proyecto PRIFEM, y que hoy en día desempeñan su labor en diversas empresas y otros centros de investigación. Entre ellos se encuentran Carlos Gómez, Tino Ortiz, Gracia Sánchez y José Pelegrí, y especialmente en el diseño y el funcionamiento de la instalación piloto de Sagunto a Joaquín Costa y a David Ferruz.

Asimismo tengo que recordar a los antiguos miembros del grupo de hardware del *Institut de Robòtica*, donde comencé mi camino dentro de la universidad: Rafael Martínez, Fernando Pardo, Carlos Pérez y José Antonio Boluda. También tengo que agradecer a Mariano Pérez las conversaciones sobre Wavelets, con las que hemos aclarado juntos algunos conceptos, y a Xaro Benavent, Esther de Ves y Pedro Murillo por su amistad y sus ánimos.

Por último quiero reservar un lugar especial a mi familia, mis padres, mis hermanos y a Sara, que me han dado siempre la mayor de las ayudas. Creo que nunca podré agradecerles suficientemente su apoyo, su cariño, y el haber compartido los buenos y los malos momentos en todos estos largos años.

A todos, gracias.



## Resumen

La transmisión de datos a través de canales ruidos o interferentes, puede verse ayudada por la utilización de modulaciones de Espectro Ensanchado. Estas modulaciones expanden el rango de frecuencias ocupado por la señal de comunicación, presentando cierto grado de inmunidad frente a ruido e interferencia de potencia elevada, que es capaz de degradar fuertemente las prestaciones de un sistema de comunicación.

En situaciones donde la potencia de la interferencia es muy elevada, el grado de inmunidad que ofrecen estas modulaciones puede resultar insuficiente, siendo necesario recurrir al procesamiento de la señal recibida, para aumentar las prestaciones del sistema. Las técnicas aplicadas en estos casos constituyen una línea de investigación con amplios antecedentes y muchas líneas por explorar. Dentro de este ámbito, el trabajo realizado se centra en las modulaciones de Espectro Ensanchado mediante Saltos en Frecuencia, normalmente usadas en situaciones donde no es posible mantener la coherencia de fase entre emisor y receptor. Para aumentar la inmunidad de estos sistemas de comunicación, se ha recurrido al procesamiento mediante Transformadas Wavelet de la señal modulada recibida.

En concreto, en la tesis se propone un algoritmo de supresión de interferencias de banda estrecha, para modulaciones de Espectro Ensanchado mediante Saltos en Frecuencia. Para separar la interferencia y la señal modulada, que combinadas forman la señal de entrada al receptor, se añade un nuevo módulo en el proceso de recepción encargado de realizar este filtrado. En el módulo se aplica un algoritmo de supresión basado en la Transformada Sobremuestreada en Paquetes Wavelet, que permite centrar la interferencia en la zona paso-banda de uno de los filtros en cada nivel de descomposición, mediante desplazamientos en frecuencia de la señal recibida, con un bajo coste computacional. Este alineado de la interferencia, con la zona paso-banda del espectro de los filtros, permite maximizar la eliminación de la potencia de la interferencia y degradar mínimamente la señal modulada.

El algoritmo propuesto se ha validado a través de un banco de pruebas formado por varios ensayos o partes. En primer lugar se han llevado a cabo simulaciones, con diferentes situaciones en cuanto a las características exactas de la modulación y diferentes tipos de interferencias de banda estrecha. Así, se han considerado interferencias del tipo onda continua y ruido gaussiano de banda estrecha, por ser las más comúnmente tratadas en la bibliografía y las que mejor se aproximan a las interferencias que aparecen en canales reales. En segundo lugar, a fin de poder aplicar el algoritmo de supresión en una situación real, se ha llevado a cabo la implementación de la Transformada Sobremuestreada en Paquetes Wavelet. Para ello, se han considerado diversas formas de realizar la descomposición de la señal, y de entre todas ellas se ha escogido una estructura mediante secciones *Lattice*.

Una vez decida la estructura final de la implementación, se ha completado la construcción del módulo de supresión mediante un dispositivo Procesador Digital de Señal,

que permite un desarrollo rápido y relativamente sencillo de los algoritmos de procesamiento de señal. A partir de estos algoritmos se ha obtenido una ecuación de coste real, que mide el número de ciclos de instrucción necesarios para completar el análisis y la síntesis, sobre la secuencia de valores de entrada al receptor.

Finalmente, el último paso en el proceso de validación, se ha llevado a cabo mediante una aplicación real. Dicha adaptación se ha realizado sobre un sistema de control industrial distribuido, aplicado al control y la gestión de grandes comunidades de regantes, que contiene un único cable como canal de comunicación y de alimentación de los elementos remotos. Para construir esta aplicación, desarrollada dentro del proyecto PRIFEM <sup>1</sup>, se han determinado las modulaciones y los elementos emisor y receptor, que mejor se han adaptado a las condiciones del medio de comunicación, pero siempre bajo las premisas introducidas al comienzo de la investigación.

Destacar que este cable de potencia constituye un buen banco de pruebas, porque contiene interferencias de onda continua generadas por las bombas inductivas presentes en la instalación de pruebas, mediante las que se ha comprobado el buen funcionamiento del algoritmo de supresión, desarrollado sobre estructuras *Lattice*.

---

<sup>1</sup>Programador para el control de Riego y Fertilización en sistemas Multiusuario (PRIFEM), proyecto financiado por la Generalitat Valenciana en el Plan Tecnológico del IMPIVA 1996 al 1999, y desarrollado dentro del *Institut de Robòtica* de la *Universitat de València*

# Parte I

## Punto de partida y objetivos



# Capítulo 1

## Introducción

### 1.1 Marco de la investigación realizada

En cualquier sistema de comunicación para la transmisión de datos, el canal impone una serie de restricciones que determinan la máxima cantidad de información que se puede transmitir, y que podrían concretarse en limitación de ancho de banda y ruido interferente [59].

La primera de estas restricciones viene dada por el ancho de banda, y acota el rango de frecuencias que puede ocupar la señal transmitida, debido a las características físicas del canal y a los componentes electrónicos usados en la implementación del emisor y el receptor.

La segunda restricción es el ruido contenido en el medio de transmisión. Existen muchos tipos de ruido e interferencias que pueden afectar a la señal de comunicación. Un ejemplo es el ruido de fondo, modelado normalmente como una señal aditiva con distribución gaussiana, que ocupa todo el rango de frecuencias. Otro ejemplo es la interferencia de onda continua y banda estrecha, que está constituida por una serie de armónicos situados en unas frecuencias concretas. Cuando la interferencia y la señal de comunicación ocupan la misma banda de frecuencias, las prestaciones del sistema se degradan. En esta situación, es posible minimizar el efecto del ruido aumentando la amplitud de la señal transmitida, aunque las limitaciones en el consumo de potencia o los componentes electrónicos utilizados pueden acotar dicho incremento.

Además de imponer estas restricciones, el canal es responsable de otros efectos nocivos como son la atenuación, la distorsión en fase o amplitud de la señal y las interferencias autoinducidas provenientes de las reflexiones, generadas en las desadaptaciones [59].

En este ámbito, el procesamiento de señal aplicado a las comunicaciones se convierte

## 1.1. Marco de la investigación realizada

---

en una herramienta, que puede ayudar enormemente a combatir los efectos nocivos del canal. Las técnicas integradas en este campo constituyen áreas de investigación con larga tradición y numerosas líneas abiertas. El objetivo es la adaptación de los sistemas para superar las restricciones impuestas por el medio, mediante el estudio de las modulaciones, la ecualización del canal, la supresión de interferencias, el diseño de los emisores y los receptores, etc.

En muchas situaciones, para mitigar los inconvenientes del canal, se recurre a las modulaciones con Espectro Ensanchado (*Spread Spectrum, SS*)<sup>1</sup>. Estas modulaciones tienen un alto nivel de inmunidad frente a interferencias, siendo las candidatas ideales para la transmisión de información en canales hostiles o ruidosos, o para compartir canales por múltiples usuarios, como en los sistemas de Acceso Múltiple por División de Código (*Code División Multiple Access, CDMA*) [24], utilizados en telefonía móvil.

Junto con las modulaciones de Espectro Ensanchado, se pueden aplicar técnicas de procesado basadas en transformaciones de la señal a diferentes dominios, con las que estudiar mejor los efectos nocivos del canal. Dentro de este conjunto de transformadas se encuentra la Transformada Wavelet (*Wavelet Transform, WT*)<sup>2</sup>. El concepto clave de esta transformada es la adaptación del análisis en el tiempo y la frecuencia de forma simultánea. En la última década, se han llevado a cabo multitud de aplicaciones de la transformada a problemas de procesado de la señal. Desde los trabajos iniciales de Daubechies [12] y Mallat [45], donde se muestra la relación entre las Wavelet y los bancos de filtros, se ha producido una constante evolución, que la ha llevado a convertirse en una herramienta flexible para la descomposición en multiresolución de señales, utilizada en multitud de campos. Algunas de sus aplicaciones en el área de las comunicaciones son: supresión de interferencias en modulaciones SS, construcción de códigos ortogonales para su aplicación en CDMA, modulaciones multitono, etc [2].

En el marco definido, el objetivo de la investigación realizada se centra en el estudio de las técnicas que mejoren la calidad de servicio de un sistema de comunicación, mediante la explotación del ancho de banda con una probabilidad de error reducida, en entornos o canales especialmente interferentes. Con este fin, la presente tesis hace uso de las modulaciones de Espectro Ensanchado combinadas con el tratamiento de la señal mediante Transformadas Wavelet, para eliminar de forma flexible la interferencia o ruido introducido. Esta combinación resulta interesante, ya que dichas modulaciones, junto con los algoritmos de supresión de interferencias, constituyen una buena técnica con la que incrementar la inmunidad de los sistemas de comunicación sobre canales ruidosos. Los métodos aplicados en estos casos constituyen una materia de investigación con amplios antecedentes [51], aunque su enorme potencial de aplicación la convierte en un área de desarrollo actual, con numerosas líneas abiertas [46].

---

<sup>1</sup>Las modulaciones de Espectro Ensanchado (*Spread Spectrum, SS*) se muestran con detalle en el capítulo 3

<sup>2</sup>La Transformada Wavelet (*Wavelet Transform, WT*) se estudia con detalle en el capítulo 2.

En concreto, la solución planteada en el presente trabajo aporta una técnica para la supresión de interferencias, en modulaciones mediante Espectro Ensanchado con Saltos en Frecuencia (*Frequency Hopping Spread Spectrum, FH-SS*). Dicha técnica esta constituida por un algoritmo de supresión incluido en el receptor FH-SS no-coherente, que descompone la señal mediante la Transformada Sobremuestreada en Paquetes Wavelet (*Undecimated Wavelet Packet Transforms, UWPT*), para aislar y eliminar el ruido y las interferencias recibidas. El uso de la Transformada Sobremuestreada en Paquetes Wavelet permite que el algoritmo minimice la distorsión de la señal de comunicación, al ser muy selectivo en el rango de frecuencias sobre el que actúa.

Cabe destacar, que la aplicación de estas modulaciones y algoritmos de recepción en sistemas reales se ve favorecida, en gran medida, por el actual desarrollo de los dispositivos de procesamiento hardware, que proporcionan implementaciones flexibles y eficaces. Los elementos más comúnmente utilizados dentro de este grupo son los Procesadores Digitales de Señal (*Digital Signal Processors, DSP*), con los que se consigue procesar señales digitales a alta velocidad y bajo coste.

Para aplicar las técnicas propuestas a un sistema real de comunicación, además de seleccionar el dispositivo sobre el que finalmente se programan los algoritmos, también es importante elegir la estructura con la que se realiza la transformación. En el caso de la Transformada Wavelet, esta estructura puede construirse de forma eficiente mediante convoluciones, pero existen otras alternativas basadas en secciones *Lattice* o en la división polifásica, que merecen ser consideradas y que permiten implementaciones menos costosas computacionalmente.

Estas dos últimas cuestiones, relacionadas con la implementación del algoritmo, se tratan con detalle en el capítulo 5.

## 1.2 Aplicación de la investigación realizada

Como prueba del buen funcionamiento del sistema propuesto, la investigación realizada se ha aplicado en la transmisión de datos, a través de un canal de poca capacidad y en un entorno altamente ruidoso e interferente, como es el cable de potencia. Concretamente, el escenario sobre el que se ha trabajado son los sistemas de control industrial

PRIFEM<sup>3</sup>, llevado a cabo dentro del *Institut de Robòtica* de la *Universitat de València*, cuyo objetivo ha sido el desarrollo de una arquitectura computacional, aplicada a la gestión y el control de instalaciones agrarias minifundistas, agrupadas en comunidades de regantes. El sistema construido se basa en un conjunto de unidades inteligentes distribuidas, que mediante su interconexión, hace posible el acceso a equipos remotos para el control de toda la instalación. El sistema permite la lectura y activación de sensores y actuadores remotos situados a distancias del orden de varios kilómetros. Para acceder y alimentar eléctricamente a dichos dispositivos, se utiliza únicamente un cable de potencia, que recorre toda la instalación y sirve a su vez de canal para la transmisión de datos.

El cable de potencia utilizado como medio de transmisión es muy hostil, conteniendo altos niveles de ruido, largas distancias de cable y está desadaptado en las terminaciones y las conexiones. Todos estos inconvenientes hacen imposible la aplicación de buses de campo comerciales, en la interconexión de los sensores y actuadores remotos. Por lo tanto, ha sido necesario desarrollar una solución con la que cubrir las necesidades planteadas en el proyecto. Además, las características adversas del canal convierten el sistema, en un buen banco de pruebas donde aplicar los algoritmos de filtrado de interferencias y las modulaciones introducidas en la sección anterior.

## 1.3 Objetivos de la tesis

A a partir de la línea de investigación introducida y el proyecto PRIFEM, descrito en el apartado anterior en el cual se enmarca el presente trabajo de investigación, los objetivos que se han planteado son:

*El estudio y la investigación de modulaciones de Espectro Ensanchado, junto con receptores y algoritmos que ayuden a aumentar la robustez de dichas modulaciones, frente a interferencias de banda estrecha. Estos algoritmos se basarán en la Transformada Wavelet y deben tener una implementación eficiente, que permita la aplicación de los resultados obtenidos al desarrollo de un sistema de comunicación por cable de potencia.*

Para completar estos objetivos, se van a plantear una serie de fases o pasos encaminados en este sentido:

- Estudio de las modulaciones de Espectro Ensanchado. En particular el estudio se centra en las modulaciones de Espectro Ensanchado con Saltos de Frecuencia (*Frequency Hopping Spread Spectrum, FH-SS*), realizando un repaso de sus características más importantes.

---

<sup>3</sup>Programador para el control de Riego y Fertilización en sistemas Multiusuario (PRIFEM), proyecto financiado por la Generalitat Valenciana en el Plan Tecnológico del IMPIVA 1996 al 1999.

- Estudio de las técnicas de procesado de señal capaces de aumentar la inmunidad de estas modulaciones frente a interferencias, sobre todos las basadas en el procesado de la señal en el dominio de la Transformada Wavelet.
- Propuesta de un nuevo receptor que incluya este tipo de procesado, encaminado a la eliminación de ruido en las modulaciones con Espectro Ensanchado por Saltos de Frecuencia, utilizando Transformadas Wavelet Sobremuestreadas.
- Realización de simulaciones que validen las características del algoritmo propuesto, considerando diversos tipos de ruido.
- Estudio y comparación de las estructuras que permitan implementar de forma eficiente las Transformadas Wavelet Sobremuestreadas. En concreto se tienen en cuenta las estructuras *Lattice*, la forma Polifásica y la forma directa, como se ve en el capítulo 5.
- Implementación eficiente de los algoritmos de filtrado basados en Transformadas Wavelet Sobremuestreadas con estructuras *Lattice*, en procesadores digitales de señal. En este desarrollo se ha utilizado el dispositivo *ADSP21065L*.
- Prueba de los algoritmos propuestos sobre el cable de potencia del proyecto PRIFEM, visto en el apartado 1.2.
- Implementación de los módulos emisores y receptores correspondientes a la modulación mediante Espectro Ensanchado con Saltos en Frecuencia, que incluyen los módulos de supresión de interferencias.
- Comprobación experimental del funcionamiento del sistema en una instalación prototipo. Recogida de datos y estudio de los resultados obtenidos.

## 1.4 Breve descripción de los contenidos

La tesis se ha dividido en cuatro partes que recogen cada uno de los pasos seguidos en la metodología de la investigación. La primera parte plantea el problema encontrado y lo enmarca dentro del contexto en que se ha realizado la investigación. La segunda trata los fundamentos teóricos, en los que se basan las Transformadas Wavelet y las modulaciones de Espectro Ensanchado, junto con las técnicas de eliminación de interferencias. La tercera parte engloba la investigación aportada en este trabajo. Dentro de esta parte se incluye el diseño del algoritmo de supresión de interferencias, junto con su implementación eficiente en un dispositivo del tipo Procesador Digital de Señal. La cuarta parte contiene las simulaciones que validan el receptor y el algoritmo de supresión. Además, esta parte explica las pruebas llevadas a cabo sobre la arquitectura de control del proyecto PRIFEM. Para concluir, la quinta parte aporta las conclusiones y el trabajo futuro, a partir del estado en el que se sitúa la investigación aportada.

Los capítulos que componen la tesis son:

- El presente capítulo contiene una breve introducción del contenido general de la tesis y también se dan los objetivos que se pretenden alcanzar, previos al comienzo de la investigación.
- En el segundo capítulo se hace un repaso de las Transformadas Wavelet, ya que van a constituir una herramienta muy importante en todo el trabajo. Se comienza introduciendo la definición formal de la transformada y se continúa mostrando sus variantes, hasta completar la exposición de las Transformadas Wavelet Sobremuestreadas, que tendrán una función clave dentro de los algoritmos desarrollados en la tesis. Para terminar el capítulo, se examinan estructuras eficientes en la implementación de la transformada.
- El tercer capítulo incluye el estudio de las modulaciones de Espectro Ensanchado. Dentro de la primera parte del capítulo se recogen las características más importantes y se muestra su rendimiento en canales con distintos tipos de ruido. La segunda parte del capítulo se centra en las técnicas más importantes para la eliminación de interferencias.
- El cuarto capítulo comienza proponiendo un nuevo receptor para modulaciones FH-SS, que incluye el procesamiento en el dominio de la transformada para la supresión de interferencias. A continuación, se consideran las posibles transformadas que pueden ser aplicadas, razonando la elección realizada y mostrando su adaptación al algoritmo de supresión.
- El quinto capítulo examina las estructuras que se pueden aplicar en la implementación del algoritmo y propone una basada en filtros *Lattice*. Esta estructura se desarrolla sobre un dispositivo del tipo Procesador Digital de Señal.
- El sexto capítulo muestra los resultados obtenidos mediante simulación del rendimiento del receptor, aplicado sobre un canal con diversos tipos de interferencias.
- En el séptimo capítulo se ve la adaptación del receptor y el algoritmo propuesto, a la construcción de las unidades emisor y receptor de la arquitectura PRIFEM. También como parte de este capítulo, se explica el proceso de experimentación y la toma de resultados sobre la instalación piloto, en la que ha sido probado el algoritmo.
- El octavo capítulo resume las aportaciones realizadas y determina las conclusiones del trabajo. Además incluye las líneas de trabajo que la tesis deja abiertas.

Especial interés tienen en la tesis los anexos aportados. En el primero de ellos se incluye un repaso breve al sistema de control propuesto dentro del proyecto PRIFEM. También se incluyen los elementos integrados en la construcción del emisor y receptor.

El segundo anexo contiene las características básicas del Procesador Digital de Señal *ADSP21065L*, ya que sus propiedades internas influyen en la implementación del algoritmo realizada en el capítulo quinto. Para finalizar, los anexos tercero y cuarto muestran el software usado en la simulación del algoritmo y en la programación del Procesador Digital de Señal.

#### 1.4. Breve descripción de los contenidos

---

## Parte II

### Revisión de la situación actual: estado del arte



---

Los contenidos de esta parte de la memoria están dedicados a la revisión de la situación actual y los antecedentes, de aquellos elementos que son relevantes para el trabajo de investigación realizado. Dichos contenidos están estructurados en los siguientes capítulos:

- Representación de señales en el tiempo, frecuencia y escala
- Comunicaciones con modulaciones de Espectro Ensanchado

---

# Capítulo 2

## Representación de señales en el tiempo, frecuencia y escala

### 2.1 Introducción

Una gran variedad de fenómenos físicos englobados en áreas como las comunicaciones, la acústica, la sismología, la ingeniería biomédica o en el procesamiento de la voz y la imagen están modelados mediante señales. Estas señales se representan matemáticamente por una función de una o varias variables, aunque normalmente la única variable independiente es el tiempo.

En muchas aplicaciones la representación del espacio temporal no es suficiente, siendo necesario conocer la información de las frecuencias contenidas en la señal. Esta información se obtiene transformando dicha señal desde el espacio temporal al espacio de frecuencias, utilizando la Transformada de Fourier (*Fourier Transform, FT*) [61]. Pero al transformar la señal se pierde toda la información temporal. La señal en el espacio de Fourier representa la intensidad de cada uno de los armónicos, sin ningún tipo de información sobre el instante en el cual se produjo esa frecuencia. Para señales estacionarias, cuyas propiedades no varían en el tiempo, el análisis de Fourier es suficiente para caracterizar por completo la señal. Sin embargo, si contiene cambios rápidos en el tiempo de sus componentes frecuenciales, la FT no es capaz de detectarlos.

Una solución que permite la representación conjunta en tiempo-frecuencia es la Transformada de Fourier Enventanada (*Short Time Fourier Transform, STFT*) [61]. La localización en el tiempo del espectro de frecuencias comienza tomando un pequeño trozo de la señal, mediante alguna función ventana de extensión temporal limitada. Asumiendo que en ese intervalo la señal está próxima a ser estacionaria, se realiza la FT sobre la señal resultante.

Pero hay señales, que por su naturaleza, cambian muy rápidamente su contexto de

frecuencias en el tiempo. En estos casos es difícil encontrar una ventana apropiada, con la que representar la señal. Se podría pensar que reducir el tamaño de la ventana puede ayudar en la búsqueda de dicho intervalo, pero aparece el problema de que este aumento en la resolución temporal hace que disminuya la resolución frecuencial, debido el principio de incertidumbre de Heisenberg. Este principio limita la máxima resolución conjunta tiempo-frecuencia que se puede lograr, cumpliendo la inecuación [65]:  $\Delta t \Delta f \geq \frac{1}{4\pi}$ , donde  $\Delta t$  es la dispersión temporal y  $\Delta f$  es la frecuencial. En la ecuación anterior se consigue la igualdad cuando  $f$  es un múltiplo de  $e^{-\alpha t^2}$ , siendo  $\alpha \geq 0$  una constante real.

En este contexto aparecen las Wavelets. La teoría de las Wavelet cubre una área muy amplia, que va desde el procesado de señales continuas, hasta la construcción de bancos de filtros en codificación subbanda de señales discretas. En particular la Transformada Wavelet (*Wavelet Transform, WT*) es de interés en el análisis de señales no estacionarias, siendo una alternativa a la STFT. La diferencia más significativa entre ambas transformadas es, que mientras la STFT utiliza una única ventana en el procesado de toda la señal, la WT utiliza pequeñas ventanas para altas frecuencias y grandes ventanas para pequeñas frecuencias.

A lo largo de este capítulo, se van a presentar las diversas partes en las que históricamente se ha dividido la teoría de las Wavelets. En primer lugar, la sección 2.2 define la transformada, junto con el marco matemático en el que se enmarca definido por Daubechies [13]. La segunda sección 2.3 muestra el esquema de multiresolución desarrollado por Mallat [43], mediante el que se relacionan las Wavelets y los bancos de filtros. La sección 2.4 se centra en el filtrado subbanda de señales discretas y las condiciones que se imponen en el diseño de los filtros. A continuación, en las secciones 2.5 y 2.6, se consideran dos variantes de la WT con una división del espacio tiempo-frecuencia en intervalos con estructura irregular. Para concluir el capítulo, la sección 2.7 presenta varias alternativas en la implementación de la transformada, dependiendo de las características del banco de filtros que la define.

## 2.2 Transformada Wavelet

La WT es una representación lineal tiempo-frecuencia similar a la transformada de Fourier enventanada. Esta transformada está definida como:

$$WT(x(t))(\tau, a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t - \tau}{a}\right) dt \quad (2.1)$$

donde  $a \in \mathbb{R}$ ,  $a \neq 0$ . La WT puede verse como el producto interno de la señal  $x(t)$  con una familia de funciones indexadas por dos parámetros  $(\tau, a)$ . Estas funciones, llamadas funciones Wavelets, pueden formar una base bajo ciertas condiciones, tal y

como se describe en la sección 2.2.2. De forma matemática las Wavelets se expresan de la siguiente manera:

$$\psi_{\tau,a}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-\tau}{a}\right) \quad (2.2)$$

La normalización se elige para que se cumpla:

$$\|\psi_{\tau,a}\|^2 = \|\psi\|^2 \quad (2.3)$$

La función básica  $\psi_{0,1}=\psi$  es la llamada Wavelet madre, función que debe cumplir tres condiciones:

- Ser una función de energía finita, pasa-banda y media cero [65].
- Debe estar bien localizada, presentando una fuerte caída tanto en frecuencias como en el tiempo.
- Debe cumplir la relación  $C_\psi = \int_0^\infty \frac{|\hat{\psi}(w)|^2}{w} dw < \infty$ , lo que asegura una representación completa y estable [45].

Las propiedades de las funciones base de la STFT y las Wavelets son diferentes. La desigualdad parte de la forma en que se obtienen ambos tipos de funciones. Así, mientras las funciones base de la STFT se construyen aplicando desplazamientos temporales y frecuenciales de la ventana, las Wavelets  $\psi_{\tau,a}$  se crean mediante desplazamientos y escalado temporal.

Las funciones base de la STFT consisten en la misma ventana trasladada hasta una cierta posición en el tiempo y rellena por oscilaciones de una determinada frecuencia. Esta elección de las funciones impone, que la resolución conjunta tiempo-frecuencia obtenida sea fija sobre todo el plano de representación.

Por el contrario las Wavelets se adaptan, siendo más anchas en el tiempo cuando contienen frecuencias más bajas y al contrario. Al incrementar la escala, aumenta el valor de  $a$  y las funciones  $\psi_{\tau,a}$  se expanden en el tiempo. Este incremento permite que la transformada extraiga el comportamiento de la señal en grandes intervalos temporales [65]. Si se considera la aplicación de la WT como si se tratara de la aplicación de un filtro, la modificación del parámetro de escalado  $a$  cambia la zona del espectro donde se sitúa dicho filtro. Así, los valores de  $|a|$  grandes se corresponden con frecuencias bajas y los valores pequeños de  $|a|$  lo hacen con frecuencias altas. Además, el ancho de banda de los filtros obtenidos al cambiar la escala también varía, siendo proporcional a la frecuencia central. Esta es la razón de que el análisis con Wavelets funcione bien para señales con componentes de alta frecuencia y pequeña duración, junto a componentes de larga duración y baja frecuencia, que es en la práctica el comportamiento más normal de las señales naturales [65].

### 2.2.1 Discretización de los parámetros tiempo-escala

Debido a propósitos prácticos de implementación computacional, el parámetro de escala se discretiza tomando potencias enteras de un valor fijo de dilatación  $a_0 > 1$ , con  $a = a_0^j, j \in \mathbb{Z}$ . Con esta elección, la localización en el tiempo se incrementa conforme  $j$  sea menor (negativo y grande), ya que las funciones Wavelet se contraen en el eje temporal al disminuir  $j$ . La discretización del parámetro  $\tau$  depende de  $j$ , en concreto la relación entre ambos es:  $\tau = nt_0 a_0^j$  donde  $t_0 > 0$  es fijo y  $n, j \in \mathbb{Z}$  [13] [65]. Así, cuando la Wavelet es estrecha se desplaza mediante pasos más pequeños para cubrir los detalles de la señal. Como resultado de este proceso de discretización se obtienen las siguientes Wavelets discretas:

$$\psi_{j,n}(t) = \psi_{a_0^j, nt_0 a_0^j}(t) = a_0^{-j/2} \psi(a_0^{-j} t - nt_0) \quad (2.4)$$

El enrejado de localizaciones tiempo-frecuencia de la WT con una discretización diádica, ( $a_0 = 2, t_0 = 1 \implies a = 2^j, \tau = n2^j$ ), es el caso más utilizado en la bibliografía. Al tomar estos valores para los parámetros, en cada nivel de detalle la resolución frecuencial y temporal se escala por un factor 2, produciendo un grid de intervalos de resolución modelado en octavas. La figura 2.1 muestra para este caso, la forma en la que las funciones Wavelet dividen el espacio tiempo-frecuencia y se compara con la STFT.

En el resto del capítulo se utiliza siempre **muestreo diádico**, ya que es el más usual en la bibliografía y es el usado en el desarrollo de los algoritmos de procesado de señal, que se introducen en capítulos posteriores.

Figura 2.1: *División del espacio tiempo/frecuencia para STFT y WT. En el caso de la WT se supone una descomposición diádica.*

Una función  $s(t)$  con energía finita puede representarse en función de los coeficientes

obtenidos mediante el producto interior definido como:

$$b_{j,n} = \int_{-\infty}^{\infty} s(t)\psi_{j,n}(t)dt \quad (2.5)$$

aproximando la señal original mediante la siguiente ecuación de reconstrucción:

$$s(t) \simeq k \sum_j \sum_n b_{j,n}\omega_{j,n} \quad (2.6)$$

Esta ecuación de reconstrucción no siempre es una buena aproximación de la función  $s(t)$ , para que lo sea es necesario que  $\psi, a_0$  y  $t_0$  cumplan una serie de requisitos. Estos requisitos imponen que  $\{\psi_{j,n}; j, n \in \mathbb{Z}\}$  constituya un *frame*, como se describe en la sección 2.2.2. Además, el conjunto de funciones  $\{\omega_{j,n}; j, n \in \mathbb{Z}\}$  con el que se reconstruye la señal original se denomina el *frame* dual [13].

En el caso de que  $a_0$  sea cercano a 1 y  $t_0$  suficientemente pequeño, las funciones Wavelet están sobremuestreadas y las condiciones sobre  $\{\psi_{j,n}\}$  para obtener una buena reconstrucción son poco restrictivas. Por el contrario, si se aplica un muestreo crítico, el número de muestras a la entrada y la salida de la transformada es el mismo y las condiciones sobre  $\{\psi_{j,n}\}$  son más restrictivas [65].

## 2.2.2 Wavelet frames

La teoría de las *Wavelet Frames* [13] genera un marco general con el que estudiar, las condiciones de convergencia de la ecuación de reconstrucción 2.6. Estas condiciones de convergencia, imponen que el conjunto de funciones  $\{\psi_{j,n}\}$  [13] con una determinada densidad de muestreo, determinado por los parámetros  $a_0$  y  $t_0$ , constituyan un *frame*. A continuación se introduce y se define el concepto de *frame*:

- El Espacio de Hilbert  $H$  es un espacio lineal sobre el que se define una operación que hace corresponder a cada par ordenado de elementos de  $H$  un número complejo llamado producto interior o escalar.
- El espacio  $L^2(\mathbb{R})$  es el espacio de dimensión infinita formado por las funciones unidimensionales de cuadrado integrable, funciones de energía finita,  $s(t)$  que cumplen:  $(\int_{-\infty}^{\infty} |s(t)|^2 dt)^{1/2} < \infty$ .

*Definición:* El conjunto de funciones  $\{\psi_{j,n}; j, n \in \mathbb{Z}\}$  constituyen un *frame* en el espacio de Hilbert  $L^2(\mathbb{R})$ , si  $\forall s(t) \in L^2(\mathbb{R})$  se cumple la siguiente relación:

$$A \| s \|^2 \leq \sum_i \sum_j \| b_{j,n}(t) \|^2 \leq B \| s \|^2 \quad \text{con } A > 0, B < \infty \quad (2.7)$$

donde  $A$  y  $B$  son los llamados límites del *frame*. Una familia de funciones *Wavelet* constituyen un *frame* si la energía de los coeficientes  $b_{j,n}$ , en relación a la energía de la señal se concentra entre las dos constantes positivas  $A$  y  $B$ . Dadas la funciones  $\{\psi_{j,n}; j, n \in \mathbb{Z}\}$  que constituyen un *frame*, existe un conjunto de funciones  $\{\omega_{j,n}; j, n \in \mathbb{Z}\}$  que forman su *frame* dual [13]. El papel de los *frames* es intercambiable, pudiéndose descomponer la señal en  $\{\omega_{j,n}\}$  y reconstruirse mediante  $\{\psi_{j,n}\}$ .

Existen muchas funciones que cumplen las condiciones 2.7 y según la relación de los valores  $A$  y  $B$  con  $A > 0, B < \infty$ , se obtienen propiedades diferentes. El primer caso de interés se obtiene cuando se cumple  $A = B = 1$ . Al ser ambas constantes iguales a 1, el *frame* está formado por elementos normalizados y constituye una base ortonormal y se llama *frame* ortogonal. Es un conjunto linealmente independiente y la base,  $\{\psi_{j,n}\}$ , coincide con la base dual,  $\{\omega_{j,n}\}$ . En este caso, si se tiene el desarrollo de la función  $s(t)$  en  $\{\psi_{j,n}\}$  mediante los coeficientes  $b_{j,n}$ , definidos en la ecuación 2.5, entonces la función puede reconstruirse mediante la ecuación:

$$s(t) = \sum_j \sum_n b_{j,n} \psi_{j,n} \quad (2.8)$$

Si  $A = B \neq 1$  se tiene un *tight frame* o marco hermético. En este caso la funciones  $\{\psi_{j,n}\}$  no forman una base porque el conjunto es redundante. Si los elementos están normalizados, el grado de redundancia viene dado por el valor  $A$ . La reconstrucción de la función  $s(t)$  a partir de la base dual sería:

$$s(t) = \sum_j \sum_n b_{j,n} \omega_{j,n} \quad (2.9)$$

Y en función de  $\psi_{j,n}$  [13]:

$$s(t) = A^{-1} \sum_j \sum_n b_{j,n} \psi_{j,n} \quad (2.10)$$

Hay que destacar que dado el *frame*  $\{\psi_{j,n}\}$  el conjunto de funciones con el que se puede realizar la reconstrucción no es único, se pueden encontrar varios conjuntos de funciones, que pueden desempeñar el papel de base dual. Esto es debido a que las funciones  $\{\psi_{j,n}\}$  que componen el *frame* no tienen porque ser linealmente independientes, forman un conjunto redundante con más elementos de los que habría en una base [13].

Por último hay que destacar, el caso de un *frame* cuyos vectores son linealmente independientes y están normalizados, constituyendo una base de Riesz. Esta base es el *frame* más cercano a una base ortonormal, siendo la *frame* dual también una base de Riesz. En esta situación se dice que las bases forman un sistema biortogonal, siendo los elementos de la base ortogonales a los elementos de la base dual [45]. La

reconstrucción en esta situación se realiza mediante la base dual utilizando la ecuación 2.9

## 2.3 Multiresolución

Al introducir el concepto de multiresolución se introduce un nuevo nivel de estructuración, ya que ahora aparece una autosimilitud a diferentes escalas de resolución entre las funciones base.

La multiresolución es una de las técnicas para obtener expansiones lineales de señales en bases de tipo Wavelet, localizadas en ambos dominios y con diferentes resoluciones espaciales para cada banda de frecuencia. El análisis multiresolución da lugar a subespacios de diferentes resoluciones, los cuales representan proyecciones del espacio continuo  $L^2(\mathbb{R})$  a través de coeficientes que generan una expansión lineal discreta. Ello supone representar la señal con aproximaciones sucesivas, manejando tanto las aproximaciones paso-bajo como los detalles de la misma.

A continuación, se formula la definición axiomática de un análisis multiresolución y sus propiedades, definición introducida por Mallat [43] y desarrollada formalmente en [12]. El estudio realizado está basado en [45], donde se particulariza en el análisis multiresolución ortogonal con muestreo diádico.

En el apartado 2.4, se estudia la estrecha relación existente entre la representación multiresolución y las Transformadas Wavelet basadas en iteración de bancos de filtros [43], y como bajo ciertas condiciones, los bancos de filtros pueden generar representaciones de este tipo [12].

### 2.3.1 Definición formal

*Definición:* Un análisis multiresolución ortogonal para  $L^2(\mathbb{R})$  consiste en una sucesión expansiva de subespacios cerrados, que llamaremos  $V_j$ , que cumplen las siguientes axiomas [43]:

- *inclusión de subespacios:* Sea  $V_\infty$ , el subespacio funcional de mínima resolución y sea  $V_{-\infty}$ , el subespacio de máxima resolución, entonces  $V_\infty \subset \dots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \dots V_{-\infty}$
- *completitud hacia arriba:*  $\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R})$ , la clausura de subespacios es  $L^2(\mathbb{R})$ , que puede expresarse también como  $j \in \mathbb{Z}, V_j \rightarrow L^2(\mathbb{R})$  para  $j \rightarrow -\infty$ ,

### 2.3. Multiresolución

---

es decir, que los subespacios contienen funciones base arbitrariamente cercanas<sup>1</sup> a cualquier función de cuadrado integrable.

- *completitud hacia abajo*:  $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$ , que puede expresarse también como  $V_j = \{0\}$  para  $j \rightarrow \infty$

resolución concentra una zona del espectro de frecuencias, que es la mitad de la zona contenida en el nivel anterior de mayor resolución. Este comportamiento es conocido como una descomposición frecuencial en octavas [43]. Dentro de este proceso de descomposición, la función  $H_0(e^{j\omega})$ , que es  $2\pi$  periódica, caracteriza totalmente al análisis multiresolución.

Sean  $\varphi_{0,0}$  y  $\varphi_{0,i}$  funciones base del subespacio funcional  $V_0$ , si se aplica la condición de ortonormalidad en dicho subespacio y aplicando el *Teorema de Rayleigh*, se obtiene en el dominio de Fourier:

$$\langle \varphi_{0,0}, \varphi_{0,n} \rangle = \frac{1}{2} \int_{-\infty}^{\infty} |\Phi(\omega/2)|^2 e^{j\omega n} d\omega = 0 \text{ si } n \neq 0 \quad (2.13)$$

y dividiendo el eje de frecuencias en tramos de  $2\pi$ , se puede expresar:

$$\langle \varphi_{0,0}, \varphi_{0,n} \rangle = \frac{1}{2} \int_0^{2\pi} e^{j\omega n} \sum_{k=-\infty}^{\infty} \left| \Phi\left(\frac{\omega + 2\pi k}{2}\right) \right|^2 d\omega \quad (2.14)$$

que con ello, se obtiene la condición necesaria de ortonormalidad con la expresión, conocida como fórmula de Poisson [12]:

$$\sum_{k=-\infty}^{\infty} \left| \Phi\left(\frac{\omega + 2\pi k}{2}\right) \right|^2 = 1 \quad (2.15)$$

Esta condición  $2\pi$  periódica gobierna la estabilidad y la ortonormalidad en la multiresolución de las translaciones de  $\varphi$ . La condición de ortonormalidad anterior puede relajarse, limitándose a exigir únicamente que  $\{\psi_{0,n}; n \in \mathbb{Z}\}$  sea una *base de Riesz* [43]. En este caso, tal como se ha visto en el apartado 2.2.2, la base dual permite la reconstrucción perfecta de la señal. La condición necesaria de ortonormalidad 2.15, utilizando la ecuación 2.12, puede expresarse como:

$$|H_0(e^{j\omega})|^2 + |H_0(e^{j(\omega+\pi)})|^2 = 2 \quad (2.16)$$

Si las funciones de escalado, según la definición de multiresolución han de representar cualquier  $L^2(\mathbb{R})$ , por las condiciones de inclusión de subespacios y la completitud hacia arriba, entonces:

$$\int_{-\infty}^{\infty} \varphi(x) dx \neq 0 \quad (2.17)$$

que además caracteriza su suavidad y continuidad (condición necesaria de estabilidad, que expresado en el dominio de Fourier equivale a  $\Phi(0) \neq 0$ , es decir la componente continua sea no nula. Por tanto considerando 2.12, se deduce:

$$|H_0(1)| = \sqrt{2} \quad (2.18)$$

por otro lado en  $\omega = \pi$ , despejando a partir de la ecuación 2.16 y 2.18, se obtiene:

$$H_0(-1) = 0 \tag{2.19}$$

quedando así perfectamente descrito el análisis multiresolución por la función  $2\pi$  periódica  $H_0(e^{j\omega})$ .

### 2.3.2 Subespacios complementarios ortogonales

Se ha visto, que un análisis multiresolución está caracterizado por una función  $H_0(e^{j\omega})$   $2\pi$  periódica con algunas propiedades adicionales. La definición axiomática de multiresolución garantiza la existencia de bases y la aproximación en espacios  $V_j$ .

Figura 2.2: *Descomposición espectral de subespacios para las técnicas de multiresolución, aplicando descomposición diádica.  $V_i$  subespacio de residuos,  $W_i$  subespacio de detalles*

Si se particulariza para el caso ortogonal, entre un subespacio  $V_j$  y el subespacio  $V_{j-1}$  de mayor resolución, debido a que el segundo subespacio incluye al primero y la representación es ortogonal, queda un subespacio llamado  $W_j$  que representa los detalles que es necesario añadir a las funciones pertenecientes al subespacio  $V_j$ , para pasar al nivel de resolución siguiente  $V_{j-1}$ , tal como se representa en la figura 2.2. Esta idea es la que justifica la *función Wavelet*, a partir de la cual se construyen por escalado y traslación las funciones que generan estos subespacios Wavelet  $W_j$ .

Al pasar de una resolución mayor a otra menor, se puede separar por una parte la versión aproximada, que pasa a menor resolución, y por otra parte el detalle que se pierde al cambiar de nivel. Este proceso genera una serie de aproximaciones sucesivas a medida que se van añadiendo más detalles a una determinada proyección.

Al igual que existe la base ortogonal para los subespacios  $V_j$ , se puede demostrar la existencia de una nueva base para los subespacios de detalle  $W_j$  [45]. Así, dada la

sucesión expansiva de subespacios que cumplen los axiomas de multiresolución, existe una base ortonormal para  $L^2(\mathbb{R})$ :

$$\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n), \quad j, n \in \mathbb{Z} \quad (2.20)$$

tal que,  $\{\psi_{j,n}\}$  con  $j, n \in \mathbb{Z}$  es una base ortonormal para  $W_j$ , donde  $W_j$  con  $j \in \mathbb{Z}$  es el complemento ortogonal de  $V_j$  en  $V_{j-1}$ .

Este resultado se obtiene al considerar que  $V_{j-1} = V_j \oplus W_j$  con  $W_j \perp W_{j'}$  si  $j \neq j'$ . De esta forma, toda  $s(t) \in L^2(\mathbb{R})$  donde  $P_j s$  representa la proyección ortogonal de  $s$  sobre  $V_j$ , se puede expresar como:

$$P_{j-1}s = P_j s + \sum_{n \in \mathbb{Z}} \langle s, \psi_{j,n} \rangle \psi_{j,n} \quad (2.21)$$

donde el significado del sumatorio consiste en la incorporación de los detalles en la nueva representación a partir de una resolución menor. Esto da lugar a que para cualquier  $V_j$ ,  $j < J$ , este subespacio se exprese como la suma de un subespacio de una determinada aproximación  $V_J$  de menor resolución, más todos los subespacios de detalle hasta dicha resolución:

$$V_j = V_J \oplus W_{J-1} \oplus W_{J-2} \oplus \dots \oplus W_{j+1} \quad (2.22)$$

que junto con las propiedades de completitud hacia arriba y hacia abajo implica que:

$$L^2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_j \quad (2.23)$$

es decir, que los subespacios  $W_j$  forman una descomposición de  $L^2(\mathbb{R})$  en subespacios mutuamente ortogonales con la propiedad de escalado  $s(t) \in W_j \iff s(2^j t) \in W_0$ , lo cual permite concluir:

Si  $\{\psi_{0,n}; n \in \mathbb{Z}\}$  es una base de  $W_0$ , entonces,  $\{\psi_{j,n}; j, n \in \mathbb{Z}\}$  es una base de  $W_j$ .

De la misma manera que se había supuesto en la ecuación 2.11, la *función madre Wavelet*  $\psi(t) \in W_0 \subset V_{-1}$  se puede poner como combinación lineal:

$$\psi(t) = \sqrt{2} \sum_{n=-\infty}^{n=\infty} g_0[n] \varphi(2t - n) \quad (2.24)$$

conocida como *ecuación Wavelet*, con la condición que  $\sum_{n=-\infty}^{\infty} |g_0[n]|^2 < \infty$ .

La ecuación de la Wavelet 2.24, junto con la ecuación de dilatación 2.11, juegan un papel crucial en la relación entre las Wavelets y los bancos de filtros. Así, el filtro paso-bajo  $H_0$  determina las propiedades de la función de escalado  $\varphi(t)$ , mientras que el filtro paso-alto  $G_0$  define la función Wavelet  $\psi(t)$  [78].

### 2.3.3 Cálculo de los coeficientes Wavelet mediante recursión: Transformada Wavelet Discreta

Una función  $s(t)$  definida en  $V_j$  se descompone en la base  $\varphi_{j,n}$  de este subespacio como:

$$s(t) = \sum_n a_{j,n} \varphi_{j,n} \quad (2.25)$$

donde  $n$  indica todos los desplazamientos temporales de la función de escalado  $\varphi_{j,n}$ . Los coeficientes  $a_{j,n}$  se obtienen mediante el producto interno definido en la ecuación 2.5, utilizando  $s(t)$  y  $\varphi_{j,n}$ . Pero debido a la propiedad de multiresolución y suponiendo una descomposición en subespacios ortonormales,  $V_j = V_{j+1} \oplus W_{j+1}$ , la función  $s(t)$  también puede ser representada mediante una combinación de las bases de estos dos últimos subespacios  $\varphi_{j+1,n}$  y  $\psi_{j+1,n}$  [78] [43]. El cambio de base implica el cálculo de los coeficientes,  $a_{j+1,n}$  y  $b_{j+1,n}$ , con los que  $s(t)$  queda representada en esta nueva base:

$$s(t) = \sum_n a_{j+1,n} \varphi_{j+1,n} + \sum_n b_{j+1,n} \psi_{j+1,n} \quad (2.26)$$

Aplicando las ecuación de dilatación 2.11 y de la Wavelet 2.24 vistas en los apartados 2.3.1 y 2.3.2, puede conseguirse una relación entre los coeficientes en una escala de resolución y la siguiente. Para la función de escalado el desarrollo comienza con:

$$\varphi_{j+1,n}(t) = 2^{-(j)/2} \sum_k h_0[k] \varphi(2^{-j}t - 2n - k) \quad (2.27)$$

invirtiendo el índice de los coeficientes de la convolución, la ecuación queda:

$$\varphi_{j,n}(t) = \sum_k h_0[k - 2n] \varphi_{j-1,k}(t) \quad (2.28)$$

Por último, al realizar el producto interior a ambos lados de la igualdad con la función  $s(t)$ , dado que las bases son ortonormales se obtiene la relación entre coeficientes [78], [43]:

$$a_{j+1,n} = \sum_k h_0[k - 2n] a_{j,k} \quad (2.29)$$

Operando de la misma forma para el caso de la ecuación de la Wavelet se obtiene:

$$b_{j+1,n} = \sum_k g_0[k - 2n] a_{j,k} \quad (2.30)$$

Hay que destacar que la convolución se realiza con el índice invertido en  $h_0$  y  $g_0$ , así realmente la operación anterior indica la convolución con  $\tilde{h}_0[k] = h_0[-k]$  y  $\tilde{g}_0[k] =$

$g_0[-k]$ . Además, el término  $(k - 2n)$  indica la eliminación de la mitad de coeficientes mediante diezmado diádico.

De forma recursiva pueden calcularse los coeficientes para  $V_{j+2}$  y  $W_{j+2}$ , a partir de los obtenidos para  $V_{j+1}$ , ya que este último subespacio se puede descomponer como suma directa de los dos anteriores subespacios de menor resolución,  $V_{j+1} = V_{j+2} \oplus W_{j+2}$ . Las ecuaciones 2.30 y 2.29 son las ecuaciones que definen la Transformada Wavelet Discreta (*Discrete Wavelet Transform, DWT*). Mediante dichas ecuaciones se calculan los coeficientes Wavelet  $b_{j,n}$  y los coeficientes de escalado  $a_{j,n}$ , para cada nivel de resolución  $j$  en función de los coeficientes de escalado del nivel anterior. Este cálculo recursivo de los coeficientes es lo que hace la WT rápida [78].

La reconstrucción de los coeficientes con los que la función  $s(t)$  se representa en el subespacio  $V_j$ , a partir de los coeficientes con los que la función queda representada en los subespacios de menor resolución,  $V_j = V_{j+1} \oplus W_{j+1}$ , en los que  $V_j$  se divide de forma ortonormal, también puede realizarse de una forma igualmente eficaz:

$$a_{j,n} = \sum_k h_0[n - 2k]a_{j+1,k} + \sum_k g_0[n - 2k]b_{j+1,k} \quad (2.31)$$

El desarrollo que demuestra la ecuación anterior puede consultarse en [78] [43], y representa la Transformada Wavelet Discreta Inversa (*Inverse Discrete Wavelet Transform, IDWT*).

## 2.4 Bancos de filtros

El filtrado subbanda consiste en descomponer la señal como suma de distintas señales con respuestas espectrales diferentes, lo que se conoce como subbandas. Diferentes técnicas han sido utilizadas para implementar dicho filtrado, entre ellas cabe destacar las técnicas basadas en la pirámide Laplaciana [4].

El punto de conexión entre los bancos de filtros y la WT, aparece mediante las ecuaciones de escalado 2.11 y de la Wavelet 2.24. Estas ecuaciones permiten implementar con una gran eficiencia computacional la WT en un análisis multiresolución. Estas transformadas se pueden realizar a través de bancos de filtros iterados, es decir realimentando una de sus salidas con la entrada, como se ha mostrado en la sección 2.3.3.

Un banco de filtros, como se muestra en la figura 2.3, está constituido por un conjunto de filtros junto con operadores de submuestreo o sobremuestreo, que en el dominio temporal dividen la señal en distintas subseñales, caracterizadas cada una de ellas por una banda de frecuencias. Esto es lo que se conoce comúnmente como descomposición subbanda, siendo el número de subbandas dependiente de la descomposición a realizar.

Figura 2.3: *Descomposición con banco de filtros para señales de una dimensión, donde  $X$  es la señal de entrada,  $h_0(n)$  y  $h_1(n)$  son los filtros paso-bajo de análisis y síntesis,  $g_0(n)$  y  $g_1(n)$  son los filtros paso-alto de análisis y síntesis y  $\hat{X}$  es la señal reconstruida.*

En el caso más general de una representación biortogonal existen dos tipos de filtros, los filtros aplicados al analizar la señal o filtros de análisis y los filtros para reconstruir la señal a partir de las diferentes subbandas o filtros de síntesis. En el caso particular de filtros ortogonales ambos filtros coinciden con una inversión alterna [78]. Si los filtros de análisis y síntesis no coinciden, han de garantizar una reconstrucción perfecta.

Denotaremos como  $h_0[n]$  y  $g_0[n]$  los filtros paso-bajo y paso-alto de análisis y como  $h_1[n]$  y  $g_1[n]$  los filtros paso-bajo y paso-alto de síntesis. A la salida de los filtros de análisis se obtiene el contenido de la señal original repartida en dos subbandas, con aproximadamente la mitad de ancho de banda de la original si se realiza una descomposición diádica.

Figura 2.4: *Espectro del filtro paso-bajo  $H(w)$  y paso-alto  $G(w)$  para multiresolución.*

La operación de submuestreo por 2 es compatible con el hecho de que al dividir el espectro en dos partes, conservando cada una de ellas la mitad de la información, hace que con aproximadamente la mitad de muestras se pueda recuperar el dominio original. No son exactamente la mitad porque los filtros no son ideales y existe, tal como se observa en la figura 2.4, *aliasing* entre las respuestas frecuenciales de  $h_0[n]$  y  $g_0[n]$  que debe ser eliminado en el proceso de síntesis [78]. En el siguiente apartado 2.4.1, se muestran cuales son las condiciones que deben cumplir los filtros de síntesis, para invertir el proceso anulando el *aliasing* y dando lugar a una reconstrucción perfecta.

Si después de aplicar los filtros  $h_0[n]$  y  $g_0[n]$  y diezmar, el vector de salida contiene un número de coeficientes igual al número de muestras de la señal original  $x[n]$ , se dice que la representación obtenida es crítica u ortogonal. Por el contrario, si el número de coeficientes es mayor que el número de muestras de la señal original, se dice que es sobrecompleta y por tanto las funciones base de análisis  $\{\varphi_k[n]\}$  no son linealmente independientes entre sí, habiendo una redundancia implícita en ellas [3], [11].

### 2.4.1 Reconstrucción perfecta

En esta sección se analiza el proceso de filtrado en subbandas mostrado en la figura 2.3. En concreto se estudian los mecanismos para la reconstrucción perfecta de la señal en el dominio de la transformada  $z$ . Mayor detalle se puede encontrar en [78], [86].

Considerando en primer lugar el procesado realizado sobre la señal  $x[n]$  por la convolución con los filtros y el diezmado por 2, llevado a cabo en la fase de análisis del banco de filtros, se obtiene aplicando el *Teorema de Plancherel*:

$$\begin{aligned} (\downarrow 2)H_0(z)X(z) &= \frac{1}{2}(H_0(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + H_0(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})) \\ (\downarrow 2)G_0(z)X(z) &= \frac{1}{2}(G_0(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + G_0(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})) \end{aligned} \quad (2.32)$$

que al sobremuestrear y filtrar con los filtros de síntesis, queda finalmente a la salida del banco de filtros [78]:

$$\begin{aligned} H_1(z)(\uparrow 2)(\downarrow 2)H_0(z)X(z) &= \frac{1}{2}H_1(z)(H_0(z)X(z) + H_0(-z)X(-z)) \\ G_1(z)(\uparrow 2)(\downarrow 2)G_0(z)X(z) &= \frac{1}{2}G_1(z)(G_0(z)X(z) + G_0(-z)X(-z)) \end{aligned} \quad (2.33)$$

de manera que al recomponer la señal, sumando la aproximación ( $H$ ) y el detalle ( $G$ ), la resultante del procesado completo tras el banco de filtros es:

$$\frac{1}{2}X(z)(H_1(z)(H_0(z) + G_1(z)G_0(z)) + \frac{1}{2}X(-z)(H_1(z)H_0(-z) + G_1(z)G_0(-z)) \quad (2.34)$$

de donde se puede extraer, con el objetivo de reconstruir la señal, las condiciones necesarias de distorsión nula y aliasing nulo :

$$\text{Condición de Distorsión nula } H_1(z)H_0(z) + G_1(z)G_0(z) = 2 \quad (2.35)$$

$$\text{Condición de Aliasing nulo: } H_1(z)H_0(-z) + G_1(z)G_0(-z) = 0 \quad (2.36)$$

similares a las obtenidas en la ecuación 2.16.

Esta última condición es muy crítica ya que controla la distorsión no armónica que degrada la reconstrucción de la señal. Sin embargo, la condición de distorsión nula

puede relajarse si se permite un retraso  $l$  de la señal original a la salida del banco de filtros. Esta condición es necesaria a la hora de realizar una implementación real y puede expresarse como:

$$H_1(z)H_0(z) + G_1(z)G_0(z) = 2z^{-l} \quad (2.37)$$

Una posible elección de los filtros anteriores, que cumple la condición 2.36 de aliasing nulo es la siguiente [78]:

$$H_1(z) = G_0(z) , G_1(z) = -H_0(-z)$$

A partir de esta elección quedan por determinar los filtros  $H_0(z)$  y  $G_0(z)$ , para satisfacer la condición definida mediante la ecuación 2.37 que cancela la distorsión. De nuevo es necesario elegir una relación entre ambos filtros, una posible solución en este caso es la relación de inversión alterna [78]:

$$G_0(z) = z^{-(N-1)}H_0(-z^{-1}) \quad (2.39)$$

con lo que la condición de distorsión nula, necesaria para obtener reconstrucción perfecta queda como:

$$H_0(z)H_0(z^{-1}) + G_0(z)G_0(z^{-1}) = 2z^{-l} \quad (2.40)$$

Además, la relación 2.39 impone la ortogonalidad de los coeficientes de ambos filtros, por lo que los bancos de filtros construidos de esta forma son ortogonales [78] [86].

### 2.4.2 Ejemplo de diseño de filtros: caso filtros de Daubechies con 4 coeficientes

En este apartado se va a mostrar la forma en la que se aplican las restricciones 2.35 y 2.36, en el proceso de diseño de los filtros. Para llevar a cabo el proceso se siguen las indicaciones propuestas en [78], basadas en la utilización y descomposición de polinomios trigonométricos, con términos  $e^{-ikx}$ , a partir del polinomio  $P_0(z) = H_1(z)H_0(z)$ . Sobre este polinomio  $P_0(z)$  se imponen las condiciones 2.35 y 2.36 para reconstrucción perfecta, quedando la expresión [78]:

$$P_0(z) - P_0(-z) = 2z^{-l} \quad (2.41)$$

Tras ello, se le insertan un número determinado  $p$  de ceros en  $z = -1$ , para que el filtro paso-bajo  $H_1(e^{j\omega})$ , tenga una respuesta plana alrededor de  $\omega = \pi$  y que impida

tener una respuesta abrupta fuera de su banda pasante. Esto implica que  $P_0(z)$  sea de la forma:

$$P_0(z) = (1 + z^{-1})^{2p} Q(z) \quad (2.42)$$

Por razones de coste computacional, como el número de ceros se relaciona con la longitud de los filtros, se toma  $p = 2$ . Para utilizar el filtro normalizado y centrado, se elige  $P(z) = P_0(z) z^l$  con  $l = 3$ , con lo que:

$$P(z) = z^2(1 + z^{-1})^2(1 + z^{-1})^2 z Q(z) = (1 + z)^2(1 + z^{-1})^2(az + b + cz^{-1}) \quad (2.43)$$

al aplicar las condiciones de reconstrucción perfecta 2.35 y 2.36 queda:

$$P(z) + P(-z) = 2 \quad (2.44)$$

resolviendo el sistema da como solución el polinomio:

$$P(z) = z^3(1 + z^{-1})^4 \left( \frac{-1}{16} + \frac{1}{4}z^{-1} + \frac{-1}{16}z^{-2} \right) \quad (2.45)$$

y despejando  $P_0(z)$ :

$$P_0(z) = \frac{1}{16}(-1 + 9z^{-2} + 16z^{-3} + 9z^{-4} - z^{-6}) \quad (2.46)$$

del cual hay que despejar dos factores por el *lema de Riesz* [78], para obtener los filtros para  $H_0(z)$  y  $H_1(z)$ . Una posible elección de  $H_1(z)$  es:

$$H_1(z) = \frac{1}{4\sqrt{2}}(1 - z^{-1})^2(1 + \sqrt{3} + (1 - \sqrt{3})z) \quad (2.47)$$

conocidos como filtros de Daubechies con 4 coeficientes. Las características más importantes de este banco de filtros son: es ortogonal y tiene respuesta en frecuencia plana en  $w = 0$  y  $w = \pi$  [78]. Además, se puede demostrar [78] que para un número determinado de coeficientes, es el filtro ortogonal que alcanza el número máximo de ceros en las frecuencias  $w = 0$  y  $w = \pi$ . Este banco de filtros se va utilizar en muchos ejemplos a lo largo de la tesis.

## 2.5 Transformadas Wavelet con descomposición irregular: Paquetes de Wavelet

Para la mayoría de señales, la WT posee una localización que es superior a la FT, debido a la descomposición iterativa de la señal original, que se realiza sobre el enrejado diádico de intervalos tiempo-frecuencia mostrado en la figura 2.1. Sin embargo,

esta división del espacio tiempo-frecuencia es fija. La Transformada por Paquetes de Wavelet (*Wavelet Packets Transform, WPT*) permite realizar una división del espacio tiempo-frecuencia adaptativo. La WPT es una generalización de la WT, en la que se descomponen sucesivamente no sólo las subbandas paso-bajo, sino que también se actúa sobre las paso-alto. De esta forma la división del espectro en frecuencias puede adaptarse dependiendo de la señal. El estudio realizado en esta sección, al igual que en secciones anteriores, se centra en descomposiciones diádicas y se basa en [90], [10] y [58].

Como se demostró en la sección 2.3, los subespacios obtenidos al aplicar la WT de forma diádica, se forman mediante la sucesiva división de los subespacios paso-bajo  $V_j \subset L^2$ , los cuales se descomponen como suma directa de dos subespacios con menor resolución  $V_j = V_{j+1} \oplus W_{j+1}$ . La relación entre las bases de los subespacios  $V_{j+1}$  y  $W_{j+1}$  con la base de  $V_j$  viene dada dada por las ecuaciones 2.11 y 2.24. En la siguiente iteración la transformación de la señal continua descomponiendo el subespacio paso-bajo  $V_{j+1} = V_{j+2} \oplus W_{j+2}$ .

En la WPT la descomposición puede realizarse tanto sobre  $V_j$  como sobre  $W_j$ , y se pueden obtener cuatro subespacios ortogonales. Siguiendo la notación dada en [58], en cada nivel de resolución  $j$  la señal puede descomponerse en  $m \in [0, \dots, 2^j - 1]$  subespacios  $\Omega_{j,m}$  construidos mediante las bases  $2^{-j/2}w_m(2^{-j}t - n)$ ,  $j, n \in \mathbb{Z}$ .

Cada subespacio del nivel  $j$ , en el siguiente nivel se divide en dos nuevos subespacios,  $\Omega_{j,m} = \Omega_{j+1,2m} \oplus \Omega_{j+1,2m+1}$ . El índice  $m$  indica todos los subespacios que es posible generar en un nivel de resolución  $j$ ,  $m \in [0, \dots, 2^j - 1]$ , y los subespacios pertenecientes al nivel  $j + 1$  resultantes en la división de  $\Omega_{j,m}$ , estos son  $\Omega_{j+1,2m}$  y  $\Omega_{j+1,2m+1}$ .

Las funciones base de los dos nuevos subespacios  $\Omega_{j+1,2m}$ ,  $\Omega_{j+1,2m+1}$  pueden expresarse a partir de las funciones base de  $\Omega_{j,m}$ . Las ecuaciones que rigen esta relación son equivalentes a las ecuaciones de dilatación 2.11 y de la Wavelet 2.24, pero particularizadas al caso de los subespacios  $\Omega_{j+1,2m}$ ,  $\Omega_{j+1,2m+1}$ :

$$w_{2m}(t) = \sqrt{2} \sum h_o[n]w_m(2t - n) \quad (2.48)$$

$$w_{2m+1}(t) = \sqrt{2} \sum g_o[n]w_m(2t - n) \quad (2.49)$$

En la figura 2.5, se muestran las subbandas que aparecen al descomponer la señal totalmente hasta el tercer nivel de resolución, aplicando los filtros paso-bajo ( $h_0$ ) y paso-alto ( $g_0$ ), sobre la señal de entrada  $X$ . El computo de los coeficientes de una función con respecto a WPT es igual de sencillo que en el caso de la WT, realizándose mediante el esquema de filtrado en subbandas, e iterando sobre las subbandas paso-bajo y las paso-alto.

En una descomposición completa WPT hasta el nivel  $j = J$ , hay  $2^J$  subespacios en el último nivel de resolución,  $\Omega_{j,m}$  con  $m \in [0, \dots, 2^J - 1]$ . Pero por supuesto, no es

Figura 2.5: *Esquema de descomposición de la WPT hasta el tercer nivel de resolución. En la figura  $h_0$  es el filtro paso-bajo y  $g_0$  es el paso-alto,  $j$  es el nivel de resolución y  $m$  el índice de los subespacios dentro de un nivel.*

necesario descomponer siempre todos los subespacios pertenecientes a cada nivel. Por lo tanto, pueden aparecer muchas estructuras distintas, que dan como resultado final diferentes bases con las que representar la señal. Existen múltiples descomposiciones diferentes, dentro del árbol de expansión de la WPT, con las que formar una base ortonormal. En [10] se citan las condiciones que debe cumplir cada descomposición para llegar a ser una base:

- Los subconjuntos que forman los vectores base pueden identificarse con intervalos de la forma:

$$I_{j,m} = [2^{-(j)}m, 2^{-(j)}(m + 1)[, \text{ con } (j \in \mathbb{Z}), (m \in [0, \dots, 2^j - 1]) \quad (2.50)$$

A partir de dichos intervalos, se define una partición  $P$  como un conjunto no vacío y disjunto de intervalos cuya unión es  $[0, 1[$ . Cada intervalo representa un zona del espectro de frecuencias normalizado,  $f/\pi$ , para señales discretas  $[0, 1[$ .

- Cada uno de los subespacios  $\Omega_{j,m}$ , en los que se divide  $L^2(\mathbb{R})$ , está asociado a un intervalo  $I_{j,m}$  de la partición que cubre  $[0, 1[$ . Las bases asociadas a cada subespacio son:

$$2^{-j/2}w_m(2^{-j}t - n), \text{ con } n, j, m \in \mathbb{Z}, \text{ definida sobre } I_{j,m} \in P \quad (2.51)$$

- Cada una de las posibles bases ortonormales corresponde a una partición disjunta  $P$ . El número total de bases distintas es  $2^N$  [10], siendo  $N$  el número de puntos de la señal de entrada.

Para buscar la mejor base con la que representar una señal determinada, se han elaborado algoritmos, que minimizando alguna función de coste aplicada sobre los coeficientes obtenidos en la WPT, determinan el conjunto de subespacios ortogonales que mejor la representan [10]. Estos algoritmos se pueden agrupar dependiendo del dominio en el que intentan optimizar dicha representación:

- Algoritmos basados en tiempo.

En este caso se busca la mejor base intentado salvar el inconveniente que representa el hecho de que la WPT, al igual que la WT, no es invariante a desplazamientos temporales. Para eliminar esta dependencia, se utilizan algoritmos que eligen la mejor base sobre todos los desplazamientos temporales posibles de la señal original [58] [41]. El coste computacional de estos algoritmos es de orden  $O(N \log_2(N))$ , siendo  $N$  el número de puntos de la señal de entrada original discreta.

- Algoritmos basados en frecuencia.

Buscan la mejor base aplicando desplazamientos en frecuencia sobre la señal original. Así, en [35] se introduce un algoritmo de búsqueda, que optimiza los desplazamientos en frecuencia de la señal de entrada. El algoritmo calcula todo el árbol de descomposición, para cada posible desplazamiento, y aplica una función de coste para obtener la mejor base. El principal inconveniente de este algoritmo es su coste computacional de orden  $O(N^2)$ .

## 2.6 Transformadas Wavelet Sobremuestreadas

La Transformada Wavelet Sobremuestreada o no diezmada (*Undecimated Wavelet Transform, UWT*), es una modificación de la WT introducida en [71], también conocida como invariante [58] [41] o estacionaria [52]. La UWT no diezma la señal después del filtrado, por lo que genera descomposiciones altamente redundantes e invariantes a traslaciones, siendo usada en aplicaciones de detección de singularidades [44] y eliminación de ruido [9]. Sus propiedades más importantes son [89]:

- La WT desarrollada hasta un nivel  $j$  requiere que el tamaño del vector original sea un múltiplo entero de  $2^j$ . Por otro lado, la UWT no requiere dicha condición porque las muestras no se diezman.
- La UWT es invariante a traslaciones circulares de los puntos que componen la señal. Si se desplaza el vector original circularmente un cierto número de puntos, los coeficientes de la Wavelet y la función de escalado sufren el mismo desplazamiento.
- No hay una pérdida de la resolución temporal, por lo que puede usarse como una aproximación discreta de las transformadas continuas. Además, los coeficientes de la UWT a distintas escalas pueden alinearse temporalmente, es decir, cada punto siempre queda localizado en la misma posición independientemente del nivel de resolución, ya que no hay diezmo y es posible cancelar la fase introducida por los filtros desplazando los puntos de salida. Este alineado permite comprobar la evolución temporal a distintas escalas de frecuencia [89].

- El gran inconveniente que presenta es el aumento del coste de la transformación, ya que no se diezma la señal a cada paso de la iteración. El coste de una descomposición completa de la UWT es de orden  $O(N(\log_2(N)))$ .
- Las condiciones para que un banco de filtros Sobremuestreado constituya un *tight frame*, consiguiendo una representación con reconstrucción perfecta de la señal, son menos restrictivas que en el caso de muestreo crítico, tal como se recoge en las referencias [11], [3].

Dado que la señal no se diezma, en cada iteración del algoritmo de filtrado, para conseguir subbandas que recojan distintos zonas del espectro de la señal original, el proceso requiere la modificación de los filtros. Inicialmente los filtros en el nivel inicial,  $j = 1$ , denotados como  $g_{0,1}(n)$  y  $h_{0,1}(n)$  coinciden con los filtros utilizados en la WT  $g_0(n)$  y  $h_0(n)$ . La FT de los coeficientes nos da su función de transferencia:

$$G(f) = \sum_{i=0}^{L_g-1} g_0(n)e^{-i2\pi fn} \text{ y } H(f) = \sum_{i=0}^{L_h-1} h_0(n)e^{-i2\pi fn} \quad (2.52)$$

donde  $L_g$  y  $L_h$  es la longitud de cada filtro. En la siguiente iteración los filtros se modifican intercalando ceros entre sus coeficientes [71]:

$$g_{0,2}(n) = [g_0(0), 0, g_0(1), 0, g_0(2), \dots, 0, g_0(L_1 - 1)] \quad (2.53)$$

$$h_{0,2}(n) = [h_0(0), 0, h_0(1), 0, h_0(2), \dots, 0, h_0(L_1 - 1)] \quad (2.54)$$

En esta iteración el espectro de los filtros se modifica y todas las frecuencias se dividen por dos, así la FT de los filtros en este nivel es:  $G(2f)$  y  $H(2f)$ . Si se itera hasta el nivel  $j$ , los coeficientes de los filtros en ese nivel serán los del nivel 1 pero con  $2^{j-1} - 1$  ceros intercalados:

$$g_{0,j}(n) = [g_0(0), \underbrace{0, \dots, 0}_{2^{j-1}-1}, g_0(1), 0, \dots, 0, g_0(L_1 - 1)] \quad (2.55)$$

$$h_{0,j}(n) = [h_0(0), \underbrace{0, \dots, 0}_{2^{j-1}-1}, h_0(1), 0, \dots, 0, h_0(L_1 - 1)] \quad (2.56)$$

siendo la longitud de los filtros en el nivel  $j$ ,  $L_j = (2^j - 1)(L_1 - 1) + 1$ . La FT de estos filtros es  $G(2^{(j-1)}f)$  y  $H(2^{(j-1)}f)$ . En la figura 2.6 se muestra el espectro de los filtros paso-bajo y paso-alto, para los 4 primeros niveles utilizando los filtros de Daubechies con 4 coeficientes, dados en la sección 2.4.2.

Del mismo modo que se define la WPT a partir de la WT, se puede desarrollar la Transformada por Paquetes Wavelet Sobremuestreada (*Undecimated Wavelet Packet*

Figura 2.6: *Espectro de los filtros para los cuatro primeros niveles, utilizando los filtros de Daubechies 4 definidos en la sección 2.4.2. De izquierda a derecha y de arriba a abajo se muestra el primer nivel,  $j = 1$ , con los filtros  $h_{0,1}(n)$  y  $g_{0,1}(n)$ , el segundo nivel,  $j = 2$ , con  $h_{0,2}(n)$  y  $g_{0,2}(n)$ , el tercer nivel,  $j = 3$ , con  $h_{0,3}(n)$  y  $g_{0,4}(n)$  y el cuarto nivel,  $j = 4$ , con  $h_{0,4}(n)$  y  $g_{0,4}(n)$*

*Transform, UWPT*) [89]. Los coeficientes, obtenidos por la transformación de la señal de entrada, en cada nivel  $j$  de resolución se definen como:

$$W_{j,m} = \{w_{j,m}(n), n = 0, \dots, N - 1\} \quad (2.57)$$

donde  $j$  es el nivel de resolución,  $m \in [0, \dots, 2^{j-1}]$  es el índice del intervalo de frecuencias en el nivel  $j$  y  $N$  es el número de puntos que contiene el vector original.

El espacio de frecuencias normalizadas,  $f/\pi$ , para señales discretas,  $[0, 1[$ , se divide en intervalos  $I_{j,m} = [2^{-(j)}m, 2^{-(j)}(m + 1)[$ , cada uno de los cuales está asociado con el subespacio de índices  $(j, m)$ , tal como se introdujo en 2.50. La aplicación de los filtros necesaria para obtener cada una de las subbandas se representa en la figura 2.7.

Para obtener los coeficientes de una subbanda en un nivel  $j$ , a partir de los del nivel  $j - 1$  se aplican los filtros de análisis adaptados a dicho nivel. Ya que los únicos coeficientes que no son cero en los filtros son los  $g_0(n)$  y  $h_0(n)$ , la convolución se puede simplificar mediante la siguiente expresión [89]:

$$w_{j,m}(n) = \sum_{l=0}^{L_1-1} f_{m,l} w_{j-1, \lfloor m/2 \rfloor}((n - 2^{j-1}l) \bmod(N)), \quad n = 0, \dots, N - 1,$$

Figura 2.7: *Esquema de subbandas obtenidas en UWPT, con niveles  $j \in [0, \dots, 4]$ .*

donde:

$$f_{m,l} = \begin{cases} h_0(l) & \text{si } (m) \bmod(4) \in [0, 3] \\ g_0(l) & \text{si } (m) \bmod(4) \in [1, 2] \end{cases} \quad (2.58)$$

En la figura 2.8 se muestra el espectro ocupado por la subbanda  $W_{4,0}$  resultante al aplicar los filtros  $h_{0,1}(n)$ ,  $h_{0,2}(n)$ ,  $h_{0,3}(n)$  y  $h_{0,4}(n)$  de forma iterativa sobre la señal original.

Dos propiedades importantes de la UWPT son:

La primera propiedad de la UWPT es la conservación de la energía en cada nivel  $j$ . Si la energía de la señal inicial es  $\|X\|^2 = \sum_{i=0}^{N-1} \|x(n)\|^2$ , en cualquier otro nivel  $j$  se cumple [89]:

$$\|X\|^2 = \sum_{m=0}^{2^j-1} \sum_{i=0}^{N-1} \|w_{j,m}(n)\|^2 \quad (2.59)$$

La segunda es la reconstrucción perfecta. Este proceso se realiza mediante la convolución con los filtros de síntesis,  $h_{1,j}(n)$  y  $g_{1,j}(n)$ , sobre las subbandas paso-alto y paso-bajo correspondientes. Posteriormente se suman punto a punto los vectores obtenidos tras dicha convolución.

Figura 2.8: *Espectro de la subbanda  $W_{4,0}$ , obtenida tras aplicar  $h_{0,1}(n), h_{0,2}(n), h_{0,3}(n)$  y  $h_{0,4}(n)$ .*

La ecuación que define la reconstrucción a partir de las subbandas pertenecientes a un mismo nivel  $j$  es [89]:

$$w_{j-1,m}(n) = \sum_{s=2m}^{2m+1} \sum_{l=0}^{L_1-1} \tilde{f}_{s,l} w_{j,s}((n - 2^{j-1}l) \bmod(N)), \quad n = 0, \dots, N - 1$$

donde  $\tilde{f}_{s,l}$  representa los filtros aplicados en el proceso de síntesis, para obtener la coeficientes de la subbanda  $w_{j-1,m}$  a partir de las subbandas  $w_{j,2m}$  y  $w_{j,2m+1}$  pertenecientes al nivel  $j$ . Analíticamente este término se representa como:

$$\tilde{f}_{m,l} = \begin{cases} h_1(l) & \text{si } (m) \bmod(4) \in [0, 3] \\ g_1(l) & \text{si } (m) \bmod(4) \in [1, 2] \end{cases} \quad (2.60)$$

siendo  $h_1(n)$  y  $g_1(n)$  los correspondientes filtros de síntesis en la WT.

## 2.7 Esquemas eficientes para el cálculo de la Transformada Wavelet

El cálculo de la WT, a partir de las fórmulas mostradas en la sección 2.3.3, representa la aplicación directa del algoritmo piramidal de Mallat [43]. Esta implementación de la WT permite de forma rápida, calcular los coeficientes Wavelet de un nivel de

resolución, a partir de los del nivel anterior. En la bibliografía dicho algoritmo se denomina *implementación en forma directa* [5], siendo equivalente a la aplicación de un banco de filtros mediante convoluciones. Pero esta estructura no es la mejor forma de calcular la transformada, porque el diezmado de la señal se realiza después de calcular todos los coeficientes. En esta sección, se introducen dos formas más eficientes de calcular la WT, que aprovechan el diezmado para reducir la carga computacional.

### 2.7.1 Cálculo mediante matriz Polifásica

La forma Polifásica de un filtro, representado en el dominio de la transformada  $z$ , separa el filtro en dos componentes. Uno de estos componentes contiene los coeficientes con potencias pares en  $z$  y el otro los de potencias impares [78]:

$$H_0(z) = H_{0,0}(z^2) + z^{-1}H_{0,1}(z^2) \quad (2.61)$$

siendo  $H_{0,0}(z^2)$  la matriz, proveniente de los coeficientes de potencias pares con ceros intercalados en las posiciones correspondientes a potencias impares

$$(h_0(0), 0, h_0(2), 0, h_0(4), \dots)$$

y  $H_{0,1}(z^2)$  la matriz contraria, que contiene ceros en las posiciones de potencias pares y coeficientes en las impares. La representación de la señal de entrada,  $X(z)$ , también se puede hacer de forma Polifásica, siendo equivalente a la mostrada en la ecuación 2.61.

El filtrado se realiza mediante la operación  $H_0(z)X(z)$ . Pero el diezmado elimina todos los coeficientes situados en las potencias pares o en las impares resultantes. Suponiendo que se eliminan las potencias impares, en el vector resultante sólo permanecen las pares. Estas potencias pares provienen de la multiplicación de potencias pares con pares o impares con impares de los términos  $H_0(z)$  y  $X(z)$ .

$$((\downarrow 2)H_0X(z^2)) = H_{0,0}(z^2)X_0(z^2) + z^{-2}H_{0,1}(z^2)X_1(z^2) \quad (2.63)$$

A continuación se diezman las ceros, que quedaban situados en las posiciones correspondientes a potencias impares y convierten  $z^2$  en  $z$ . Así se obtiene [78]:

$$((\downarrow 2)H_0X(z)) = H_{0,0}(z)X_0(z) + z^{-1}H_{0,1}(z)X_1(z) \quad (2.64)$$

donde  $H_{0,0}(z)$  y  $X_0(z)$  son los coeficientes provenientes de potencias pares, del filtro paso-bajo y de la señal respectivamente. Mientras que,  $H_{0,1}(z)$  y  $X_1(z)$  son los procedentes de las impares. Hay que destacar que el componente  $H_{0,0}(z^2)$  no es igual a

$H_{0,0}(z)$ , la diferencia se debe a que el primero de ellos contiene ceros en las posiciones de las potencias pares, mientras que en el segundo se han eliminado tras el diezmado. Esto mismo sucede con los vectores  $X_0(z)$  y  $X_0(z^2)$ .

La forma más eficiente de implementar la ecuación 2.64, implica el diezmado inicial de la señal y el posterior filtrado de los coeficientes pares e impares en paralelo. En forma de matriz el banco de filtros de análisis queda de la forma [78]:

$$V = \begin{bmatrix} (\downarrow 2)H_0(z)X(z) \\ (\downarrow 2)G_0(z)X(z) \end{bmatrix} = \begin{bmatrix} H_{0,0}(z)H_{0,1}(z) \\ G_{0,0}(z)G_{0,1}(z) \end{bmatrix} \cdot \begin{bmatrix} X_0(z) \\ z^{-1}X_1(z) \end{bmatrix} = H_p(z) \begin{bmatrix} X_0(z) \\ z^{-1}X_1(z) \end{bmatrix} \quad (2.65)$$

donde  $H_p(z)$  es la matriz Polifásica de análisis:

$$H_p(z) = \begin{bmatrix} H_{0,0}(z)H_{0,1}(z) \\ G_{0,0}(z)G_{0,1}(z) \end{bmatrix} \quad (2.66)$$

De la misma forma se puede definir la matriz Polifásica para los filtros de síntesis  $F_p(z)$  [78], con lo que la implementación del banco de filtros queda tal como se muestra en la figura 2.9.

Figura 2.9: banco de filtros con implementación Polifásica. Donde  $X$  es el vector de entrada y  $\hat{X}$  es el vector reconstruido,  $H_p(z)$  es la matriz Polifásica de análisis y  $F_p(z)$  es la matriz Polifásica de síntesis.

Esta implementación Polifásica del banco de filtros, consigue reducir el número de operaciones necesarias para realizar el filtrado, debido a que el diezmado se realiza antes. De esta manera, el número total de operaciones, tanto multiplicaciones como sumas, es la mitad que en la forma directa [78].

### 2.7.2 Cálculo mediante estructura *Lattice*

Una estructura de filtrado *Lattice* [61] posibilita la implementación de filtros FIR o IIR, de forma jerárquica. A modo de ejemplo, la estructura *Lattice* mostrada en la figura 2.10 permite la implementación de un filtro de orden  $N - 1$ . Además, esta construcción es modular, ya que se puede componer un filtro de orden mayor añadiendo

más secciones. Por otro lado, ya que muchos de los filtros utilizados en la implementación de la WT son filtros FIR, resulta interesante considerar las condiciones bajo las cuales, un banco de filtros puede representarse mediante una estructura *Lattice*, como la mostrada en la la figura 2.10.

Figura 2.10: *Estructura Lattice para un banco de filtros, donde  $K_0, \dots, K_{N-1}$  son los coeficientes que definen la estructura,  $P$  y  $Q$  son las salidas de cada una de las secciones y  $z^{-1}$  representa los elementos de retraso.*

La relación entre dos salidas de la estructura,  $P_m$  y  $Q_m$ , siendo  $m \in [1, \dots, N-1]$ , en función de los términos a la salida de la sección anterior es:

$$P_m(z) = P_{m-1}(z) - K_m z^{-1} Q_{m-1}(z) \quad (2.67)$$

$$Q_m(z) = K_m P_{m-1}(z) + z^{-1} Q_{m-1}(z)$$

De forma recursiva se pueden calcular las salidas para todas las demás etapas.

Tal como se demuestra en [85], un banco de filtros con  $N$  coeficientes implementado mediante la estructura *Lattice* de la figura 2.10, puede cumplir las siguientes condiciones de reconstrucción perfecta, dadas en la sección 2.4.1:

$$Q_{N-1}(z) = z^{-(N-1)} P_{N-1}(-z^{-1}) \quad (2.69)$$

$$Q_{N-1}(z)Q_{N-1}(z^{-1}) + P_{N-1}(z)P_{N-1}(z^{-1}) = 2z^{-l}$$

Las restricciones que debe satisfacer la estructura, para cumplir las condiciones de reconstrucción perfecta son [85]:

1. Los coeficientes deben elegirse de la siguiente forma:  $K_0 = 1/\sqrt{2}$ ,  $K_{2m} = 0$  para  $m > 0$ .
2.  $N - 1$  debe ser impar.

Además, la relación funciona en ambos sentidos, es decir, si se considera un banco de filtros de orden  $N - 1$ , que cumple las condiciones de reconstrucción perfecta 2.69 anteriores, su implementación mediante la estructura *Lattice* tendrá todos los coeficientes pares nulos, excepto  $K_0$ . Por otro lado, tal como se demuestra en [86], al cumplir la relación 2.69 el banco de filtros es ortogonal. Por lo tanto, sólo es posible implementar bancos de filtros ortogonales.

La condición anterior, que impone la nulidad de todos los coeficientes pares, modifica la estructura del banco de filtros, tal como se muestra en la figura 2.11.

Figura 2.11: *Estructura Lattice para un banco de filtros ortogonal con reconstrucción perfecta, donde  $K_0, \dots, K_{N-1}$  son los coeficientes que definen la estructura,  $P$  y  $Q$  son las salidas de cada una de las secciones y  $z^{-1}$  representa los elementos de retraso. Además,  $E(z^2)$  es la matriz definida como combinación de las secciones de la estructura, contenidas en el cuadrado definido en la figura.*

donde  $E(z^2)$ , que se ha definido en la figura anterior 2.11, se puede expresar de forma matemática como:

$$E(z^2) = 1/\sqrt{2} \begin{bmatrix} 1 & -K_{N-1}(z^{-2}) \\ K_{N-1} & (z^{-2}) \end{bmatrix} \begin{bmatrix} 1 & -K_3(z^{-2}) \\ K_3 & (z^{-2}) \end{bmatrix} \cdots \begin{bmatrix} 1 & -K_1 \\ K_1 & 1 \end{bmatrix} \quad (2.71)$$

de igual manera se define  $R(z^2)$ , matriz inversa a  $E(z^2)$ , cuya estructura se puede consultar en [85].

La dos matrices anteriores,  $E(z^2)$  y  $R(z^2)$ , constituyen el banco de análisis y síntesis respectivamente. Además, pueden expresarse en  $z$ , si se lleva a cabo el diezmado de la señal antes del filtrado. Este cambio de variable se realiza modificando los elementos de retardo, para que retrasen la señal sólo una posición. El banco de filtros obtenido tras el intercambio del diezmado es el que se muestra en 2.12.

Otra ventaja destacable de esta implementación de la WT es su robustez frente a

Figura 2.12: *Esquema del banco de filtros obtenido en la implementación Lattice, definido mediante la matriz de análisis  $E(z)$  y la de síntesis  $R(z)$ .*

la cuantificación de los coeficientes, ya que el redondeo de estos valores no afecta al cumplimiento de las condiciones 2.69, que impone la reconstrucción perfecta [85].

## 2.8 Conclusiones

La teoría de las Wavelets puede ser vista como un marco común, que contiene diversas técnicas que han sido desarrolladas de forma independiente. De esta manera, los conceptos de banco de filtros, codificación subbanda, descomposición multiresolución o la teoría de representación de señales continuas en *frames* pertenecientes a un espacio de Hilbert, se engloban y complementan en un marco común, que comunica la matemática aplicada con las aplicaciones en ingeniería.

Actualmente la WT, con sus variantes WPT y UWPT, se ha convertido en una herramienta, usada de forma exitosa en multitud de aplicaciones de procesamiento de la señal. Una de sus aplicaciones en el campo de las comunicaciones, como se ve en el próximo capítulo 3, es la supresión de interferencias superpuestas a señales de comunicación del tipo de Espectro Ensanchado. En este campo, la transformada juega un papel clave a la hora de diferenciar entre el ruido y la señal transmitida. En el presente trabajo de investigación, la WT se utiliza como una herramienta con la que profundizar en los métodos de supresión de interferencias, que permita la transmisión de señales de comunicación en entornos con altos niveles de interferencias.

Otro punto importante, en el que se centra el trabajo aportado, es la implementación de la transformada sobre dispositivos de computación. Tal como se ha definido en la sección 2.7, existen varias estructuras que permiten la implementación de la WT, en concreto se han considerado tres formas. La primera mediante la aplicación del algoritmo de Mallat, designada como forma directa. A continuación, se ha presentado la implementación Polifásica que permite un ahorro computacional al diezmar la señal antes de filtrar. Y por último, la estructura *Lattice* que también es una implementación eficiente y permite implementar bancos de filtros ortogonales [61].

Para comparar los tres métodos en función de su coste computacional, se puede considerar el número de multiplicaciones necesario para descomponer la señal de entrada

## 2.8. Conclusiones

---

en dos subbandas al aplicar la WT. Así, para un filtro con  $N$  coeficientes, el número de multiplicaciones por punto del vector de entrada es:  $2N$  para la forma directa,  $N$  para la implementación Polifásica y  $N + 1$  con estructura *Lattice*. Por lo tanto, las implementaciones mediante estructura Polifásica y *lattice* prácticamente tienen el mismo coste.

# Capítulo 3

## Comunicaciones con modulaciones de Espectro Ensanchado

### 3.1 Introducción

El término modulaciones de Espectro Ensanchado (*Spread-Spectrum, SS*) ha sido usado desde los años 50, para describir modulaciones digitales en las cuales el ancho de banda ocupado es mucho mayor, que la velocidad de transferencia de información. Esta redundancia hace que las señales SS presenten ciertas propiedades de inmunidad frente a interferencias que las hacen apropiadas para su utilización en canales con altos niveles de ruido, como por ejemplo en comunicaciones militares, transmisiones vía satélite, comunicaciones móviles y comunicaciones por cable de potencia [62].

La expansión del ancho de banda de la señal se realiza normalmente mediante algún código pseudo-aleatorio, también denominado como pseudo-ruido (*Pseudo-Noise, PN*) [59], sólo conocido por el emisor y el receptor, lo que las hace muy difíciles de interceptar por receptores no autorizados. Además, este grado de pseudo-aleatoriedad en la forma de onda de la señal transmitida combate posibles interferencias intencionadas (*jamming*) que traten de corromper los símbolos transmitidos. De forma resumida, algunas de las características más importantes que debe presentar un código PN son:

- No debe estar correlacionado. Su función de autocorrelación, de forma ideal, debe ser  $\phi(0) = n$  y  $\phi(j) = 0$ , para  $1 \leq j \leq n - 1$ , siendo  $n$  el número de elementos de la secuencia.
- Para evitar escuchas o interferencias intencionadas, el periodo del código debe ser lo suficientemente grande, como para que resulte muy complicado por un receptor no autorizado su aprendizaje.

### 3.2. Señales moduladas con Espectro Ensanchado

---

- En aplicaciones del tipo Acceso Múltiple por División de Código (*Code Division Multiple Access, CDMA*) [24], las secuencias PN utilizados por distintos usuarios no deben estar correlacionadas. Los códigos que cumplen esta propiedad se les denomina ortogonales.

Así, las propiedades básicas de las modulaciones SS son:

- Alto grado de inmunidad frente a interferencias intencionada, interferencia producida por otros usuarios del canal o interferencia autoinducida debida a multitrayecto (*multipath*).
- Baja potencia de su densidad espectral, lo que les hace difíciles de detectar por otros usuarios en presencia de ruido de fondo.
- Permiten comunicaciones seguras debido a la codificación pseudo-aleatoria.

Estas características de las señales SS permitieron que sus primeras aplicaciones aparecieran en el campo de las comunicaciones militares, donde la confidencialidad e inmunidad frente a interferencias son cuestiones muy importantes. Posteriormente, la necesidad de combatir las interferencias autoinducidas debidas a multitrayecto en comunicaciones móviles, sirvió de comienzo para ampliar su campo de aplicación hacia otros sistemas, incluyendo sistemas comerciales [69].

Este capítulo incluye en la sección 3.2, un breve repaso de la características de las modulaciones SS. La siguiente sección 3.3 estudia de forma teórica, las prestaciones de

Figura 3.1: *Esquema del sistema de comunicación utilizando técnicas SS.*

Esta modulación determina la forma más adecuada de expandir el espectro, dando como resultado diferentes técnicas SS.

En el primer tipo de modulaciones, la información se codifica mediante cambios en la fase de la portadora. Un ejemplo dentro de este grupo es la modulación por Desplazamientos de Fase (*Phase Shift Keyed, PSK*). Para que el receptor pueda recuperar la información debe estar sincronizado en fase con el emisor. Este caso suele dar lugar a señales SS, en las que la expansión del ancho de banda se realiza introduciendo varios cambios de fase por símbolo transmitido, denominándose modulación de Espectro Ensanchado en Secuencia Directa (*Direct-Sequence Spread-Spectrum, DS-SS*).

Por el contrario, cuando la información se codifica mediante cambios en la frecuencia de la portadora, se obtienen modulaciones con Desplazamientos en Frecuencia (*Frequency Shift Keyed, FSK*). En este caso, la expansión del ancho de banda ocupado se lleva a cabo mediante cambios pseudo-aleatorios en la frecuencia de la portadora. El resultado son las técnicas denominadas Espectro Ensanchado con Saltos de Frecuencia (*Frequency-Hopping Spread-Spectrum, FH-SS*).

Aunque las dos técnicas anteriores son las más usuales, existen otros tipos de modulaciones SS, como las híbridas DS/FH o las de barrido en frecuencia comentadas brevemente en los apartados siguientes.

### 3.2.1 Espectro Ensanchado en Secuencia Directa

La técnica DS-SS se suele combinar con la modulación PSK. Esta modulación se usa en aplicaciones donde es posible mantener la coherencia de fase entre emisor y receptor. Para expandir el espectro, la secuencia PN divide los símbolos PSK en elementos de menor periodo y varía su fase de forma pseudo-aleatoria. El proceso de modulación comienza considerando una secuencia de bits, con una velocidad de transmisión  $R = 1/T_b$  bps, siendo  $T_b$  el periodo de transmisión de un bit. Si se supone modulación PSK binaria, cada bit se corresponde con un símbolo transmitido. Pero la expansión DS modifica la secuencia de bits mediante el codificador del canal, tal como muestra la figura 3.1, utilizando un código en bloque lineal binario  $(n, k)$  o

convolucional binario  $k/n$  [59]. Estos códigos a partir de  $k$  bits generan  $n$  elementos binarios. A continuación, se aplica el código PN sobre los  $n$  elementos que forman la secuencia. Dicha aplicación se puede llevar a cabo mediante multiplicaciones, si los valores que presentan ambos códigos están en el alfabeto  $\{+1, -1\}$ , como se muestra en la figura 3.2. Por el contrario, si los elementos de los códigos están en el alfabeto  $\{1, 0\}$ , los cambios de fase se realizan mediante suma modulo-2.

Los elementos en los que se divide un bit después de la expansión se denominan *chips*. Cada uno de ellos tiene un periodo  $T_c$  y la velocidad de transmisión es  $V = 1/T_c$ . El número de cambios de fase por bit es  $L = n/k$ , valor que normalmente debe ser entero y grande. De esta forma se tiene que  $T_b \gg T_c$ , lo que produce una expansión del ancho de banda ocupado por la señal, utilizando una zona del espectro mucho mayor para transmitir la misma cantidad de información. El factor que mide el grado de expansión del ancho de banda,  $B$ , se define como:

$$B = \frac{T_b}{T_c} = \frac{V}{R} \quad (3.1)$$

Figura 3.2: *Ejemplo de aplicación mediante multiplicaciones, de la secuencia PN sobre los bits transmitidos en el alfabeto  $\{+1, -1\}$ . La primera línea representa los bits transmitidos, la segunda la secuencia PN y la tercera la secuencia de chips.*

Después de realizar la expansión con DS-SS, el espectro de la señal original se modifica, obteniendo un espectro expandido como se ve en la figura 3.3. El lóbulo principal de este espectro ocupa un ancho de banda  $2R_c$ , siendo  $R_c = 1/T_c$  la tasa binaria de cambios de fase por segundo.

### 3.2.2 Espectro Ensanchado con Saltos de Frecuencia

La modulación FSK es apropiada en aplicaciones donde la coherencia de fase no puede ser mantenida, debido por ejemplo a variaciones temporales en las características del canal. En este caso se simplifica el esquema del receptor al no ser necesaria la sincronización de fase con el emisor, pero en general se obtienen peores rendimientos

Figura 3.3: *Espectro de la señal DS-SS expandida, junto con la señal PSK original. Donde  $2R$  y  $2R_c$  representan la amplitud del lóbulo principal de la señal antes y después de la expansión.*

que en el caso anterior, ya que la modulación PSK da mayores rendimientos que la FSK en canales con Ruido Blanco Gaussiano Aditivo (*Additive White Gaussian Noise, AWGN*), como se demuestra en [59].

La secuencia PN se encarga de expandir el espectro de la señal original FSK, mediante el desplazamiento de la frecuencia de la portadora sobre un rango amplio de frecuencias de forma pseudo-aleatoria, como se muestra en la figura 3.4. Aunque la modulación FSK es la más utilizada, en algunos casos las señales FH-SS utilizan otras modulaciones, como la FSK de Fase Continua (*Continuous Phase Frequency Shift Keyed, CPFSK*) y la PSK Diferencial (*Differential Phase Shift Keyed, DPSK*) [59].

Figura 3.4: *Ejemplo de distribución de frecuencias para FH-SS en función del tiempo, los saltos de frecuencia entre chips están dictados por el código PN. El valor  $T_c$  corresponde con el periodo de un chip generado por la secuencia PN.*

Cada uno de los intervalos de tiempo  $T_c$  en los cuales se transmite una frecuencia con valor dado por la secuencia PN, se denomina *chip* como en el caso DS. Cada una de las posibles posiciones en frecuencia que puede ocupar la portadora, después de ser desplazada se denomina canal.

Cuando las frecuencias utilizadas en la expansión FH-SS están situadas en in-

tervalos fijos, con un desplazamiento entre ellas  $1/T_c$ , el conjunto de portadoras es ortonormal. Esta distribución de frecuencias se denomina *block hopping* [59]. El periodo  $T_c$  puede coincidir con el periodo de un símbolo de la señal FSK, habiendo un único desplazamiento en frecuencia por símbolo transmitido. Este tipo de modulación se conoce como de Salto Lento (*Slow-Hopping, SFH-SS*). Por el contrario, cuando hay varios desplazamientos por símbolo, se denomina de Salto Rápido (*Fast-Hopping, FFH-SS*). En general, este segundo caso aunque es más inmune a interferencias intencionadas, que repiten la última frecuencia transmitida, o a interferencias de banda estrecha, tiene peor rendimiento en canales con AWGN debido a la combinación no-coherente. Esta combinación lineal suma la energía de varios *chips*, después de realizar su recepción de forma no-coherente, para completar la energía de un bit [59].

Aunque la modulación FH-SS expande el rango de frecuencias usado por la señal original, el ancho de banda ocupado en cada instante es pequeño si se compara con la expansión producida en DS-SS. De esta forma, una interferencia de banda estrecha formada por un Onda con Tono Continuo, (*Continuous Wave Tone, CWT*), puede corromper una frecuencia particular en un determinado instante de tiempo, pero los demás canales de frecuencia quedan limpios. Esta propiedad hace que los sistemas FH-SS sean resistentes a CWT, aunque son más susceptibles frente a interferencias de banda ancha que afecten a múltiples frecuencias [62].

### 3.2.3 Otros tipos de modulaciones de Espectro Ensanchado

Aparte de los dos métodos de comunicación SS ya mostrados existen otras variantes, entre las que se encuentran las técnicas de expansión que combinan DS con FH y las modulaciones por barrido en frecuencia. Las características básicas de cada una de ellas son:

#### Métodos híbridos DS/FH

Una de las ventajas de los sistemas FH es que la energía de la señal, puede expandirse sobre todo el rango de frecuencias deseado. Este no es el caso en la modulación DS, donde la distribución de energía en frecuencias exhibe la forma típica  $\text{sinc}(x)$ . Esta forma del espectro en modulaciones DS no permite utilizar por igual todo el ancho de banda disponible. Además, impone restricciones sobre los filtros de entrada y salida, necesarios para eliminar los lóbulos no deseados [62].

Los métodos híbridos permiten combinar las ventajas de DS y FH. La combinación de las dos técnicas se realiza modulando primero la señal como DS, y a continuación variando la frecuencia de la portadora de forma pseudo-aleatoria con FH. Con esta combinación se consigue expandir la señal sobre todo el ancho de banda disponible, aunque se complica el diseño del emisor y el receptor.

## Barrido en frecuencia

Este tipo de modulación SS fue desarrollada en principio para su utilización en sistemas de radar, aunque se ha adaptado a otras aplicaciones, como es el caso del estándar CEBus para la transmisión de información por el cable de potencia. La idea es transmitir una señal con envolvente constante y barrido en frecuencia lineal denominada *chirp*. El receptor en este caso es parecido al utilizado en DS-SS, siendo necesaria la coherencia de fase entre emisor y receptor. En la entrada se sitúa un filtro adaptado que comprime la energía de la señal en el dominio temporal [62].

### 3.3 Estudio del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia no-coherente

En esta sección se examina el receptor FH-SS no-coherente, como continuación en el estudio de las modulaciones SS. En concreto se considera primero el esquema físico, con el que implementar el receptor, a partir del cual se pueden examinar las prestaciones del sistema de comunicación. Estas prestaciones se miden mediante la Tasa de Bits Erróneos (*Bit Error Rate, BER*), considerando diversas situaciones caracterizadas por el tipo de ruido o interferencia, que afecte a la señal de comunicación. En primer lugar, se trata el caso con ruido AWGN, que es el ruido más habitual en cualquier canal de comunicación. Posteriormente se examinan los casos de canales con interferencias del tipo: CWT y ruido gaussiano de banda estrecha. Los resultados obtenidos en esta sección, junto con las carencias encontradas, van a servir como punto de partida del trabajo de investigación realizado.

Para comenzar la presentación, se analiza el esquema del receptor FSK Binario (*Binary-FSK, BFSK*) con detección no-coherente. Este receptor es la base, sobre el que posteriormente se introducen las modificaciones necesarias, para aplicar la expansión de la señal en las modulaciones SS. Al insertar dicha expansión, el rendimiento del receptor varía de forma sensible. También se va a llevar a cabo la distinción de los resultados en función del tipo de expansión realizado: SFH-SS o FFH-SS.

En BFSK se utilizan dos frecuencias ortonormales para enviar un dato binario, cada bit transmitido se codifica en un símbolo que matemáticamente se expresa como:

$$s(t) = \sqrt{2S} \cos(2\pi(b_i/T_b + f_c)t + \phi_s) \quad (3.2)$$

donde  $S$  es la potencia de la señal,  $T_b$  es el periodo de un símbolo,  $f_c$  es la frecuencia de la portadora,  $1/T_b$  es la separación entre las dos posibles frecuencias usadas por

### 3.3. Estudio del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia no-coherente

---

la BFSK, y  $\phi_s$  es la fase introducida por el modulador, la cual se supone uniformemente distribuida en  $[0, 2\pi)$ . Los símbolos transmitidos se eligen mediante el valor correspondiente de  $b_i$  elegido entre:  $\{+1, 0\}$ .

El esquema del receptor BFSK no-coherente se muestra en la figura 3.3. El receptor posee dos correladores en cuadratura sintonizados a cada una de las posibles frecuencias utilizadas,  $f_0 = f_c$  y  $f_1 = f_c + 1/T_b$ . La salida de los correladores pasa a través de un filtro paso-bajo, y a continuación se integra durante el periodo de un símbolo. Por último, las dos componentes en cuadratura se combinan de forma cuadrática para eliminar la dependencia con la fase. La entrada al detector está formada por los valores provenientes de cada una de las dos frecuencias utilizadas. El receptor decide cuál ha sido el símbolo transmitido, comparando los dos valores y seleccionando el mayor de ellos.

Figura 3.5: *Esquema del receptor BFSK no-coherente. La señal  $r(t)$  se correlaciona en cuadratura con las frecuencias  $f_0$  y  $f_1$ , posteriormente se aplica un filtro paso-bajo y se combina la energía de ambos correladores de forma no-coherente, para eliminar la dependencia con la fase  $\phi_s$ . La variable de decisión resultante con mayor energía,  $|U_1|^2$  ó  $|U_2|^2$ , indica el bit transmitido.*

En el siguiente análisis de las prestaciones del receptor, se supone siempre que la modulación a nivel de bit es BFSK no-coherente. Además, el receptor visto en la figura 3.3, se modifica para incluir las variantes con las expansiones SFH-SS y FFH-SS.

#### 3.3.1 Recepción en canales con ruido aditivo blanco gaussiano

Los canales más habituales son los que contienen únicamente ruido AWGN, por lo que constituye un primer paso en el estudio del comportamiento del receptor. Para el

caso de modulación BFSK sin expansión SS, suponiendo que la señal transmitida se ve afectada sólo por AWGN, la entrada al receptor es:

$$r(t) = s(t) + n(t) \quad (3.3)$$

donde  $s(t)$  es el símbolo transmitido en BFSK, expresado matemáticamente en la ecuación 3.2, y  $n(t)$  es AWGN con densidad espectral de potencia (*Spectral Power Density, SPD*)  $J_o$ . Bajo estas condiciones el sistema tiene una cierta probabilidad de error en los bits transmitidos, dada por la siguiente ecuación cuyo desarrollo puede consultarse en [59]:

$$P(e) = \frac{1}{2} e^{-\frac{\gamma_b}{2}} \quad (3.4)$$

donde  $\gamma_b = \xi_b/J_o$  es la Relación Señal a Ruido (*Signal Noise Ratio, SNR*) por bit, y  $\xi_b$  es la energía de un símbolo.

Una vez visto el caso BFSK, a continuación se considera la recepción de señales expandidas mediante SFH-SS y FFH-SS.

Un sistema de comunicación empleando modulación SFH-SS basada en BFSK, con AWGN se comporta de manera similar a un sistema con sólo modulación BFSK, sin SS [59]. De esta forma, la probabilidad de error es la misma que la expresada en la ecuación 3.4. El esquema del receptor sólo varía en el número de correladores, que ahora se eleva hasta el número total de frecuencias utilizadas por la expansión SFH-SS.

En el caso FFH-SS la probabilidad de error es distinta, ya que se combinan las entradas al detector procedentes de varios *chips* transmitidos. En el caso de realizarse una combinación no-coherente lineal, es decir, si las variables de entrada al detector están formadas por la suma de los valores obtenidos a la entrada del detector, para cada uno de los *chips* transmitidos se obtiene:

$$|U_0|^2 = \sum_{k=0}^{L-1} |U_{0,k}|^2 \quad |U_1|^2 = \sum_{k=0}^{L-1} |U_{1,k}|^2 \quad (3.5)$$

siendo  $L$  el número de *chips* que componen un bit,  $k \in [0, \dots, L - 1]$  es el índice que identifica cada *chip* y la pareja de valores,  $|U_{0,k}|^2$

### 3.3. Estudio del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia no-coherente

---

siendo  $\gamma_b = \xi_b/J_o$ ,  $\xi_b$  la energía de un símbolo,  $J_o$  la SPD del ruido gaussiano y  $L$  el número de desplazamientos en frecuencia por símbolo. La relación entre la SNR por símbolo  $\gamma_b$  y la SNR por *chip*  $\gamma_c$  es  $\gamma_b = L\gamma_c$  [59].

Por lo tanto, para AWGN, la probabilidad de error es la misma si se utiliza BFSK ó BFSK combinada con SFH-SS. Pero el error aumenta, si se utiliza BFSK junto a FFH-SS con combinación lineal no-coherente.

#### 3.3.2 Recepción en canales con ruido aditivo gaussiano de banda estrecha

En el caso de tener BFSK con expansión SFH-SS, si el ruido de entrada es ruido gaussiano de banda estrecha el comportamiento del receptor cambia. Este tipo de interferencia se modela como un proceso aleatorio gaussiano de media cero y SPD plana sobre un fracción  $\rho$  de todo el ancho de banda ocupado por la señal. El valor  $\rho$  también se puede definir como: la relación entre el espectro ocupado por la interferencia de banda estrecha, frente al ocupado por el AWGN equivalente. El valor de la SPD es  $J_o/\rho$ , donde  $J_o$  es la SPD del AWGN al que equivale. Se supone que la interferencia afecta a la señal de comunicación SFH-SS, con una probabilidad  $\rho$ . También se asume que la interferencia al afectar a un canal de la expansión, lo cubre totalmente. La probabilidad de error en este caso es [59] [72]:

$$P_e = \frac{1}{2}\rho e^{-\frac{\rho\xi_b}{2J_o}} \quad (3.7)$$

siendo  $\xi_b$  la energía de un símbolo y  $J_o$  la SPD. Esta ecuación se puede diferenciar, obteniendo el valor de  $\rho$  que maximiza el error:

$$\rho_{opt} = \begin{cases} \frac{1}{\xi_b/2J_o} & (\xi_b/2J_o \geq 2) \\ 1 & (\xi_b/2J_o < 2) \end{cases} \quad (3.8)$$

Sustituyendo el valor de  $\rho_{opt}$  en la ecuación 3.7, se obtiene el valor del error con interferencia gaussiana de banda estrecha bajo el caso peor es:

$$P_e = \frac{e^{-1}}{\xi_b/J_o} \quad (3.9)$$

la ecuación anterior muestra que el error en estas condiciones disminuye de forma inversa al cociente  $\xi_b/J_o$ , mientras que con ruido AWGN lo hacía de forma exponencial, como se ve en la ecuación 3.4.

Si se considera modulación FFH-SS con BFSK utilizando combinación lineal de las variables  $|U_{i,k}|^2$  con  $i \in \{1,0\}$ , dada en la ecuación 3.5, la probabilidad de error del

sistema aumenta con respecto del caso SFH-SS [37], [72]. Para evitar esta degradación, en [38] se propone un esquema de receptor FFH-SS con combinación no-lineal de los valores  $|U_{i,k}|^2$ , que utiliza control automático de ganancia [59]. Este control divide cada término  $|U_{i,k}|^2$  en la ecuación 3.5, por el valor de la varianza del ruido presente en dicho término. Al realizar esta división, la varianza del ruido de cada *chip* se normaliza a la unidad, y la señal FFH-SS correspondiente se escala. Esto significa que cuando la señal FFH-SS, para una determinada frecuencia, está afectada por mucha potencia de ruido la varianza aumenta, mientras que con poca potencia de ruido la varianza disminuye. El gran inconveniente de esta técnica es que se hace necesario tener conocimiento del estado de la interferencia, para cada valor  $|U_{i,k}|^2$ . El desarrollo de la expresión del error para este caso se puede consultar en [59]:

$$P_e \leq \left( \frac{4}{e\gamma_c} \right)^L \quad (3.10)$$

donde  $\gamma_c = \frac{\gamma_b}{L} = \frac{\xi_b}{LJ_o}$ ,  $L$  es el número de *chips* por bit,  $\xi_b$  es la energía de un bit y  $J_o$  es la SPD. Para cada nivel de interferencias hay un  $L_{opt}$ , óptimo, que hace  $\gamma_c = 4$ . Si se sustituye en la ecuación 3.10,  $\gamma_c = 4$  y  $L = \frac{\gamma_b}{4}$  la probabilidad de error en ese caso es:

$$P_e \leq e^{-\gamma_b/4} \quad (3.11)$$

Esta probabilidad de error decae exponencialmente, con sólo una pérdida de  $3dB$  con respecto el error para BFSK con ruido AWGN. Así pues, en presencia de una interferencia gaussiana de banda estrecha el receptor FFH-SS se comporta mejor que el SFH-SS, eligiendo un valor de  $L$  adecuado.

Existen otros receptores diferentes al FFH-SS con control automático de ganancia, que también implementan una combinación no-lineal de la energía de los *chips*, sin la necesidad de conocer los valores estadísticos de la interferencia. Estos receptores aunque son más simples presentan peores resultados. Entre ellos destaca el receptor con limitador de amplitud (*clipper receiver*) [38], que limita los valores  $|U_{i,k}|^2$  mediante un umbral, antes de realizar la suma. El receptor auto-normalizado que divide para cada *chip* los valores  $|U_{i,k}|^2$ , mediante algún factor dependiente de ese mismo *chip* [50] [32], o el receptor mediante FFT desarrollado en [84], que implementa los correladores del receptor mediante los correspondientes puntos extraídos de la FFT.

### 3.3.3 Recepción en canales con ruido aditivo senoidal continuo

Este apartado se divide en dos casos. El primero con modulación BFSK, en el cual la interferencia está compuesta por un único tono continuo de tipo senoidal. El segundo lleva a cabo la extensión de los resultados a las modulaciones SFH-SS y FFH-SS, para

### 3.3. Estudio del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia no-coherente

---

el cual la interferencia se supone compuesta por varios tonos, cada uno de los cuales afecta a un canal distinto de la expansión SS.

#### Caso BFSK: tono simple

Las interferencias del tipo senoidal continuo son señales, que aparecen de forma habitual en canales de comunicación reales. Estas señales se producen en presencia de motores, fuentes de alimentación conmutadas, otros equipos de comunicación, etc. En este apartado se presentan las ecuaciones que rigen la probabilidad de error del receptor, cuando en el canal aparece dicha interferencia sumada a la señal de comunicación. Asumiendo que la interferencia está compuesta únicamente por un término en seno/coseno, con una frecuencia determinada, se puede expresar matemáticamente como:

$$j(t) = \sqrt{\frac{2S}{\alpha}} \cos(2\pi(f_c + \delta f_i)t + \phi_j) \quad (3.12)$$

donde  $S$  es la potencia de la señal de comunicación,  $\alpha$  es la relación entre la potencia de la señal y la interferencia,  $f_c + \delta f_i$  es la frecuencia de la interferencia y  $\phi_j$  es la fase supuesta uniformemente distribuida en el intervalo  $[0, 2\pi)$ . Con el fin de facilitar los cálculos, la frecuencia se ha expresado como la suma de  $f_c$ , frecuencia de la portadora, más un desplazamiento  $\delta f_i$ .

Realizando los cálculos para el caso BFSK y suponiendo que se ha transmitido un símbolo de valor  $b_i = 1$ , las dos entradas al detector  $|U_1|$  y  $|U_0|$ , mostradas en la figura 3.3, son como se demuestra en [7]:

$$|U_1| = |e^{j\phi} + \frac{1}{\alpha} \text{sinc}(\delta f_j T_b - 1) + n_1| \quad (3.13)$$

$$|U_0| = |\frac{1}{\alpha} \text{sinc}(\delta f_j T_b) + n_0| \quad (3.14)$$

donde  $T_b$  es el periodo de un bit y  $n_1/n_0$  son los términos procedentes del ruido AWGN, en la salida de los dos correladores. Si se define  $\mu = \delta f_j T_b$ , la interferencia estará centrada en la frecuencia correspondiente al bit  $b_i = 0$ , cuando  $\mu = 0$ , y en la frecuencia del bit  $b_i = 1$ , cuando  $\mu = 1$ . Además, al considerar el caso sin ruido AWGN,  $n_0 = n_1 = 0$ , las ecuaciones anteriores quedan, tal como se puede comprobar en [7] [8]:

$$|U_1| = |e^{j\phi} + \frac{1}{\alpha} \text{sinc}(\mu - 1)| \quad (3.15)$$

$$|U_0| = |\frac{1}{\alpha} \text{sinc}(\mu)| \quad (3.16)$$

Y la probabilidad de error condicionada a la transmisión del símbolo  $b_i = 1$  es:

$$P_{e,1}(\mu, \alpha) = Pr[|U_1|^2 < |U_0|^2/b_i = 1] \quad (3.17)$$

Como se demuestra en [7] esta probabilidad se puede expresar como:

$$P_{e,1}(\mu, \alpha) = P'(\mu, \alpha)I(\text{sinc}(\mu - 1) > 0) + (1 - P'(\mu, \alpha))I(\text{sinc}(\mu - 1) < 0) \quad (3.18)$$

donde  $I(c)$  es 1 si  $c$  es verdadero y 0 en caso contrario, y  $P'$  se define como [7]:

$$P'(\mu, \alpha) = \begin{cases} \frac{1}{\pi} \cos^{-1}(g(\mu, \alpha)) & |g(\mu, \alpha)| \leq 1 \\ 0 & g(\mu, \alpha) > 1 \\ 1 & g(\mu, \alpha) < -1 \end{cases} \quad (3.19)$$

donde

$$g(\mu, \alpha) = \frac{\alpha - \text{sinc}^2(\mu) + \text{sinc}^2(\mu - 1)}{2\sqrt{\alpha}\text{sinc}(\mu - 1)} \quad (3.20)$$

La probabilidad de error para  $b_i = 0$  es simétrica siendo:

$$P_{e,0}(\mu, \alpha) = P_{e,1}(1 - \mu, \alpha) \quad (3.21)$$

Y la probabilidad total es [7]:

$$P_e(\mu, \alpha) = (1/2)[P_{e,1}(\mu, \alpha) + P_{e,0}(\mu, \alpha)] = (1/2)[P_{e,1}(\mu, \alpha) + P_{e,1}(1 - \mu, \alpha)] \quad (3.22)$$

En la figura 3.6, se muestra una representación gráfica de la probabilidad de error dada en la ecuación 3.22. En dicha gráfica, los valores de  $\mu$  son los que indican la posición de la interferencia. Así, para  $\mu = 0$  la interferencia tiene una frecuencia que coincide con la de la portadora del símbolo  $b_i = 0$ . Cuando el valor de  $\mu$  varía, la frecuencia de la interferencia se desplaza, llegando a coincidir con la frecuencia de la portadora del bit  $b_i = 1$ , para  $\mu = 1$ .

Como se observa en la gráfica 3.6, los valores de  $\mu$  que maximizan la probabilidad de error son aquellos que sitúan la interferencia sobre una de las portadoras, como cabría esperar. Aunque para interferencias con potencia elevada,  $\alpha$  pequeño, la interferencia es capaz de producir una gran probabilidad de error, aunque se sitúe en posiciones de frecuencia intermedias entre las dos portadoras. También es lógico, el comportamiento de la probabilidad de error en función de la potencia de la interferencia, así para valores  $\alpha < 1$  la interferencia tiene una potencia superior a la de la portadora produciendo una elevada tasa de errores. Por el contrario,  $\alpha > 1$  implica una potencia de la interferencia menor que la de la portadora y una tasa de errores menor.

### 3.3. Estudio del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia no-coherente

---

Figura 3.6: *Probabilidad de error para BFSK con interferencia en tono simple, donde  $\alpha$  es la relación entre la potencia de la señal y la interferencia.*

#### **Caso SFH-SS y FFH-SS: multitono**

Antes de pasar a considerar el caso CWT para las expansiones SFH-SS y FFH-SS, hay que hacer una modificación en el modelo de la interferencia. Esta distinción se basa en el hecho de que ahora ya no se tendrá un único tono, sino que la interferencia se divide en múltiples términos senoidales de la forma 3.12, cada uno de los cuales afecta a un canal distinto, aunque no todos los canales de la expansión tienen que estar afectados. De esta manera, la potencia total se divide en  $Q$  términos senoidales, con la misma potencia y fases aleatorias. Al haber  $Q$  componentes, la interferencia altera  $Q$  canales de la expansión SFH-SS. Esta interferencia se suele denominar en la bibliografía como interferencia multitono, y se representa matemáticamente como:

$$j(t) = \sum_{l=1}^Q \sqrt{(2R/Q)} \cos(\omega_l t + \theta_l) \quad (3.23)$$

donde  $R$  es la potencia total de la interferencia,  $\omega_l$  son las frecuencias de cada término senoidal y  $\theta_l$  es la fase de cada uno de ellos supuesta aleatoria en el intervalo  $[0, 2\pi)$ . La relación de la potencia de la señal SS,  $S$ , y la potencia de uno de los tonos de la interferencia,  $R/Q$ , definida como  $\alpha$  es:

$$R/Q = S/\alpha \quad (3.24)$$

Los tonos se distribuyen por todo el ancho de banda,  $W_{ss}$ , ocupado por la señal SFH-SS. Para distribuir la potencia de la interferencia entre los canales formados tras la expansión SFH-SS, se pueden utilizar distintas estrategias. De entre todas las posibles distribuciones, el estudio se va a restringir al que produce resultados más pesimistas, es decir, el caso peor. Esta situación se obtiene mediante las siguientes dos suposiciones:

1. Los tonos se sitúan sobre alguna de las frecuencias usadas por la expansión SFH-SS. Ya que, como se ha visto en la gráfica 3.6, la probabilidad de error se maximiza cuando la interferencia se sitúa sobre alguna de las dos portadoras que forman un canal. Aunque esta condición raramente se da, simplifica el análisis y produce peores resultados de rendimiento.
2. De todas las distribuciones posibles de la potencia de la interferencia, entre los canales de la expansión SFH-SS, hay una que es la más dañina. Ésta se produce cuando  $\alpha$ , definido anteriormente en 3.24, se mantiene constantemente a 1, para todos los canales afectados por la interferencia. Lo que se modifica, al variar la potencia total de la interferencia  $R$ , es el número de canales afectados  $Q$ , pero en todos ellos la interferencia es capaz de producir errores con probabilidad 1/2.

Esta afirmación se puede comprobar en la gráfica 3.6, porque para  $\alpha = 1$  y  $\mu = 1$  ó  $\mu = 0$ , que es la situación planteada al cumplirse la condición anterior, la probabilidad de error alcanza el nivel 1/2.

Si se define la SPD de la interferencia como:  $J_i = R/W_{ss}$ , donde  $W_{ss}$  es el ancho de banda ocupado por la modulación. La relación entre energía de los bits  $E_b$  y la SPD de la interferencia  $J_i$  es:

$$\frac{E_b}{J_i} = \frac{m\alpha}{2QK} \quad (3.25)$$

donde  $m$  es el número de frecuencias utilizado en la expansión SFH-SS y  $m/2$  el número de canales, porque se utiliza BFSK a nivel de bit. El término  $K$  es el número de bits codificados en cada símbolo de la transmisión FSK, que con modulación BFSK es  $K = 1$ . Bajo estas condiciones la probabilidad de error para un sistema SFH-SS con un *chip* por símbolo transmitido, como se demuestra en [72] es:

$$P_b = \begin{cases} \frac{1}{2} & \frac{E_b}{J_i} \geq \frac{2^K}{K} \\ \frac{2^{K-1}}{KE_b/J_i} & \frac{E_b}{J_i} \leq \frac{2^K}{K} \end{cases} \quad (3.26)$$

al suponer BFSK se puede sustituir el valor  $K = 1$ .

A la vista de la ecuación 3.26, se pueden hacer tres comentarios:

### 3.3. Estudio del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia no-coherente

---

1. El primer término en la ecuación 3.26, que dicta la probabilidad de error para SFH-SS, con interferencia multitono bajo el caso peor, es  $\frac{1}{2}$ , valor constante e independiente de la potencia de la interferencia. Esta situación se produce, cuando la interferencia tiene suficiente potencia para distribuir por lo menos un término senoidal, en cada uno de los canales de la expansión SFH-SS. Como todos los canales se ven afectados la probabilidad es  $\frac{1}{2}$ .
2. El segundo término constituye la situación en la que la potencia de la interferencia disminuye. En este caso, la interferencia no es capaz de cubrir todos los canales con un término senoidal, bajo las restricciones de que la potencia de cada término asegure que  $\alpha = 1$ . Así, como se desprende de la fórmula 3.26, la probabilidad de error decae de forma inversamente lineal con  $E_b/J_i$ .
3. En el apartado anterior 3.3.2, bajo interferencias gaussianas de banda estrecha, el rendimiento más pesimista del receptor SFH-SS, dado por la ecuación 3.7, también decae de forma inversamente proporcional a  $E_b/J_i$ . Pero en dicha ecuación el numerador contiene el término  $e^{-1}$ , por lo tanto la probabilidad de error es menor que el caso actual.

Para concluir este apartado, se va a considerar el rendimiento de los receptores para la modulación FFH-SS, con interferencia CWT bajo el caso peor. Debido a la gran extensión que presenta el desarrollo de las ecuaciones, que rigen la probabilidad de error para este caso, sólo se muestran los resultados de forma gráfica, indicando las fuentes donde se pueden obtener las ecuaciones teóricas.

En primer lugar, se considera la probabilidad de error para el receptor con combinación lineal. En este caso, la energía proveniente de los  $L$  *chips* en los que se divide cada bit, se suma de forma lineal después de realizar la recepción no-coherente. Las ecuaciones que rigen el comportamiento de la probabilidad de error para este receptor, se pueden consultar en [81]. A partir de estas ecuaciones se obtienen los resultados mostrados en la gráfica 3.7.

En la figura 3.7, se comprueba que la probabilidad de error del receptor con combinación lineal bajo el caso peor aumenta, conforme se incrementa el número de *chips* en los que se divide un bit. Así pues, el sistema se degrada al pasar de una expansión SFH-SS, caso  $L = 1$ , a la expansión FFH-SS y combinación lineal, casos  $L = 3$  y  $L = 5$ .

Al considerar receptores FFH-SS con combinación no-lineal de los *chips* que componen un bit, los resultados cambian. Por ejemplo, para el receptor con limitador de amplitud, las ecuaciones de la probabilidad de error en el caso peor están desarrolladas en [83]. De forma gráfica, esta probabilidad de error se muestra en la figura 3.8. En dicha figura se puede comprobar que existe un intervalo de valores de la SNR, para los cuales la extensión FFH se comporta mejor que la SFH,  $L = 1$ . Además, también

Figura 3.7: Probabilidad de error para el receptor FFH-SS, no-coherente con combinación lineal de los chips, con interferencia multitono en el caso peor para distintos números de chips  $L$ . Esta gráfica ha sido obtenida de la referencia [81].

se observa que para cada intervalo de hay un  $L$  óptimo, que minimiza la probabilidad de error.

### 3.4 Supresión de interferencias en modulaciones de Espectro Ensanchado

En las secciones 3.2 y 3.3 se han descrito las características de inmunidad frente a interferencias, que poseen las señales SS. Sin embargo, puede haber situaciones donde la interferencia presente una energía lo bastante grande como para degradar fuertemente la comunicación. En estos casos es posible aumentar la inmunidad frente a interferencias de las señales SS, mediante el incremento del ancho de banda ocupado por la señal [59] [46]. Pero en sistemas reales, donde se dispone de un ancho de banda limitado, esta expansión de la información puede no ser suficiente para conseguir un rendimiento aceptable. En este caso, es necesario usar algún otro método para mitigar la energía de la interferencia.

Cuando la interferencia tiene un ancho de banda estrecho, es posible mejorar mucho el rendimiento del sistema mediante técnicas de supresión. Estas técnicas explotan las diferencias entre las formas de onda de la señal SS y de la interferencia. En concreto, estas técnicas concentran, mediante el procesado de la señal, la energía de la interferencia y la eliminan perturbando mínimamente la señal SS. Dependiendo del dominio

Figura 3.8: *Probabilidad de error para el receptor FFH-SS, no-coherente con limitador de amplitud, con interferencia multitono en el caso peor para distintos números de chips  $L$ . Esta gráfica ha sido obtenida de la referencia [83].*

donde se produce este procesamiento, se pueden encontrar técnicas de supresión aplicadas sobre el dominio temporal o sobre el dominio de alguna transformada. También se puede distinguir entre las técnicas aplicadas sobre DS o FH, ya que aunque el fundamento de la supresión se basa en los mismos conceptos, la forma de llevarlo a cabo difiere, porque las características de las señales y los esquemas de emisor y receptor son distintos.

A continuación, se va a pasar a estudiar estos métodos de supresión tanto en el dominio temporal como en el frecuencias.

#### 3.4.1 Supresión de interferencias en el dominio temporal

En el dominio temporal la supresión se realiza mediante el filtrado adaptativo de la señal de entrada. El filtro explota las diferencias entre la forma de onda de la señal de comunicación SS, con múltiples cambios aleatorios de fase o frecuencia, y la forma de onda continua de la interferencia. Esta disparidad permite, que el filtro adaptativo pueda eliminar casi completamente la interferencia recibida, mientras que sólo una pequeña parte de la señal original se ve afectada.

El proceso de ajuste de los coeficientes del filtro, no se ve alterado por la señal de comunicación SS ni por el ruido AWGN, ya que estos procesos no están localmente

correlacionados, lo que dificulta la predicción de sus valores futuros a partir de sus valores pasados [60] [51]. Por el contrario, la interferencia al ser un proceso de banda estrecha es fácilmente predecible a partir sus valores pasados. El nivel actual de ruido, estimado por el filtro, se sustrae a la señal de entrada, eliminando dicha interferencia y dejando la entrada compuesta, casi únicamente, por la señal SS y el ruido AWGN.

A continuación se detalla la aplicación de estos algoritmos, sobre modulaciones DS-SS y FFH-SS, tal como se propone en la bibliografía.

### Algoritmos aplicados sobre modulaciones DS

Para modulaciones DS-SS el esquema de receptor es el que aparece en la figura 3.9. Este receptor realiza la demodulación de forma coherente, ya que la señal DS-SS se basa en modulación PSK codificada mediante cambios de fase, como se ha visto en la sección 3.2.1. Para recuperar los símbolos transmitidos hay que comprimir la señal, eliminando los cambios de fase introducidos por la secuencia PN en el emisor. Para ello se integra la señal sobre cada periodo  $T_c$ , a continuación se realiza la correlación con la secuencia PN y por último, se suman todos los valores que forman un bit.

*Figura 3.9: Esquema del receptor DS con filtro adaptativo para la eliminación de interferencias de banda estrecha. Dicho receptor está formado por un correlador centrado en la frecuencia de la portadora  $\omega_0$ , un integrador sobre el periodo de un chip  $T_c$ , el filtro adaptativo y por último el correlador y el sumador para comprimir la señal.*

Como se ve en el esquema 3.9, el receptor incluye un filtro lineal adaptativo situado entre la salida del integrador y el correlador de secuencia PN [51] [60], dicho filtro tiene como objetivo la eliminación de las interferencias de banda estrecha. Los coeficientes del filtro se pueden hallar resolviendo el conjunto de ecuaciones de Wiener-Hopf [51], aunque este cálculo necesita conocer la matriz de covarianza de la señal de entrada. Sin embargo tal como se muestra en [29], los coeficientes del filtro pueden ser estimados recursivamente mediante algoritmos iterativos como el conocido LMS (*Least Mean Squared*) [27], que no requiere ningún conocimiento de los valores estadísticos de la señal de entrada. Además, puede ser implementado en tiempo real, como se describe en [27]. La limitación que impone el uso del algoritmo LMS es que la interferencia debe permanecer estacionaria, durante el periodo necesario para que el algoritmo converja, siendo imposible su utilización con interferencias de banda estrecha no-estacionarias [40].

En ausencia de interferencia, la salida del integrador está formada por la suma de AWGN y de la señal DS, generada por la secuencia pseudoaleatoria PN. Esta señal no está correlacionada [59], lo que imposibilita al filtro predecir el valor actual en su entrada a partir de valores pasados. En esta situación, la única forma de minimizar el error cuadrático medio es anulando todos los coeficientes del filtro, y éste deja de actuar. Cuando en la entrada del receptor hay señal DS, AWGN e interferencia, la única de ellas que está correlacionada es la interferencia, y el filtro predice su valor al intentar minimizar el error cuadrático medio.

Esta técnica de supresión de interferencias presenta una objeción importante, debido a que la secuencia de valores transmitidos por el emisor, tras la expansión de la señal con la secuencia PN, es una secuencia binaria independiente e idénticamente distribuida. Por lo tanto, es una secuencia no-gaussiana y el filtro óptimo, para predecir un proceso de banda estrecha en presencia de una secuencia no-gaussiana, es un filtro no-lineal [87] [23]. El filtro lineal adaptativo utilizado en el esquema de receptor 3.9, aunque produce un aumento del rendimiento del sistema, no es el óptimo. Para profundizar en este tema se puede consultar [23] y [60].

#### Algoritmos aplicados sobre modulaciones FFH

El rendimiento del receptor FFH-SS, como se ha visto en la sección 3.3, se ve muy afectado por interferencias de banda estrecha y la comunicación se ve fuertemente degradada. Como se ha visto en la sección 3.3, para disminuir el BER se pueden utilizar técnicas basadas en la combinación no-lineal de la energía de los *chips* en los que divide cada bit la modulación FFH-SS.

En este apartado se presenta un receptor FFH-SS, que implementa la recepción de los símbolos transmitidos mediante combinación lineal de los valores de cada *chip*, y utiliza las técnicas de supresión de interferencias en el dominio temporal, para evitar la degradación que producen las interferencias de banda estrecha. En la entrada del receptor se sitúa un filtro lineal adaptativo similar al utilizado en la figura 3.9 para modulaciones DS-SS. Este filtro es el encargado de eliminar la interferencia, ajustando sus coeficientes de forma adaptativa mediante algoritmos como el LMS, citado en el apartado 3.4.1.

Esta técnica está inspirada en el modelo implementado para la modulación DS-SS. En este caso, se puede demostrar que la señal FFH-SS transmitida está descorrelacionada a intervalos  $T_c/(2m)$  [28], donde  $T_c$  es el periodo de un *chip*,  $m$  es número de frecuencias utilizadas en la expansión FFH-SS y  $m/2$  es el número de canales, suponiendo modulación BFSK a nivel de bit. De esta forma, se puede predecir la interferencia sin destruir la señal FFH, ya que la señal de comunicación no está correlacionada, y por tanto no es predecible por el estimador cuadrático medio. Por otro lado, como la interferencia de banda estrecha está altamente correlacionada, puede ser

fácilmente predicha y eliminada [28].

El esquema de este receptor presentado en [28] se muestra en la figura 3.10. El receptor tiene a la entrada dos correladores en cuadratura para situar la señal en banda base. A continuación, se sitúa el filtro adaptativo, que contiene los coeficientes  $a_k$  y elementos de retraso con periodo  $T_c/(2m)$ , formando un filtro complejo. La salida del filtro se vuelve a correlar con una de las frecuencias de la expansión FFH-SS. Seguidamente, las componentes en cuadratura de la señal pasan a través de los filtros paso-bajo y se combinan de forma cuadrática para eliminar la fase. Por último los valores obtenidos para cada *chip* se suman de forma lineal, lo que completa la recepción no-coherente.

Figura 3.10: *Esquema del receptor FFH-SS con filtrado adaptativo para eliminación de interferencias de banda estrecha. El receptor contiene dos correladores en cuadratura sintonizados a la frecuencia de la portadora,  $\omega_0$ , a continuación se sitúa el filtro adaptativo y por último el receptor no-coherente, que combina los  $L$  chips en los que se divide un bit en la modulación FFH-SS.*

El BER para este receptor es mejor que el obtenido para receptores con combinación no-lineal, como el receptor autonormalizado, aunque no alcanza tan buenos resultados como el receptor no-lineal con control automático de ganancia [28], que está muy cercano al óptimo en términos de señal a ruido.

Pero al igual que se comentó para el caso DS-SS, la gran limitación, que supone la utilización de filtros adaptativos en la supresión de interferencias, es que ésta debe ser estacionaria durante un cierto intervalo de tiempo, para que el algoritmo LMS pueda converger.

#### 3.4.2 Supresión de interferencias en dominio de la transformada

El objetivo principal de la supresión de interferencias mediante transformada es representar la señal recibida, en el dominio de la transformada, donde la señal SS y la interferencia son rápidamente distinguibles. Bajo condiciones ideales, en este dominio la interferencia aparece como un impulso, mientras que el espectro de la señal SS es mucho más plano. La zona del espectro donde se sitúa la interferencia se elimina, produciendo una reducción drástica de su energía y una pequeña pérdida en la señal SS.

De esta forma, la supresión de la interferencia se reduce a la detección y eliminación de los puntos transformados que superen un cierto nivel de energía. Tras ello, se realiza la transformada inversa cuyo resultado es una señal cercana a la señal original SS sin interferencia. Debido a la forma binaria en la cual se produce la supresión, con este proceso se obtienen peores BER que mediante técnicas más complejas, con filtrado adaptativo en el dominio de la transformada [46] [66] [40].

Sin embargo en muchos casos, la simplicidad y la habilidad para reaccionar rápidamente ante cambios en las características de la interferencia, sin tener que esperar a la convergencia de los algoritmos adaptativos, hacen aconsejable la utilización del receptor basado en la detección y eliminación de los puntos transformados [46]. Estas razones han sido tenidas en cuenta en el estudio realizado, que se ha restringido a receptores que implementan esta supresión binaria.

Las principales ventajas de la supresión de interferencias en el dominio de la transformada, frente al dominio temporal son:

1. En primer lugar, la adaptación más rápida a los cambios que presente la interferencia. El filtrado adaptativo en el dominio temporal necesita un cierto intervalo de tiempo hasta que los coeficientes del filtro convergen [40].
2. Además, en el dominio de la transformada es posible realizar fácilmente procedo no-lineal, siendo mucho más complejo en el dominio temporal [40].

La desventaja del dominio transformado, es que todo el proceso de supresión esta supeditado a la adecuada elección de la transformación, que debe concentrar la mayor cantidad de energía de la interferencia en unos pocos puntos. Gráficamente el proceso de supresión con transformada se muestra en la figura 3.11

En la bibliografía, la supresión mediante transformada ha sido utilizada exclusivamente en modulaciones DS-SS. El esquema de receptor DS-SS modificado se muestra en la figura 3.12. Este esquema coincide con el mostrado en la figura 3.9 cambiando el filtro lineal temporal por el filtro de supresión con transformada. La inclusión de este

Figura 3.11: *Proceso de supresión de interferencias con transformada. Los puntos que superan un cierto umbral de energía en el dominio transformado son eliminados. En la figura  $R_c = 1/T_c$  representa el periodo de un chip.*

módulo se realiza una vez que la señal ha sido demodulada, antes de ser correlada con la secuencia PN [46]. Suponiendo que el receptor está sincronizado perfectamente en fase con el emisor, el vector obtenido tras la demodulación es de tipo real. Este vector constituye la entrada al módulo de supresión, estando compuesto por un número de muestras igual al tamaño del bloque usado en la transformada.

Una vez en el dominio de la transformada, el vector resultante contiene los coeficientes sobre los que se realiza la supresión, eliminando los puntos cuya energía supere un cierto umbral. Gráficamente, la supresión se representa mediante la multiplicación con el vector  $\lambda$ , tal como se ve en la figura 3.12. Este vector  $\lambda$  se construye con valores iguales a 1, en las componentes que deben conservarse dentro del vector transformado, y 0 para las que deben anularse.

Para recuperar la señal en el tiempo, es necesario realizar la transformada inversa representada mediante  $T^{-1}$ . Por último, se realiza la correlación con la secuencia PN, y se suman todos los valores correspondientes a los *chips* que forman un símbolo, lo que permite comprimir la señal.

### Supresión mediante Transformada de Fourier

La FT fue la primera transformada utilizada en la supresión de interferencias. Su aplicación se analiza en los siguientes artículos [49] [14] [15]. En ellos la transformación de la señal recibida se realiza mediante su segmentación en bloques de tamaño finito, sobre los cuales se realiza la FFT. Una vez eliminados los puntos con una amplitud que superan un cierto umbral se realiza la transformada inversa.

Figura 3.12: *Esquema del receptor DS con módulo de supresión basado en transformada. El bloque que realiza la transformada, a la salida del correlador y el integrador con periodo  $T_c$ , se representa por  $T$ . A continuación, el bloque transformado se multiplica por el vector  $\lambda$  para eliminar los coeficientes cuya energía supere un cierto umbral. Por último, se realiza la transformada inversa  $T^{-1}$  y se comprime la señal mediante el correlador y el sumador con la secuencia PN.*

La supresión utilizando la FFT da resultados satisfactorios cuando la interferencia es estacionaria, es decir, sus propiedades no varían con el tiempo. El problema aparece al considerar interferencia no-estacionarias, con transiciones bruscas en el tiempo que expanden su energía en el dominio de frecuencias, lo que dificulta la distinción entre señal SS e interferencia [47].

Otro problema asociado a la FFT, que aparece al usar un bloque con tamaño finito, es el enventanado de la señal. El enventanado produce lóbulos adyacentes al lóbulo principal en el espectro de la interferencia, que no son despreciables. Por lo tanto, esta expansión de la energía de la interferencia en frecuencia, reduce la habilidad del supresor para diferenciar entre ambas señales. Se puede reducir el efecto de la ventana, recurriendo a funciones no rectangulares [14] [15] [64], pero el inconveniente es el incremento en la complejidad del receptor, producida por el uso de segmentos solapados necesarios para obtener una reconstrucción perfecta de la señal SS [46].

#### **Supresión mediante Transformada Wavelet**

La aplicación de la WT en la supresión de interferencias de banda estrecha, pretende concentrar la energía de la interferencia en unos pocos puntos pertenecientes a cualquier subbanda, en las que se transforma la señal original, con el fin de eliminarlos y de degradar lo menos posible la señal SS [46], [47]. El esquema del receptor es el mismo que el dado en la figura 3.12, donde  $T$  representa en este caso la WT.

Si la interferencia es una señal CWT, como la definida en el apartado 3.3.3, los resultados obtenidos en [47] muestran que la diferencia en el BER, utilizando FFT o WT no difieren mucho, aunque la WT supera ligeramente a la FFT. Por otro lado, esta comparación en el caso de ruido no-estacionario, modelado como un proceso gaussiano de banda estrecha con un periodo de encendido y otro de apagado, muestra la imposibilidad de la FFT para concentrar señales no-estacionarias en pocos puntos.

En este caso, las características de la WT la hacen muy superior en términos de BER, ya que concentra mucho mejor la energía de la interferencia [47].

### Supresión mediante bancos de filtros

La aplicación de los bancos de filtros a la supresión de interferencias se muestra en [31], [63], [46] y en [48]. La principal ventaja que presentan los bancos de filtros frente a la FFT es la mayor resolución en frecuencia. Además, la propiedad de reconstrucción perfecta permite descomponer y reconstruir la señal SS original sin distorsiones, como se ha visto en el apartado 2.4.1. A continuación se van a analizar algunas de las alternativas propuestas en la bibliografía:

1. El esquema presentado en [31], utiliza los filtros de análisis para realizar una división regular del espectro de la señal original. Una vez descompuesta la señal, se elimina aquella subbanda cuya energía supere un cierto umbral. Por último, se reconstruye la señal mediante los filtros de síntesis, diseñados junto con los de análisis para cumplir la condición de reconstrucción perfecta.
2. En [63] se utilizan los bancos de filtros para separar la señal en subbandas de frecuencia, pero sin introducir ningún diezmado después del análisis y no perder resolución temporal, lo que constituye una transformación altamente redundante. Además, el banco está diseñado para que la reconstrucción se realice mediante la simple suma de sus bandas obtenidas a partir de los filtros de análisis, teniendo todos ellos fase lineal [63]. Estas condiciones se obtienen construyendo un banco de filtros con reconstrucción perfecta, como se mostró en la sección 2.4.1, combinando para cada subbanda el filtro de análisis y síntesis para formar un único filtro de análisis. La supresión se realiza sobre cada una de las subbandas, eliminando los intervalos de puntos que posean un nivel de energía superior a un umbral.
3. Por último, en [48] se utilizan transformadas solapadas. Estos bancos de filtros constituyen un caso particular en el cual, el tamaño del bloque utilizado tiene una longitud  $M = 2KN$ . Donde  $N$  es el tamaño del vector obtenido a salida del integrador, como se muestra en la figura 3.12, y  $K$  es el factor de solape [78]. El bloque utilizado por la transformada es de  $M$  puntos, estando formado por el vector de salida del integrador actual con  $N$  puntos, junto con  $K$  vectores de salida del integrador anteriores necesarios completar  $M$  puntos. De esta forma, los bloques utilizados en la transformada contienen segmentos solapados. Esta propiedad de la transformada incrementa la atenuación de las zonas de corte de los filtros, que dividen el espectro de la señal.

#### Supresión mediante Paquetes de Wavelets

En último lugar, se considera un algoritmo basado en la aplicación de las WPT. Esta transformada cuyas características se han mostrado en el apartado 2.5, permite una descomposición irregular del espacio de representación tiempo-frecuencia. En concreto este apartado se centra en el algoritmo mostrado en [80], que propone un método para la supresión de interferencias denominado supresión Adaptativa en Tiempo-Frecuencia (*Adaptive Time-Frequency, FTA*). Sus dos principales características son:

- La evaluación y elección de cuál es el mejor dominio para suprimir la interferencia, el temporal o el de la transformada.
- La adaptación de la descomposición en subbandas para representar la interferencia de la forma más eficiente posible, mediante la aplicación de la WPT.

El mejor dominio para realizar la supresión viene dado por las características de la interferencia. Así una interferencia de tipo impulsional, concentrada en el tiempo, tendrá su energía muy expandida en el dominio frecuencial. En este caso, el mejor dominio para eliminar la energía de la interferencia es el temporal. Por el contrario, una interferencia de tipo CWT tiene su energía comprimida en frecuencias y expandida en el tiempo. Por lo tanto, este segundo caso es mejor tratarlo en el dominio frecuencial.

El algoritmo FTA, para determinar el espacio donde llevar a cabo la supresión, calcula la energía del vector a la salida del integrador, agrupando sus componentes en intervalos. Si una interferencia está contenida en un intervalo, y el número de puntos que compone dicho intervalo es menor de un cierto valor  $N_c$ , entonces son eliminados en el dominio temporal. En este caso, al estar la energía de la interferencia concentrada en el tiempo, no es necesario transformar el vector, y la supresión se lleva a cabo sobre su representación temporal. Si por el contrario el número de puntos del intervalo, que concentra la interferencia, es mayor que  $N_c$ , la distribución de energía está lo suficientemente expandida en el tiempo, como para ser recomendable su tratamiento en el dominio de la transformada [80].

Las técnicas analizadas anteriormente basadas en FFT, WT y bancos de filtros, utilizan una transformación fija para representar la señal. Para cualquiera de las transformadas anteriores, se mantienen las limitaciones impuestas por el solape en frecuencias entre subbandas y la división fija de espacio tiempo-frecuencia [80]. De esta forma, si una interferencia queda situada en posiciones de frecuencia cercanas a la zona de corte entre dos subbandas, su energía queda repartida entre ambas.

Para evitar este problema el FTA recurre a una descomposición diádica o triádica, en cada paso del árbol de descomposición irregular, basado en WPT. La descomposición diádica ha sido la única utilizada en el capítulo 2, por ser la más común. En el caso de utilizar descomposiciones triádicas el número de filtros aplicados sobre la

señal de entrada es tres, obteniendo tres subbandas, que se diezman por tres, es decir, sólo uno de cada tres puntos se mantiene el resto se eliminan [78]. Gráficamente, una descomposición triádica es tal como se muestra en la figura 3.13:

Figura 3.13: *Banco de filtros triádico donde  $h_0$  y  $h_1$  son los filtros de análisis y síntesis para la subbanda paso-bajo,  $b_0$  y  $b_1$  son los filtros paso-banda y  $g_0$  y  $g_1$  los paso-alto. Al ser un filtro triádico la señal se diezma y se sobremuestra por 3.*

El algoritmo en cada nodo del árbol elige entre la descomposición diádica o triádica, que concentra más energía en una sola subbanda. El siguiente nivel de resolución se obtiene iterando sobre la subbanda, que contiene la interferencia. Por último, el algoritmo finaliza cuando los nodos hijos presentan una concentración de energía menor que la del padre. En la figura 3.14, se puede observar un ejemplo del árbol de descomposición, junto con la división del espectro frecuencial, obtenido al aplicar el FTA.

Los resultados alcanzados con este algoritmo se muestran en [80]. Su rendimiento es cercano al ideal en escenarios con interferencias del tipo CWT, ruido gaussiano de banda estrecha e interferencias no estacionarias, superando los receptores con supresión de interferencia basada en FFT, WT y bancos de filtros.

## 3.5 Conclusiones


En este capítulo se ha hecho un repaso de las características más destacables de las modulaciones SS, haciendo especial énfasis en sus características de inmunidad frente a interferencias. Esta inmunidad permite su aplicación en canales, con altos niveles de interferencias o compartidos por múltiples usuarios.

También se ha realizado un estudio exhaustivo de las características del receptor BFSK no-coherente con expansión SFH-SS y FFH-SS, frente a tres tipos de ruido e interferencias. Los resultados mostrados indican que el rendimiento del receptor se degrada, cuando aparecen interferencias de banda estrecha gaussiana o CWT. En resumen, los resultados mostrados en los apartados 3.3.1, 3.3.2 y 3.3.3 son:

- Caso ruido aditivo blanco gaussiano:

### 3.5. Conclusiones

---

Figura 3.14: 

La conclusión que se extrae del resumen anterior, es que la expansión FFH-SS se comporta mejor que la SFH-SS, para interferencia de banda estrecha. En un escenario que incluya la suma de varios tipos de ruido e interferencia, la mejora de la FFH-SS frente a SFH-SS se produce, siempre que la degradación que sufre la FFH por la combinación no-coherente de varios *chips* con ruido AWGN, sea menor que la ganancia obtenida por la reducción de la potencia de la interferencia.

La degradación que sufre la recepción de señales SS, en presencia de interferencia se puede suavizar mediante la inclusión de técnicas de supresión de interferencias de banda estrecha. Dichos métodos explotan la diferencia entre las formas de onda de la interferencia y la señal SS, para eliminar la energía de la interferencia, sin degradar la señal de comunicación. Dependiendo del dominio donde se realice la excisión se pueden encontrar, técnicas en el dominio temporal, basadas en filtros adaptativos, y técnicas en el dominio frecuencial, que utilizan alguna transformada para diferenciar las dos señales.

Una cuestión, que ha quedado abierta tras la exposición de la revisión bibliográfica, es la posible aplicación de las técnicas de supresión de interferencias basadas en transformada sobre sistemas FH-SS, que consiga aumentar la calidad de servicio de estos sistemas (probabilidad de error) en entornos con interferencias de banda estrecha. Estas interferencias como se ha demostrado son muy perjudiciales y obligan a recurrir a la expansión de la energía de los bits en múltiples *chips*, modulaciones FFH-SS, para evitar el aumento de la tasa de error.

### 3.5. Conclusiones

---

## Parte III

**Análisis, diseño e implementación  
del receptor con supresión de  
interferencias para modulaciones de  
Espectro Ensanchado con Saltos de  
Frecuencia propuesto.**



---

Revisado el estado del arte, se ha llegado a la conclusión que bajo los requisitos y condiciones analizados en la introducción es necesario un estudio adicional, objetivo de la presente tesis. Bajo el supuesto de la situación vista en dicha introducción, resumida bajo las siguientes condiciones del sistema a diseñar: sistemas cuya calidad de servicio (probabilidad de error) no se degrade en entornos o canales interferentes, una posible alternativa a tener en cuenta, la cual se va a desarrollar detenidamente a través de la fase de análisis, diseño e implementación es la de un receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia, con un módulo de supresión de interferencias basado en la Transformada Wavelet.

Los contenidos de esta parte se dividen en los dos capítulos siguientes:

- Receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias mediante Transformadas Wavelet Sobremuestreadas
- Implementación del algoritmo de supresión de interferencias mediante estructuras *Lattice* sobre Procesadores Digitales de Señal



# Capítulo 4

## Receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias mediante Transformadas Wavelet Sobremuestreadas

### 4.1 Introducción

Las modulaciones SS son robustas frente a interferencias, pero como se ha mostrado en el capítulo 3, la tasa de bits erróneos en sistemas con modulaciones FH-SS aumenta mucho, cuando la señal se ve afectada por interferencias de banda estrecha. Estas interferencias se modelan, habitualmente, como señales del tipo CWT, representadas mediante un término senoidal de una determinada frecuencia, o también se pueden modelar como ruido gaussiano de banda estrecha.

Las soluciones propuestas a este problema en la bibliografía se basan en el aumento del número de *chips* por bit. Este aumento modifica la modulación que pasa a ser FFH-SS. Para disminuir la tasa de bits erróneos, la recepción FFH-SS no-coherente combina de forma no-lineal la energía de los *chips*. El inconveniente es que esta combinación no-coherente aumenta la tasa de bits erróneos, en situaciones donde únicamente existe AWGN.

Con estos antecedentes, el objetivo de este capítulo es el diseño de un algoritmo de supresión de interferencias de banda estrecha para modulaciones FH-SS. Se pretende

que el algoritmo aumente la inmunidad del sistema sin tener que aumentar el número de *chips* por bit. La incorporación del algoritmo modifica el receptor FH-SS no-coherente mostrado en la sección 3.3, mediante la inclusión de un nuevo módulo encargado de filtrar la interferencia de banda estrecha, sin degradar la señal de comunicación recibida.

Para la construcción del filtro de supresión, se van a considerar todas las alternativas que se han recogido de la bibliografía y que se han expuesto en el capítulo 3. Dentro de los objetivos de este capítulo se encuentra: determinar el mejor dominio donde realizar la supresión y estudiar la mejor transformada que se adapte al desarrollo final.

El capítulo comienza con la presentación del receptor FH-SS modificado y el estudio de la señal obtenida en cada una de las etapas de la recepción. En la sección 4.2, se identifican las señales obtenidas a la entrada de cada módulo del receptor, y se determina el punto donde incluir el filtro de supresión. El siguiente paso es el diseño del algoritmo que integra dicho filtro. Este desarrollo constituye la sección 4.3, incluyendo el análisis inicial de requisitos y el diseño propiamente dicho. Para aclarar la forma en la que actúa el algoritmo, se muestra un ejemplo de su aplicación sobre una señal con características similares a una interferencia CWT, representando las subbandas desarrolladas en el dominio temporal y frecuencial. Este ejemplo completa la sección 4.4.

A partir de este capítulo de la tesis y en todos los posteriores, al nombrar la modulación FH-SS, se hace referencia a expansiones SFH-SS, es decir, con un único *chip* por cada bit transmitido. Cuando sea necesario hacer referencia a modulaciones con múltiples *chips* por bit transmitido, se utilizará el término expansión FFH-SS de forma expresa.

## 4.2 Diseño del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias

En la sección 3.3 se ha mostrado el esquema del receptor FH-SS no-coherente. El objetivo de esta sección es modificar dicho esquema, para incluir filtros de supresión de interferencias. Con esta modificación se persigue eliminar la mayor cantidad de energía proveniente de interferencias de banda estrecha, sin distorsionar la señal de comunicación. El receptor modificado que se propone se muestra en la figura 4.1. A continuación se explica con detalle este nuevo receptor, y las señales obtenidas a la salida de cada uno de los módulos en los que se divide.

En primer lugar hay que especificar la señal transmitida. Esta señal se supone

Figura 4.1: *Diagrama de bloques del receptor FH-SS modificado, con filtro de supresión de interferencias de banda estrecha.*

modulada con BFSK a nivel de bit y expandida en frecuencia con FH-SS. El conjunto de frecuencias que el emisor puede usar es  $f_c$ , con  $c \in [0, \dots, m - 1]$ , donde  $m$  es el número total de frecuencias del conjunto. Además se supone que la entrada al receptor  $r(t)$  está compuesta de señal FH-SS, representada por  $s(t)$ , ruido AWGN, representado por el término  $n(t)$  con Densidad Espectral de Potencia (*Spectral Power Density, SPD*)  $J_o$ , e interferencia de banda estrecha  $j(t)$ .

$$r(t) = s(t) + n(t) + j(t) \quad (4.1)$$

Los símbolos transmitidos provienen de la modulación BFSK expandida con FH-SS, expresándose matemáticamente de la forma:

$$s(t) = \sqrt{2S} \cos(2\pi(b_i/T_b + f_h + f_p)t + \phi_s) \quad (4.2)$$

donde  $\sqrt{2S}$  es la amplitud de la señal,  $\phi_s$  es la fase uniformemente distribuida en  $[0, 2\pi)$ ,  $f_p$  es la frecuencia de la portadora y  $T_b$  es el periodo de un bit, que coincide con el de un símbolo, ya que se ha considerado BFSK con expansión FH-SS. Los valores  $b_i \in [+1, 0]$  representan los dos posibles símbolos transmitidos por la BFSK. Los desplazamientos  $f_h \in [0, 2/T_b, 4/T_b, \dots, 2(\frac{m}{2} - 1)/T_b]$  están determinados por el código aleatorio utilizado en la expansión FH-SS y  $m/2$  es el número total de canales disponibles, porque la modulación a nivel de bit es BFSK y  $m$  es el número total de frecuencias utilizadas en la expansión FH. La energía de un símbolo es equivalente a la energía de un bit y su valor es  $E_b = ST_b$ .

## 4.2. Diseño del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias

---

En la ecuación 4.1, la interferencia  $j(t)$  es del tipo CWT y se expresa matemáticamente como:

$$j(t) = \sqrt{\frac{2S}{\alpha}} \cos(2\pi(f_p + \delta f_i)t + \phi_j) \quad (4.3)$$

donde  $S$  es la potencia de la señal de comunicación,  $\alpha$  es la relación entre la potencia se la señal y la interferencia,  $f_c + \delta f_i$  es la frecuencia de la interferencia y  $\phi_j$  es la fase supuesta uniformemente distribuida en el intervalo  $[0, 2\pi)$ . Con el fin de facilitar los cálculos, la frecuencia se ha expresado como la suma de  $f_c$ , frecuencia de la portadora, más un desplazamiento  $\delta f_i$ .

Los módulos en los que se divide el receptor, junto con las señales obtenidas en cada uno de ellos, son:

### 1. Correlador.

En primer lugar la señal se correlaciona con cada una de las  $m$  frecuencias provenientes de la modulación FH-SS. Para cada una de estas frecuencias se tienen dos correladores en cuadratura definidos como:

Correlador para parte real:

$$c_r(t) = V \cdot \cos(2\pi f_c t) \quad (4.4)$$

Correlador para parte imaginaria:

$$c_i(t) = V \cdot \sen(2\pi f_c t)$$

donde  $V = \frac{1}{\sqrt{2ST_b}}$  y  $f_c$  con  $c \in [0, \dots, m-1]$  son todas las posibles frecuencias transmitidas.

### 2. Filtro paso-bajo (FPB).

Durante el intervalo de tiempo correspondiente al periodo del bit transmitido  $t \in (t, t + T_b)$ , la salida del filtro paso-bajo del correlador en cuadratura sintonizado con la frecuencia del símbolo transmitido es:

$$\begin{aligned} x_{cr}(t) &= \frac{1}{T_b} \cos(\phi_s) + \frac{1}{\alpha T_b} \cos(2\pi \delta f_i t + \phi_j) + n_r(t) \\ x_{ci}(t) &= \frac{1}{T_b} \sen(\phi_s) + \frac{1}{\alpha T_b} \sen(2\pi \delta f_i t + \phi_j) + n_i(t) \end{aligned} \quad (4.6)$$

donde  $n_r$  y  $n_i$  son variables gaussianas.

Este mismo correlador, en los periodos donde ambas frecuencias no están sintonizadas, presenta la siguiente salida:

$$\begin{aligned} x_{cr}(t) &= \frac{1}{\alpha T_b} \cos(2\pi\delta f_i t + \phi_j) + n_r(t) \\ x_{ci}(t) &= \frac{1}{\alpha T_b} \text{sen}(2\pi\delta f_i t + \phi_j) + n_i(t) \end{aligned} \quad (4.8)$$

En las ecuaciones anteriores 4.6 y 4.8, el término  $\delta f_i$  representa el desplazamiento entre las frecuencias de la interferencia y los correladores, siendo este término distinto para cada pareja de correladores.

A la vista de las ecuaciones 4.6 y 4.8, se puede afirmar que la salida del filtro paso-bajo se compone de 3 señales superpuestas:

- La primera señal es binaria con valores:
  - 0, en los intervalos de tiempo donde el correlador no está sintonizado con el símbolo transmitido.
  - $\frac{1}{T_b} \cos(\phi_s)$  ó  $\frac{1}{T_b} \text{sen}(\phi_s)$ , dependiendo del correlador en cuadratura considerado. Esto términos son constantes durante todo el periodo del bit en el que las frecuencias coinciden.
- La segunda señal es un término de frecuencia constante proveniente de la interferencia,  $\frac{1}{\alpha T_b} \cos(2\pi\delta f_i t + \phi_j)$  ó  $\frac{1}{\alpha T_b} \text{sen}(2\pi\delta f_i t + \phi_j)$ .
- La terera señal es  $n_r$  ó  $n_i$ , que representan variables gaussianas.

### 3. El muestreador.

La salida de los correladores, después de ser filtrada paso-bajo, se muestrea y se divide en bloques. En este momento la variable temporal independiente  $t$  pasa a ser el índice del vector  $n$ . Los dos bloques provenientes de los dos correladores en cuadratura forman un vector complejo, con un número de puntos igual al tamaño del bloque utilizado por la transformada, dicho tamaño se definirá con mayor detalle en la sección 4.3.

Para una pareja de correladores  $c_r(t)$  y  $c_i(t)$ , dentro de los periodos de bit en los que la frecuencia queda sintonizada con la frecuencia del símbolo transmitido, el vector complejo muestreado es:

$$\begin{aligned} x_{cr}(n) &= \frac{1}{T_b} \cos(\phi_s) + \frac{1}{\alpha T_b} \cos(2\pi\delta f_i n / f_s + \phi_j) + n_r(n) \\ x_{ci}(n) &= \frac{1}{T_b} \text{sen}(\phi_s) + \frac{1}{\alpha T_b} \text{sen}(2\pi\delta f_i n / f_s + \phi_j) + n_i(n) \end{aligned} \quad (4.10)$$

donde  $n$  es el índice del vector y  $f_s$  es la frecuencia de muestreo.

Para estos mismos correladores, el vector en los intervalos donde no están sintonizados con el símbolo transmitido es:

## 4.2. Diseño del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias

---

$$x_{cr}(n) = \frac{1}{\alpha T_b} \cos(2\pi\delta f_i n / f_s + \phi_j) + n_r(n) \quad (4.12)$$

$$x_{ci}(n) = \frac{1}{\alpha T_b} \text{sen}(2\pi\delta f_i n / f_s + \phi_j) + n_i(n)$$

A partir de las ecuaciones 4.10 y 4.12, se observa que el término  $\frac{1}{\alpha T_b} \cos(2\pi\delta f_i n / f_s + \phi_j)$  ó  $\frac{1}{\alpha T_b} \text{sen}(2\pi\delta f_i n / f_s + \phi_j)$  generado por la interferencia obstaculiza la correcta recepción de los bits transmitidos.

El módulo de supresión va a ser el encargado de eliminar este término, cuya frecuencia  $2\pi\delta f_i n / f_s$  es totalmente aleatoria, apareciendo añadido en la salida de ambos correladores en cuadratura. Este filtrado debe realizarse antes de continuar con el proceso de recepción de la señal de forma no-coherente.

### 4. Transformación.

Una vez muestreada la señal, el vector complejo obtenido para cada pareja de correladores se denota:  $X_c$ , con  $c \in [0, \dots, m - 1]$ . Este vector constituye la entrada al filtro de excisión en el dominio de la transformada. Cada uno de los componentes de este vector complejo proviene de uno de los dos correladores en cuadratura, representándose como:  $X_c = X_{cr} + jX_{ci}$ . Los vectores tienen un número de puntos que engloba varios símbolos transmitidos, porque su tamaño coincide con el bloque utilizado por la transformada.

Los puntos del vector forman una secuencia, cuyos valores coinciden con los de la ecuación 4.10, cuando el índice  $n$  está comprendido dentro del periodo de un símbolo con frecuencia sintonizada a la del correlador. Pero si las frecuencias no son iguales, los valores vienen dados por la ecuación 4.12.

Al final, tal como se ha explicado en el punto dedicado al filtro paso-bajo, la secuencia  $X_c$  se puede considerar como la suma de una secuencia binaria, con intervalos a nivel alto múltiplos del periodo de un bit, otra secuencia senoidal de frecuencia  $2\pi\delta f_i n / f_s$  y una secuencia de valores con distribución gaussiana. El filtro de supresión es el encargado de eliminar el segundo término, como se describe con más detalle en la sección 4.3.

### 5. Sumador.

El siguiente módulo suma todos los puntos correspondientes a un mismo símbolo. La salida de este módulo tiene la forma:

$$U_{cr} = \sum_M x_{cr}(n) \quad (4.14)$$

$$U_{ci} = \sum_M x_{ci}(n)$$

donde  $M$  es el número de puntos que forman el símbolo. Suponiendo  $\sum_M = T_b$  y aproximando el valor  $\sum_M \cos(n) \simeq \int_0^{T_b} \cos(x) dx$ , el sumatorio 4.14 se puede expresar para los intervalos con frecuencias coincidentes entre símbolo y correladores como:

$$U_{cr} = \cos(\phi_s) + \frac{1}{\sqrt{\alpha}} \text{sinc}(\mu) \cos(\mu + \phi_j) + N_c \quad (4.16)$$

$$U_{ci} = \text{sen}(\phi_s) + \frac{1}{\sqrt{\alpha}} \text{sinc}(\mu) \text{sen}(\mu + \phi_j) + N_i$$

donde  $\mu = \delta f_i T_b$ . Además los términos  $N_c$  y  $N_i$  representan distribuciones gaussianas.

Si la frecuencia del símbolo transmitido no coincide con la frecuencia del correlador, la salida del sumatorio es:

$$U_{cr} = \frac{1}{\sqrt{\alpha}} \text{sinc}(\mu) \cos(\mu + \phi_j) + N_c \quad (4.18)$$

$$U_{ci} = \frac{1}{\sqrt{\alpha}} \text{sinc}(\mu) \text{sen}(\mu + \phi_j) + N_i$$

En este capítulo se supone que la sincronización entre el emisor y el receptor a nivel de símbolos transmitidos es perfecta. Las técnicas para conseguir la sincronización se pueden consultar en [59].

#### 6. Combinador no-coherente.

Las salidas de los sumadores en cuadratura se combinan para eliminar la dependencia con la fase de la portadora. La pareja  $U_{cr}$  y  $U_{ci}$  representa las dos componentes de una señal compleja  $U_c$ :

$$U_c = e^{j\phi_s} + \frac{1}{\sqrt{\alpha}} \text{sinc}(\mu) e^{j\phi_j} + Z \quad (4.20)$$

donde  $Z = N_c + jN_i$ . La salida de este módulo es la variable de decisión  $|U_c|^2$ , que elimina la dependencia con la fase  $\phi_s$ .

#### 7. Detector.

Este último módulo determina el símbolo transmitido a partir de las entradas  $|U_c|^2$ . Para cada bit transmitido se recoge un valor de salida asociado a cada una de las  $m$  frecuencias de la expansión, aunque solamente es necesario comparar los valores  $|U_c|^2$  correspondientes a las 2 frecuencias que utiliza la modulación BFSK, desplazadas hasta el canal indicado por el código PN.

### 4.3. Diseño del filtro de supresión de interferencias

---

El receptor FH propuesto está basado en [15]. Sin embargo, el receptor que se describe en [15] tiene como objetivo la detección no-coherente de señales DS-SS y la supresión de interferencias de banda estrecha. En dicha referencia se plantea el diseño del módulo de filtrado, tanto con algoritmos en el dominio de la transformada de Fourier, como mediante algoritmos adaptativos en el dominio temporal. El receptor introducido en [15] presenta las siguientes diferencias:

- Contiene una única pareja de correladores en cuadratura.
- El orden en el que se realiza el cuadrado y el sumatorio de la señal discreta está invertido. Esta inversión de bloques se debe a que en el caso DS-SS sólo hay una portadora, es decir, no hay interferencia proveniente de las portadoras de canales contiguos, y la señal puede elevarse al cuadrado antes de ser integrada sobre el periodo del símbolo.

## 4.3 Diseño del filtro de supresión de interferencias

En esta sección se presenta el algoritmo de filtrado propuesto. Como ya se ha comentado en la sección 4.2, este módulo de filtrado integrado en el receptor FH-SS debe eliminar la mayor cantidad posible de energía correspondiente a la interferencia, degradando mínimamente la señal de comunicación.

Para comenzar el proceso de diseño, primero se define el tipo de interferencia frente a la que se desea aumentar la fiabilidad del sistema FH-SS. Posteriormente, se analizan las opciones que se han recogido en la bibliografía, la mayoría de ellas aplicadas sobre modulaciones DS-SS, estudiando su posible aplicación al sistema planteado. En último lugar, se especifica el algoritmo desarrollado, mostrando sus características y el pseudocódigo para su programación.

### 4.3.1 Hipótesis iniciales

En el desarrollo del algoritmo se va a suponer que el ancho de banda ocupado por la interferencia es menor, que el ocupado por un canal correspondiente a la expansión FH-SS. La interferencia se supone de tipo CWT, tal y como se describe en la sección 3.3.3. También se asume que la zona de frecuencia ocupada por la interferencia puede variar de forma aleatoria, estando la interferencia compuesta por múltiples tonos. Cada uno de ellos afecta a un canal distinto de los ocupados por la señal modulada en BFSK y expandida con FH-SS.

También se exige que el algoritmo tenga un coste computacional reducido, de cara a su posible implementación y aplicación en sistemas de comunicación reales, que

utilicen una expansión FH-SS. En concreto se persigue su posible aplicación sobre un dispositivo programable del tipo Procesador Digital de Señal (*Digital signal processors, DSP*).

Estas suposiciones o hipótesis iniciales no son excesivamente restrictivas, ya que se ajustan a las interferencias encontradas en canales reales, como el presentado en el capítulo 6. Además, no quitan generalidad al estudio, dado que son tomadas habitualmente en trabajos similares [59], [72].

### 4.3.2 Análisis del algoritmo

En este apartado se plantean y analizan las posibles alternativas que se pueden tomar en el diseño del algoritmo. Estas cuestiones se ven afectadas por el tipo de modulación utilizada y las condiciones de la interferencia supuestas en el apartado 4.3.1 anterior. Las consideraciones a tener en cuenta son:

1. Dominio de la supresión.

La primera cuestión que se plantea en el desarrollo del algoritmo es la elección del dominio donde realizar la supresión de la interferencia. Como se ha expuesto en el capítulo 3, las técnicas basadas en el dominio temporal necesitan algoritmos adaptativos para ajustar los coeficientes de los filtros. El problema que aparece es que estos algoritmos necesitan un cierto intervalo de tiempo, para que los coeficientes converjan hacia los valores óptimos, siendo difícil su utilización en situaciones donde la interferencia no es estacionaria, o varía su posición dentro del espacio de frecuencias rápidamente [28]. Por otro lado, los algoritmos basados en el dominio transformado tienen la ventaja de su fácil implementación y su rápido ajuste frente a las variaciones de la interferencia [40]. Estas dos razones hacen que el algoritmo propuesto se base en el dominio de la transformada.

2. Relación entre los anchos de banda.

La segunda consideración importante es la relación entre el ancho de banda ocupado por la señal FH-SS y el ocupado por la interferencia. Como ya se mencionó en el capítulo 3, la modulación DS-SS expande la señal instantáneamente sobre toda la zona del espectro disponible, siendo la diferencia entre el ancho de banda ocupado por la interferencia y la señal DS muy grande. Esto permite que al eliminar la zona del espectro afectada por la interferencia, no se degrade fuertemente la señal DS. Sin embargo no sucede lo mismo al considerar la modulación FH, cuyo espectro instantáneo ocupa un intervalo de frecuencias estrecho, siendo necesario obtener una buena resolución en frecuencias que permita distinguir ambas señales. Esta condición depende de la transformada y de la expansión FH:

### 4.3. Diseño del filtro de supresión de interferencias

---

- La dependencia con la transformada:

Para llevar a cabo implementaciones reales, con FFT o con DWT, es necesario que la señal de entrada tenga una longitud finita [59], [19], con un número determinado de muestras que se denomina  $N$ . La señal de entrada se puede considerar como un bloque de elementos con un tamaño fijo  $N$ . El tamaño del bloque dependerá de la implementación final que se haga de la transformada. Aunque es cierto que cuanto mayor sea el tamaño del bloque, mejor resolución frecuencial se obtiene, como se desprende directamente de las propiedades de la FFT [59], o del hecho de que la DWT puede llegar hasta un nivel de descomposición  $J = \log_2(N)$  [78].

Además la transformada debe concentrar la interferencia en un pequeño número de puntos, debido a la mayor degradación que supone para la señal FH eliminar zonas del espectro. Por lo tanto, es necesario que el algoritmo sea muy selectivo en frecuencia.

- La dependencia con la modulación:

Para conseguir mayor resolución, la expansión de la energía que realiza el código PN debe ser equitativa entre todos los canales, ya que esto produce un ensanchamiento del espectro de la señal, frente al espectro ocupado por la interferencia.

### 3. La transformada.

La tercera cuestión es la transformada elegida. Las alternativas más adecuadas, como se ha comentado en el apartado 3.4.2 del capítulo anterior, son la FFT y la WT. Como primera alternativa se puede considerar la utilización de la FFT, dado que es una transformada para la que existen algoritmos rápidos, pero el principal problema que presenta es la dispersión de la energía de la interferencia en forma de  $\text{sinc}(x)$ , que se produce al utilizar un tamaño de bloque finito, por el fenómeno del enventanado.

Una segunda posibilidad es la WPT, que permite optimizar el conjunto de coeficientes con el que se representa la señal, como ya se ha expuesto en el apartado 2.5. Con esta transformada se pretende detectar y confinar en una única subbanda la energía de la interferencia. Pero dependiendo del tipo de descomposición que se haga pueden aparecer problemas en las zonas de corte entre subbandas, como se discute en [80]. Estos problemas aparecen porque siempre existe un pequeño solapamiento entre los espectros de los filtros, dado que no son ideales. Si la interferencia se sitúa exactamente en la zona de corte de ambas subbandas, ninguna de ellas la contendrá totalmente, como ya se expuso en el apartado 3.4.2 para modulaciones DS-SS.

Para evitar que la interferencia quede dividida entre dos subbandas, existen dos posibles soluciones consideradas en la bibliografía:

- Una primera solución presentada en [80] y comentada en el apartado 3.4.2, utiliza filtros diádicos o triádicos en cada nivel de la transformada, seleccionando la descomposición que mejor se ajusta al espectro de la señal. Este algoritmo tiene un coste computacional de orden  $O(N)$  [80], siendo  $N$  el número de puntos del vector de entrada a la transformación.
- Una segunda solución planteada en [35] y comentada en el apartado 2.5, desplaza la señal en frecuencia para buscar la mejor base con la que representarla. Esta alternativa no ha sido desarrollada para la eliminación de interferencias en modulaciones SS, y calcula todo el árbol de descomposición para cada posible desplazamiento en frecuencia de la señal. Con este método se obtiene un algoritmo de coste de orden  $O(N^2)$ .

Los inconvenientes de estas dos soluciones son:

- La primera propuesta evita el problema de que la interferencia quede situada entre dos subbandas, mediante la doble división diádica o triádica del árbol de descomposición. Pero el algoritmo no puede asegurar que la interferencia quede totalmente centrada en la zona paso-banda de uno de los filtros. Así, dependiendo de la frecuencia en la que se sitúe, parte de su energía puede quedar en subbandas vecinas.
- La segunda solución presenta un coste computacional muy elevado. La razón es que el algoritmo no está diseñado para la eliminación de interferencias que ocupan una sola zona del espectro, su planteamiento es más general. El algoritmo aborda el problema de la búsqueda de la mejor base con la que representar una señal, utilizando descomposiciones en WPT.

Ya que estos dos algoritmos presentan inconvenientes, el objetivo del siguiente apartado 4.3.3 es el diseño de un nuevo algoritmo, que se ajuste mejor a las condiciones dadas por el tipo de interferencia, asumida en el apartado de hipótesis 4.3.1, y a las modulaciones FH-SS. Este algoritmo se va a basar en la UWPT, vista en la sección 2.6, que permite realizar una transformación invertida desde los niveles de mayor resolución frecuencial, hacia los de menor resolución. Con esta transformación invertida se puede centrar la interferencia en la zona paso-banda de los filtros con un bajo coste computacional, como se demuestra en la próxima sección 4.3.3.

### 4.3.3 Algoritmo basado en Transformadas Wavelet Sobremuestreadas

En este apartado se propone un nuevo algoritmo, para la detección y concentración de la energía de la interferencia en una subbanda mediante la UWPT, transformada introducida en el apartado 2.6. Este algoritmo utiliza desplazamientos de la señal en

### 4.3. Diseño del filtro de supresión de interferencias

---

frecuencia, para obtener la mejor representación en el dominio de la transformada con un bajo coste computacional. Con el fin de optimizar la eliminación de energía de la interferencia, el algoritmo la sitúa en las posiciones de frecuencia centradas en la zona paso-banda del espectro de uno de los filtros, como se muestra en la figura 4.2. Para ello, previamente se requiere calcular el desplazamiento en frecuencia óptimo que sitúa la interferencia en esta zona.

También hubiera sido posible desplazar los filtros en vez de la señal, multiplicando cada filtro por el factor  $e^{i2\pi dn/L}$ , siendo  $d$  el desplazamiento,  $L$  el número de coeficientes del filtro y  $n$  el índice de la multiplicación. Pero al realizar esta operación los coeficientes del filtro pasan a ser valores complejos, lo que complica el diseño del algoritmo.

Figura 4.2: *Desplazamiento en frecuencia y centrado de la señal sobre una subbanda.*

Los filtros en la transformación UWPT, como se comentó en la sección 2.6, cambian su ancho de banda en cada nivel de resolución  $j$ , tal como se muestra en la figura 4.3. Si se comienza el desarrollo desde el nivel de menor resolución frecuencial, nivel  $j = 1$  donde los filtros ocupan cada uno de ellos la mitad del espectro, existen muchos desplazamientos posibles, para centrar la interferencia en la zona paso-banda de  $H(f)$  ó  $G(f)$ , esta propiedad se puede comprobar en la figura 4.3. Para seleccionar el desplazamiento que sitúa la interferencia en la zona paso-banda, es necesario primero desplazar la señal multiplicándola por el factor  $e^{i2\pi dn/N}$ , siendo  $d$  el desplazamiento,  $N$  el número total de puntos que contiene el vector de entrada y  $n$  el índice de la multiplicación. A continuación, la señal desplazada se filtra con  $H$  y  $G$ . Estas dos operaciones hay que repetirlas para todos los desplazamientos posibles. Por consi-

guiente, si el algoritmo comenzase en  $j = 1$  el coste computacional sería muy elevado, del orden del obtenido en [35].

Por el contrario, si el algoritmo comienza en un nivel de resolución frecuencial mayor, en la figura 4.3 se corresponde con  $j = 4$ , el número de desplazamientos que hay que aplicar es menor, porque el ancho de banda de las zonas paso-banda de los filtros en este nivel es menor.

*Figura 4.3: Representación del espectro de los filtros de Daubechies con 4 coeficientes para cuatro niveles de resolución  $j \in [4, \dots, 1]$ . La figura muestra el ancho de banda de los filtros en diferentes niveles de resolución.*

Para reducir el número de desplazamientos, la descomposición comienza en un nivel de mayor resolución frecuencial, nivel donde los filtros tienen intervalos paso-banda de menor tamaño.

El nivel donde comienza la descomposición invertida es el de mayor resolución frecuencial, dentro del desarrollo normal del árbol que forma la WPT. Pero para llevar a cabo la descomposición invertida, en los términos comentados, no es posible utilizar

### 4.3. Diseño del filtro de supresión de interferencias

---

dicha transformada. Esta es la razón por la que se ha utilizado la UWPT, ya que al no diezmar la señal, sí que permite realizar el desarrollo invertido.

Como conclusión a la discusión sobre las condiciones iniciales de diseño, hay que recapitular las decisiones tomadas:

1. Sobre la señal de entrada, compuesta por un bloque de  $N$  puntos, se van aplicar desplazamientos en frecuencia, con los que situar la interferencia en la zona paso-banda de uno de los filtros utilizados en la transformación.
2. La descomposición comienza en el nivel de mayor resolución frecuencial, con el fin de reducir el número de desplazamientos posibles de la señal.
3. La transformada utilizada en la descomposición es la UWPT, porque permite llevarla a cabo de forma invertida.

Una vez tomas las decisiones iniciales, el próximo apartado se centra en su diseño del algoritmo. Para comenzar se propone el algoritmo de análisis y posteriormente se determina el de síntesis.

#### Algoritmo de análisis

La descomposición de análisis comienza en el nivel de mayor resolución frecuencial, denominado nivel  $J$ . Los demás niveles son tratados de forma diferente por lo que se va a distinguir entre dos casos:

- Caso 1: Descomposición del nivel  $J$ .

En este nivel el algoritmo recoge el vector complejo  $X_J$ , proveniente de la salida del filtro paso-bajo correspondiente a una pareja de correladores en cuadratura:  $X_J = X_{Jr} + jX_{Ji}$ . Este vector  $X_J$  se corresponde con el mostrado y explicado en la ecuación 4.10.

Para conseguir centrar la interferencia en la zona paso-banda de uno de los filtros, el vector de entrada  $X_J$  se modifica mediante desplazamientos en frecuencia. Los desplazamientos se realizan mediante la multiplicación del vector  $X_J$ , con la exponencial  $e^{i2\pi dn/N}$ . El conjunto de todos los desplazamientos en frecuencia en el nivel  $J$  viene dado por el ancho de las zonas paso-banda de los filtros. Este conjunto denotado  $d(J)$  tiene sus valores comprendidos en  $d(J) \in [0, \dots, B_J]$ , siendo  $B_J$  el ancho de las zonas paso-banda de los filtros correspondientes en este nivel. Gráficamente el intervalo de desplazamientos  $d(J)$  se representa en la figura 4.4

Una vez aplicado un desplazamiento  $d(J)$  al vector de entrada, el algoritmo lo descompone de forma diádica mediante convolución con los filtros definidos para

Figura 4.4: *Espectro de los filtros de Daubechies con 4 coeficientes para el nivel  $j = 4$ . La gráfica muestra el intervalo de desplazamientos  $d(j)$  que se aplican en el caso  $J = 4$ .*

ese nivel. Dichos filtros son:  $h_{0,j}(n)$ , el filtro paso-bajo, y  $g_{0,j}(n)$ , el filtro paso-alto, respectivamente. Estos dos filtros están formados por los coeficientes de los filtros paso-alto y paso-bajo,  $h_0(n)$  y  $g_0(n)$ , con  $2^{j-1} - 1$  ceros intercalados entre sus coeficientes. Para un nivel genérico  $j$  esta relación, tal como se introdujo en la sección 2.6, es:

$$g_{0,j}(n) = [g_0(0), \underbrace{0, \dots, 0}_{2^{j-1}-1}, g_0(1), 0, \dots, 0, g_0(L_1 - 1)] \quad (4.21)$$

$$h_{0,j}(n) = [h_0(0), \underbrace{0, \dots, 0}_{2^{j-1}-1}, h_0(1), 0, \dots, 0, h_0(L_1 - 1)] \quad (4.22)$$

Hay que destacar que los filtros del nivel  $j = 1$  coinciden con los filtros definidos para la TW,  $h_{0,1}(n) = h_0(n)$  y  $g_{0,1}(n) = g_0(n)$ .

Como resultado de esta operación se obtienen los vectores  $W_{J,0,d(J)}$  y  $W_{J,1,d(J)}$ . El primero  $W_{J,0,d(J)}$  correspondiente a la subbanda paso-baja y el segundo  $W_{J,1,d(J)}$  a la paso-alta, los cuales contienen el mismo número de puntos que  $X_J$ , ya que la UWPT no diezma la señal.

Para cada desplazamiento  $d(J)$ , se obtienen las dos subbandas  $W_{J,0,d(J)}$  y  $W_{J,1,d(J)}$ , cuyas ecuaciones son:

### 4.3. Diseño del filtro de supresión de interferencias

---

#### Vector paso-bajo

$$w_{J,0,d(J)}(n) = \sum_{l=0}^{L-1} h_0(l)(x_J(n - 2^{J-1}l)e^{i2\pi(n-2^{J-1}l)d(J)/N}) \quad (4.23)$$

#### Vector paso-alto

$$w_{J,1,d(J)}(n) = \sum_{l=0}^{L-1} g_0(l)(x_J(n - 2^{J-1}l)e^{i2\pi(n-2^{J-1}l)d(J)/N}) \quad (4.24)$$

siendo  $n \in [0, \dots, N-1]$ ,  $d(J)$  el desplazamiento,  $L$  es el número de coeficientes de los filtros,  $X_J$  es el vector complejo de entrada y  $N$  es el número total de puntos de dicho vector.

Para seleccionar el desplazamiento  $d(J)$ , que sitúa la interferencia centrada en la zona paso-banda de uno de los filtros, se utiliza la diferencia de energía entre los vectores paso-alto y paso-bajo:

$$\Delta E_{J,d(J)} = \left| \sum_{n=0}^{N-1} \|w_{J,1,d(J)}(n)\|^2 - \sum_{n=0}^{N-1} \|w_{J,0,d(J)}(n)\|^2 \right| \quad (4.25)$$

El valor de  $d(J)$ , que maximiza la diferencia de energía entre las dos subbandas para este nivel  $J$ , se denomina  $D(J)$  y se define:

$$D(J) = \{d(J) | \max(\Delta E_{J,d(J)}), \quad d(J) \in [0, \dots, B_J]\} \quad (4.26)$$

Esta diferencia de energía será máxima, cuando la interferencia esté centrada en la zona paso-banda de uno de los filtros.

Una vez determinado  $D(J)$ , el algoritmo selecciona la pareja de vectores paso-bajo y paso-alto correspondientes a dicho desplazamiento  $W_{J,0,D(J)}$  y  $W_{J,1,D(J)}$ . Entre los dos vectores considerados, el algoritmo continúa descomponiendo el que tiene mayor energía. El vector descartado, junto con el desplazamiento realizado  $D(J)$ , se almacenan de cara a su utilización en el proceso de síntesis, que se realiza posteriormente.

Es posible que todos los valores  $\Delta E_{J,d(J)}$  estén muy próximos, y el valor máximo sea menor que un cierto umbral. En este caso, la interferencia presenta un ancho de banda mayor que la zona paso-banda de los filtros en ese nivel de resolución. Ante esta situación, el algoritmo comienza en el nivel  $J-1$ , puesto que un único filtro en el nivel  $J$  no es capaz de concentrar toda la energía de la interferencia.

En estas condiciones el algoritmo elige el nivel de menor resolución frecuencial  $J - 1$  como nivel inicial, siendo la zona paso-banda de los filtros más ancha en este nivel.

Hay que destacar que el desarrollo del algoritmo está condicionado por dos parámetros:

- El primero de ellos es el incremento del desplazamiento en frecuencia, definido como  $\Delta d$ . Este factor es el paso de resolución en la selección de los valores  $d(J)$ . Dependiendo de este parámetro se obtiene mayor o menor precisión en la alineación entre filtros e interferencia. Cuanto menor sea más precisión se consigue, pero el número de iteraciones de filtrado en el nivel  $J$  aumenta y afecta al coste del algoritmo. Debido a que el parámetro afecta a la precisión en el alineado y al coste, su elección se basa en una relación de compromiso
- El segundo parámetro es el nivel inicial del desarrollo  $J$ . Este término puede estar comprendido en el rango  $J \in [1, \dots, N]$ , siendo  $N$  número de elementos en el vector de entrada. El valor  $J$  define el tamaño del ancho de banda de los filtros en el primer nivel. Pero si aumenta este ancho de banda, el número de iteraciones en dicho nivel también aumenta, es decir, se incrementa el coste. Por otro lado, si disminuye el ancho de banda, puede ser demasiado estrecho para contener totalmente la interferencia en una única subbanda. Este factor debe elegirse de nuevo mediante una relación de compromiso entre el coste del algoritmo, y el ancho de banda mínimo que engloba la mayor parte de la energía.

Ambos parámetros se estudian con mayor profundidad en el capítulo 6, donde se comprueba específicamente el rendimiento del algoritmo propuesto.

- Caso 2: Descomposición del nivel  $j \in [J - 1, \dots, 1]$ .

Una vez concluido el nivel  $J$ , el algoritmo continua de forma iterativa con el siguiente nivel de resolución. El nuevo vector de entrada es el que tiene mayor energía entre  $W_{J,0,D(J)}$  y  $W_{J,1,D(J)}$ . Este vector se representa en la nueva iteración como  $X_{J-1}$ .

En los siguientes niveles de resolución, debido al alineado de las zonas de corte y las zonas paso-banda de los filtros, correspondientes a dos niveles de descomposición consecutivos, la interferencia sólo puede estar en dos posiciones:

1. En una zona de corte entre las dos subbandas.
2. En la zona paso-banda de uno de los filtros.

Esta propiedad se puede observar en la gráfica 4.5, donde se representa el espectro de los filtros de Daubechies con 4 coeficientes para cuatro niveles de resolución.

### 4.3. Diseño del filtro de supresión de interferencias

---

En dicha gráfica se comprueba el alineado de las zonas de corte y las zonas paso-banda de los filtros en diferentes niveles de resolución. Al centrar la interferencia en la zona paso-banda de uno de los filtros en el nivel  $J$ , en el nivel  $J - 1$  sólo puede encontrarse en dos posiciones diferentes dentro del espectro de frecuencias de la señal.

Es posible conocer exactamente su posición dependiendo de la subbanda de la que procede  $X_{J-1}$ :

1. Si  $X_{J-1}$  procede de  $W_{J,0,D(J)}$ , subbanda paso-baja obtenida con  $h_{0,J}(n)$ , la interferencia está en la zona paso-banda de un filtro en el nivel  $J - 1$ .
2. Si  $X_{J-1}$  procede de  $W_{J,1,D(J)}$ , subbanda paso-alta obtenida con  $g_{0,J}(n)$ , la interferencia está en la zona de corte de los filtros en el nivel  $J - 1$ .

Esta propiedad se puede comprobar en la figura 4.5. El algoritmo desplaza la señal en frecuencia con valores  $D(J - 1) = 0$ , si  $X_{J-1}$  procede de  $W_{J,0,D(J)}$ , o  $D(J - 1) = B_{J-2}/2$ , si  $X_{J-1}$  procede de  $W_{J,1,D(J)}$ . De esta manera, sólo es necesario un desplazamiento para seguir manteniendo la señal alineada con las zonas paso-banda de los filtros en el nuevo nivel  $J - 1$ .

Una vez desplazada, la señal  $X_{J-1}$  se descompone en las nuevas subbandas  $W_{J-1,0,D(J-1)}$  y  $W_{J-1,1,D(J-1)}$ . Pero en este nivel sólo hay que filtrar la señal una única vez, es decir, sólo hay una pareja de vectores  $W_{J-1,0,D(J-1)}$  y  $W_{J-1,1,D(J-1)}$ .

Las ecuaciones que calculan los vectores paso-alto y paso-bajo son las mismas que las del nivel  $J$  de resolución, pero restringiendo el valor  $D(J - 1)$ . Las ecuaciones en un nivel genérico  $j \in [J - 1, \dots, 1]$  son:

$$w_{j,0,D(j)}(n) = \sum_{l=0}^{L-1} h_0(l)(x_j(n - 2^{j-1}l)e^{i2\pi(n-2^{j-1}l)D(j)/N}) \quad (4.27)$$

$$w_{j,1,D(j)}(n) = \sum_{l=0}^{L-1} g_0(l)(x_j(n - 2^{j-1}l)e^{i2\pi(n-2^{j-1}l)D(j)/N}) \quad (4.28)$$

Entre los dos vectores se elige aquel que concentra la interferencia y se sigue iterando sobre él. El vector descartado, junto con el desplazamiento aplicado en este nivel  $D(J - 1)$ , se almacenan de cara a la reconstrucción.

Los demás niveles de resolución  $j \in [J - 2, \dots, 1]$  se calculan de la misma forma que  $J - 1$ . Al considerar el nivel  $j$ , si  $X_j$  proviene de  $W_{j+1,0,D(j+1)}$ , entonces  $D(j) = 0$ , y si  $X_j$  proviene de  $W_{j+1,1,D(j+1)}$ , entonces  $D(j) = B_j/2$ . Aplicando las ecuaciones 4.27 y 4.28, se obtienen las nuevas subbandas  $W_{j,0,D(j)}$  y  $W_{j,1,D(j)}$ , de las cuales se selecciona la mayor y se almacena la descartada junto con  $D(j)$ , para el proceso de reconstrucción.

Figura 4.5: *Representación del espectro de los filtros de Daubechies con 4 coeficientes para cuatro niveles de resolución  $j \in [4, \dots, 1]$ . La figura muestra el alineado de las zonas de corte y paso-banda de los filtros en niveles de resolución consecutivos.*

Al llegar al nivel de menor resolución frecuencial, nivel 1, la interferencia queda situada en una frecuencia cercana a 0 ó  $\pi$ , ya que estas son las zonas pasa-banda para los filtros  $h_0(n)$  y  $g_0(n)$ . En este nivel de resolución, el vector con mayor energía entre los dos vectores  $W_{1,0,D(1)}$ ,  $W_{1,1,D(1)}$  es el que engloba la interferencia. El algoritmo de análisis queda resumido en forma de pseudocódigo en la figura 4.6.

En la figura 4.7, se muestra un ejemplo de la aplicación del algoritmo de análisis. La gráfica representa en primer lugar el espectro de la señal en cada nivel de resolución. La señal original elegida tiene un espectro plano sobre todo el rango de frecuencias, junto con una interferencia de banda estrecha superpuesta en una determinada frecuencia. En cada nivel de resolución el espectro de esta señal se va modificando al aplicar los desplazamientos, que centran la interferencia, y los filtros, que van diezmado zonas del espectro. Como resultado de la descomposición, en el nivel  $J = 1$  la interferencia queda centrada en la zona paso-banda del  $G(f)$ . La figura también muestra el esquema

Figura 4.6: *Pseudoódigo del algoritmo de análisis. En el algoritmo la función desplazamiento( $X, d$ ), aplica el desplazamiento en frecuencia  $d$  sobre la señal  $X$ . La función conv( $X_1, X_2$ ) realiza la convolución de las dos señales. Por último la función energía( $W$ ) calcula la energía del vector de entrada, como se definió en la ecuación 4.25, y la función valor de  $d(J)$  que maximiza  $|\Delta E[d(J)]|$  devuelve el índice  $d(J)$ , con mayor diferencia de energía entre las dos subbandas. El algoritmo distingue entre el primer nivel de resolución  $J$  y el resto  $j \in [J - 1, \dots, 1]$ .*

de la transformación, indicando en cada nivel el vector de entrada y la subbanda de salida seleccionada. En concreto la descomposición realizada elige en cada iteración las siguientes subbandas:  $W_{4,1,D(4)} = X_3$ ,  $W_{3,0,D(3)} = X_2$ ,  $W_{2,1,D(2)} = X_1$ ,  $W_{1,1,D(1)} = X_0$ .

#### Algoritmo de síntesis

Finalizadas las iteraciones de análisis, comienza el proceso de síntesis. En la reconstrucción, primero se anulan todos los componentes del vector  $X_0$ , porque son los que concentran la interferencia. Posteriormente se aplican las ecuaciones de síntesis correspondientes, que dependen del vector almacenado en el proceso de análisis:

- Si el vector almacenado en el nivel  $j$  fue  $W_{j,1,D(j)}$ , la síntesis es:

Figura 4.7: *Ejemplo de descomposición con el algoritmo de supresión de interferencias propuesto. La figura muestra los espectros de la señal y los filtros en cada nivel de resolución, comenzando en  $J = 4$ . Los filtros son los de Daubechies 4. También se muestra de forma esquemática las subbandas, que el algoritmo ha elegido en la descomposición y los vectores de entrada y salida en cada nivel.*

$$x_j(n) = \sum_{l=0}^{L-1} h_1(l)x_{j-1}(n - 2^{j-1}l) + \sum_{l=0}^{L-1} g_1(l)w_{j,1,D(j)}(n - 2^{j-1}l) \quad (4.29)$$

- Si el vector almacenado en el nivel  $j$  fue  $W_{j,0,D(j)}$ , la ecuación es:

$$x_j(n) = \sum_{l=0}^{L-1} h_1(l)(w_{j,0,D(j)}(n - 2^{j-1}l)) + \sum_{l=0}^{L-1} g_1(l)x_{j-1}(n - 2^{j-1}l) \quad (4.30)$$

donde los filtros  $h_1$  y  $g_1$  son los correspondientes filtros pasa-bajo y paso-alto de síntesis.

En el proceso de síntesis es necesario cancelar los retrasos introducidos por los filtros, para poder combinar las subbandas en las ecuaciones 4.29 y 4.30. Esta corrección

### 4.3. Diseño del filtro de supresión de interferencias

---

del desfase hay que realizarla antes de cancelar el desplazamiento en frecuencia, que se anula mediante la expresión:

$$x_j(n) = x_j(n)e^{i2\pi n(-D(j))/N} \quad (4.31)$$

La multiplicación con el término  $e^{i2\pi n(-D(j))/N}$  lleva la señal resultante, hasta su posición original en frecuencias con un desplazamiento  $-D(j)$ . El vector resultante en el proceso de síntesis del nivel  $j$ ,  $X_j$ , junto con el vector descartado y almacenado en el proceso de análisis para el nivel  $j + 1$ ,  $W_{j+1,1,D(j)}$  ó  $W_{j+1,0,D(j)}$  se convierten en los nuevos vectores de entrada, en el siguiente paso de síntesis del nivel  $j + 1$ . Operando sucesivamente sobre todos los niveles de resolución, se consigue reconstruir la señal original sin interferencia. El pseudocódigo del algoritmo de síntesis se puede ver en la figura 4.8.

Figura 4.8: *Pseudocódigo del algoritmo de síntesis. En el algoritmo la función desplazamiento( $X,d$ ) aplica el desplazamiento en frecuencia  $d$  sobre la señal  $X$ . La función conv( $X_1, X_2$ ) realiza la convolución de las dos señales. Por último la función cancelar-desfase( $X$ ) cancela el desfase introducido por los filtros sobre el vector  $X$ . Este desfase depende de los filtros utilizados y del nivel de resolución considerado. Al igual que en el caso de análisis, el algoritmo distingue entre el primer nivel de resolución  $J$  y el resto  $j \in [J - 1, \dots, 1]$ . El algoritmo aplica directamente las ecuaciones de reconstrucción propuestas en 4.29 y 4.30.*

## 4.4 Ejemplo de aplicación del algoritmo propuesto: validación

Para comprobar de forma experimental el funcionamiento del algoritmo, en esta sección se muestra su aplicación sobre un vector de entrada, que simula la combinación de una señal FH-SS y una interferencia de banda estrecha. Este ejemplo permite comprobar la descomposición de la señal original, en cada paso de iteración, y la concentración de la energía de la interferencia en una única subbanda.

La señal simulada está formada por la suma de una interferencia CWT y ruido blanco gaussiano AWGN. La amplitud y la frecuencia normalizada ( $f/\pi$ ) del tono CWT son 2.2 y 0.025 respectivamente, siendo estos valores constantes durante el periodo que comprende el tamaño del vector de entrada. El AWGN se ha elegido con media 0 y desviación estándar 10.

Los filtros utilizados en la descomposición son los filtros ortogonales de Daubechies con 4 coeficientes. Estos filtros, cuyas características se pueden comprobar en [12], dividen el espectro dependiendo del nivel de resolución de forma similar a los mostrados en la figura 2.7.

El algoritmo comienza en el nivel  $J = 6$  y la señal se descompone hasta alcanzar el nivel  $j = 1$ . El árbol de descomposición producido en el análisis se muestra en la figura 4.11. En esta descomposición el vector  $X_5$  se obtiene al aplicar  $g_{0,6}(n)$  sobre  $X_6$ . Las señales  $X_4$ ,  $X_3$ ,  $X_2$  y  $X_1$  también se obtienen mediante el correspondiente filtro paso-alto  $g_{0,5}(n)$ ,  $g_{0,4}(n)$ ,  $g_{0,3}(n)$  y  $g_{0,2}(n)$  de su nivel. Sin embargo, la señal a la salida del último nivel  $X_0$  se corresponde con la subbanda obtenida a partir de  $h_{0,1}(n)$  y  $X_1$ . El vector  $X_0$  engloba la interferencia en el nivel de menor resolución temporal, y sus coeficientes se igualan a cero al comenzar el proceso de síntesis.

En la gráfica 4.9 se observa el espectro de los vectores  $X_j$  con  $j \in [6, \dots, 1]$ , resultantes en el proceso de análisis. Estas mismas señales se pueden observar representadas en el dominio temporal en la gráfica 4.10

Como ya se ha comentado, la aplicación invertida de la transformada permite que la complejidad del algoritmo sea reducida, ya que el ancho de banda que presentan los filtros en este primer nivel es pequeño. El número de desplazamientos aplicados para centrar la interferencia en la zona pasa-banda de uno de los filtros se reduce. Una vez centrada la interferencia en este primer nivel  $J = 6$ , en los demás niveles sólo es necesario un desplazamiento para centrar la interferencia. En la gráfica 4.9, se puede ver el resultado de la elección realizada en cada nivel  $j$ . Además se comprueba el desplazamiento de la interferencia en la zona pasa-banda de uno de los filtros.

Por otro lado, las señales representadas en el dominio temporal 4.10, muestran los cambios de frecuencia por las que se va pasando durante la aplicación del algoritmo.

Figura 4.9: *Representación en frecuencia de las subbandas desarrolladas y los filtros de Daubechies 4 aplicados en cada nivel. Las señales representadas son:  $X_6$ ,  $X_5 = W_{6,1,D(6)}$ ,  $X_4 = W_{5,1,D(5)}$ ,  $X_3 = W_{4,1,D(4)}$ ,  $X_2 = W_{3,1,D(3)}$ ,  $X_1 = W_{2,1,D(2)}$ ,  $X_0 = W_{1,0,D(1)}$ .*

También se demuestra el alineado en la última subbanda, que concentra la señal con frecuencias cercanas a 0.

## 4.5 Conclusiones

En este capítulo se ha propuesto un nuevo receptor FH-SS, que integra un filtro de supresión de interferencias de banda estrecha. La aplicación del filtro reduce la energía, que las interferencias inyectan en los correladores del receptor. Para eliminar esta energía, el algoritmo distingue entre la interferencia y la señal FH-SS. Los cambios en la frecuencia de la portadora permiten que las salidas de los correladores estén formadas, por una señal binaria y un término proveniente de la interferencia superpuesto.

Capítulo 4. Receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias mediante Transformadas Wavelet Sobremuestreadas

---

Figura 4.10:

Figura 4.11: *Árbol de la transformación resultante. Las subbandas elegidas en cada iteración son:  $X_5 = W_{6,1,D(6)}$ ,  $X_4 = W_{5,1,D(5)}$ ,  $X_3 = W_{4,1,D(4)}$ ,  $X_2 = W_{3,1,D(3)}$ ,  $X_1 = W_{2,1,D(2)}$ ,  $X_0 = W_{1,0,D(1)}$ .*

para comprobar su funcionamiento. Las gráficas aportadas muestran las subbandas, en las que el algoritmo descompone la señal, y el alineado del espectro de la interferencia, con las zonas paso-banda de los filtros en cada nivel de resolución.

Las prestaciones del receptor y el algoritmo propuestos se determinan en el capítulo 6, mediante la Tasa de Bits Erróneos (*Bit Error Rate, BER*) recibidos, en la transmisión simulada a través de canales con diversos tipos y niveles de ruido e interferencias.

# Capítulo 5

## Implementación del algoritmo de supresión de interferencias mediante estructuras *Lattice* sobre Procesadores Digitales de Señal

### 5.1 Introducción

El algoritmo de supresión de interferencias para modulaciones FH-SS, propuesto en el capítulo 4, se basa en el cálculo de la UWPT. Para implementar este algoritmo de forma eficiente, es necesario considerar estructuras de filtrado capaces de realizar dicha transformada con un bajo coste computacional. Estas estructuras deben sintetizarse sobre dispositivos hardware, con los que obtener una alta velocidad de procesamiento. El objetivo de este capítulo es la implementación eficiente del módulo de supresión de interferencias del receptor FH-SS.

La estructura de filtrado utilizada para implementar el algoritmo puede basarse en la forma directa, la estructura *Lattice* o la matriz Polifásica, mostradas en el sección 2.7. En el capítulo 4 se han propuesto las ecuaciones que implementan el algoritmo de supresión de interferencias. Dichas ecuaciones están fundamentadas en la forma directa, es decir, en la convolución de la señal de entrada con los correspondientes filtros paso-alto y paso-bajo. En cada nivel de resolución, los filtros aplicados se modifican insertando un determinado número de ceros entre sus coeficientes, lo que varía su respuesta frecuencial tal como se expone en [89].

Si se atiende al dispositivo sobre el que se realiza la transformación, se pueden encontrar diversos ejemplos de implementación de la transformada, mediante arquitecturas de Muy Alto Nivel de Integración (*Very Large Scale Integration, VLSI*),

realizadas sobre diversos dispositivos de hardware programable: Circuitos Integrados de Aplicación Específica (*Application Specific Integrated Circuits, ASIC*), Campos de Matrices de Puertas Programables (*Field Programmable Gate Array, FPGA*), Dispositivos de Lógica Programable Reprogramables (*Erasable Programmable Logic Device, EPLD*) o mediante Procesadores Digitales de Señal (*Digital Signal Processors, DSP*). Al considerar las arquitecturas VLSI que implementan la Transformada Wavelet, hay que introducir primero aquellas que reproducen directamente la forma directa. Todas ellas, para aumentar la eficiencia del algoritmo, tienen en común el intercalado de las convoluciones de varias subbandas. Este intercalado es posible porque la señal introducida de forma serie se diezma antes de su procesamiento. Dentro de este grupo, la primera arquitectura propuesta [34] utiliza dos unidades aritméticas, que realizan la convolución directamente sobre el flujo serie de entrada de datos. Los resultados para cada subbanda se almacenan en registros de desplazamiento, con un tamaño similar al de los filtros utilizados. En cada impulso de entrada de datos, debido al diezmado, la convolución se realiza sobre una de las subbandas almacenadas en uno de los registros de desplazamiento. El encaminamiento de los datos y de los resultados, desde las memorias hasta las unidades aritméticas, se realiza mediante multiplexores. Otras aproximaciones difieren en la forma de almacenar y encaminar los resultados intermedios, que se pueden realizar mediante redes de encaminamiento sistólicas [5], [88], memoria distribuida [22], memoria RAM [5] o registros intermedios [53].

Mientras que las referencias anteriores ofrecen soluciones óptimas a nivel de la arquitectura implementada, otras aportaciones buscan mayores prestaciones modificando el esquema para el cálculo de la transformada. En [17] [74] [73] se cambia el algoritmo directo utilizado anteriormente, por estructuras *Lattice* y *cordic* fundamentadas en desplazamientos, que reducen el número de operaciones y aumentan la velocidad del procesamiento.

También hay que considerar la realización de la Transformada Wavelet en Procesadores Digitales de Señal, tal como se muestra en [75], [19]. En ambas referencias se utiliza la estructura en forma directa, y se aprovecha el hardware específico de los DSP para realizar la transformada. La ventaja que presentan los DSP frente a los microprocesadores convencionales, es su arquitectura orientada a la realización de operaciones multiplicación-acumulación, donde una sólo instrucción realiza en paralelo dos accesos de datos, una multiplicación y una suma. Además incluyen potentes circuitos aritméticos, que pueden trabajar con operandos en representación en punto flotante, y unidades de generación de direcciones, que permiten realizar múltiples modos de direccionamiento e implementar fácilmente *buffers* circulares.

El objetivo final de este capítulo es la consecución de una latencia pequeña, en el módulo de supresión de interferencias del receptor FH-SS modificado. Para alcanzar dicho objetivo, primero es necesario desarrollar una estructura que calcule la UWPT de forma eficiente. La sección 5.2 introduce las estructuras *Lattice*, estudiando su posible aplicación a la UWT y al cálculo del algoritmo de supresión. Posteriormente

se compara dicha estructura con la implementación en forma directa, y se determina el rendimiento que se obtiene con cada una de las dos alternativas.

No se ha incluido en esta comparación la implementación Polifásica, debido a que la Transformada Wavelet Sobremuestreada no diezma la señal y no se puede anular ninguna fase. En estas condiciones, el rendimiento obtenido es del mismo orden que con la estructura directa, ya que al fin y al cabo la transformación se basa también en convoluciones.

Una vez decida la estructura con la que realizar el algoritmo, la sección 5.4 trata su implementación en un dispositivo del tipo DSP, mostrando la metodología seguida en el desarrollo del software y los módulos en los que se ha dividido el código.

## 5.2 Estructura *Lattice* para Transformadas Wavelet Sobremuestreadas

En el capítulo 2 se ha introducido la estructura *Lattice* para la implementación eficiente de la Transformada Wavelet con filtros ortogonales. Es deseable poder aplicar esta estructura también en el caso de las UWT. Como ya se ha comentado, la diferencia más significativa entre los dos tipos de transformadas es el diezmado. Esta diferencia impide que la UWT pueda utilizar exactamente la misma estructura *Lattice* que la WT.

Al realizar una transformación con UWT y un árbol de descomposición diádico, la estructura *Lattice* del primer nivel con un filtro de orden 4 es la mostrada en la figura 5.1 [85]. Este esquema es similar al de la figura 2.11, con la salvedad de que la señal no se diezma después del filtrado. De esta manera, en el primer nivel de descomposición ambas estructuras coinciden, salvo en el diezmado de la señal de salida.

Matemáticamente la estructura presentada en la figura 5.1 anterior se puede expresar mediante la matriz  $E(z^2)$ , dada en la sección 2.7.2. A partir de ella se define una nueva matriz:

$$T(z) = A \cdot E(z^2) \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \quad (5.1)$$

Al desarrollar la ecuación anterior, se obtiene la relación entre los coeficientes del filtro y los coeficientes de la estructura *Lattice* equivalente. En este desarrollo se supone que los filtros de los que se ha partido están compuestos por cuatro coeficientes  $H_0 = (a, b, c, d)$ .

Figura 5.1: Estructura *Lattice* para un banco de filtros sobremuestreado con 4 coeficientes. En la figura  $X$  es el vector de entrada,  $A, K_1$  y  $K_2$  son los coeficientes de la estructura y  $P_1, Q_1, P_2, Q_2$  son las salidas de las dos secciones que componen la estructura. Las salidas de la estructura,  $P_2$  y  $Q_2$ , son equivalentes a  $H_0$  y  $G_0$ .

$$T(z) = A \cdot \begin{bmatrix} 1 - K_1(z^{-1}) - K_1K_2(z^{-2}) - K_2(z^{-3}) \\ K_2 - K_1K_2(z^{-1}) + K_1(z^{-2}) + z^{-3} \end{bmatrix} \quad (5.2)$$

Comparando la ecuación anterior con la implementación mediante convoluciones, los coeficientes *Lattice* en función de los coeficientes del filtro  $H_0$  son:  $A = a, K_1 = -b/a$  y  $K_2 = -d/a$ . La ecuación 5.2 también demuestra que los coeficientes  $g_0(n)$  y  $h_0(n)$  cumplen la relación:

$$g_0(n) = (-1)^{N-1-n} h_0(N-1-n) \quad (5.3)$$

siendo  $N$  la longitud de los filtros, como se corresponde con filtros obtenidos con la inversión alterna de sus coeficientes [78].

Figura 5.2: *Estructura Lattice de la Transformada Wavelet Sobremuestreada diádica, donde en cada nivel de iteración  $j$ , la matriz de la transformación con estructura Lattice  $T(z)$  se modifica según  $T(z^{2^{j-1}})$ .*

$$T^{-1}(z) = A \cdot \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} R(z^2) \quad (5.4)$$

En el nivel de reconstrucción  $j = 1$ , la matriz  $T^{-1}(z)$  está definida en la figura 5.3. En niveles de reconstrucción superiores se opera de forma similar al caso de análisis, aplicando en cada nivel de resolución  $j \in \{J, \dots, 1\}$  la matriz  $T^{-1}(z^{2^{j-1}})$ , inversa de  $T(z^{2^{j-1}})$ .

Figura 5.3: *Estructura Lattice de la Transformada Wavelet Sobremuestreada inversa con 4 coeficientes. En la figura las entradas  $H'_0$  y  $G'_0$  se corresponden con los filtros de síntesis del banco de filtros,  $A, K_1$  y  $K_2$  representan los coeficientes de la estructura y  $\hat{X}$  es la señal reconstruida.*

### 5.2.1 Comparación en coste computacional de las estructuras en forma directa y *Lattice*

En esta sección se compara el coste computacional de las implementaciones de la UWT en forma directa y con estructura *Lattice*. Como se vio en la sección 2.7, la mayor ventaja de la estructura *Lattice* frente a la implementación en forma directa es la reducción del coste computacional. Aunque las conclusiones alcanzadas en dicha sección, no se pueden aplicar directamente a la transformada UWT, ya que esta última tiene un coste superior, de orden  $(N \log_2(N))$  frente a  $(N)$  de la WT.

En este apartado se cuantifica la diferencia en el coste computacional de ambas estructuras, atendiendo al número de multiplicaciones, sumas, operaciones de lectura/escritura y tamaño de la memoria necesaria.

#### 1. Multiplicaciones.

El número de multiplicaciones que se requieren para implementar el banco de análisis, utilizando filtros con  $L$  coeficientes y estructura *Lattice*, es  $L + 1$  por punto del vector de entrada. El valor  $L + 1$  se refiere a las dos multiplicaciones por etapa para calcular las dos subbandas, más la multiplicación correspondiente al primer coeficiente,  $A$  en la figura 5.1. Este valor se obtiene porque el número de etapas de la estructura con coeficientes distintos de cero es  $L/2$ . El número de multiplicaciones es el mismo en todos los niveles de resolución, ya que al cambiar de nivel la matriz  $T(z^{2^{j-1}})$  sólo varía los retrasos introducidos, y permanece constante el número de puntos del vector de entrada.

En el caso de implementación mediante forma directa, el número de multiplicaciones por punto del vector de entrada, para un banco de filtros de análisis con  $L$  coeficientes, es  $2L$ . Este valor se consigue al multiplicar por 2 el número de coeficientes, puesto que se calculan las dos subbandas. Al igual que para la estructura *Lattice*, el número de multiplicaciones no varía en los distintos niveles de resolución, ya que la UWT no diezma la señal.

#### 2. Sumas.

El número de sumas por punto del vector de entrada para la estructura *Lattice* es dos veces el número de etapas. Si los filtros tienen  $L$  coeficientes, existen  $L/2$  etapas en la estructura *Lattice* con coeficientes distintos de cero. El resultado son  $L$  sumas por punto del vector de entrada. Igual que en el caso de las multiplicaciones, el número de sumas no disminuye con el nivel de resolución.

Para la forma directa el coste en número de sumas, con filtros de  $L$  coeficientes, es  $L - 1$  por subbanda. En total este número se eleva hasta  $2L - 2$  sumas por punto de vector de entrada.

#### 3. Accesos a memoria.

Se supone que todos los términos implicados en la estructura *Lattice* están almacenados en memoria, incluyendo los provenientes de los elementos de retraso, los coeficientes  $A, K_1, \dots, K_{L/2}$  y el valor correspondiente del vector de entrada  $x(n)$ . La salida está formada por los valores pertenecientes a las dos subbandas la paso-alto y a la paso-bajo, que se almacenan también en memoria. El número total de accesos a memoria por punto del vector de entrada se muestra en la tabla 5.1.

Lecturas de los coeficientes $A, K_1, \dots, K_{L/2}$	$L/2 + 1$
Lecturas/escrituras de los elementos de retraso	$2(L/2)$
Lecturas para $x(n)$	1
Escrituras de las subbandas	2
Total	$3L/2 + 4$

Tabla 5.1: *Número de accesos a memoria para la implementación del proceso de análisis de la UWT con estructura Lattice, por punto del vector de entrada.*

Al comparar con la forma directa, cuyos accesos se detallan en la tabla 5.2, se aprecia que el número de accesos en la estructura *Lattice* es inferior.

Lecturas de los coeficientes pertenecientes a $g_0(n)$ y $h_0(n)$	$2L$
lecturas de los puntos del vector $(x(n), \dots, x(n - (L - 1)))$	$L$
Escrituras de las subbandas	2
Total	$3L + 2$

Tabla 5.2: *Número de accesos a memoria para la implementación del proceso de análisis de la UWT con estructura directa, por punto del vector de entrada.*

#### 4. Memoria.

Suponiendo que cada dato ocupa una palabra de memoria, en la implementación *Lattice* el número de palabras ocupadas comprende los *buffers* de retraso,  $2^j \cdot (L/2 - 1) + (2^{j-1})$ , siendo  $j$  el nivel de resolución, más los coeficientes  $L/2 + 1$ . Si se incluyen los vectores de entrada y salida, suponiendo que cada uno de ellos tiene un tamaño  $N$ , el número total de palabras para almacenar un nivel de resolución es  $3N + 2^j \cdot (L/2 - 1) + (2^j - 1) + L/2 + 1$ , donde  $N$  es el número de puntos del vector de entrada.

En la forma directa la cantidad de memoria requerida es menor, porque sólo es necesario almacenar los coeficientes de los filtros, los elementos del vector de entrada y los de salida. En total  $3N + 2L$  posiciones de memoria.

La tabla 5.3 resume el coste computacional en términos de accesos a memoria, número de multiplicaciones, número de sumas y memoria requerida para las dos es-

### 5.3. Estructura *Lattice* para la implementación del algoritmo de supresión de interferencias propuesto

---

estructuras consideradas. En esta tabla se indica el coste del computo de un punto del vector de entrada, en un nivel de resolución cualquiera del proceso de análisis. El coste de la descomposición total, suponiendo que el vector de entrada está constituido por  $N$  puntos, se calcula multiplicando el coste de un punto por  $N \log_2(N)$ .

Como resultado de este apartado se puede afirmar, que la estructura *Lattice* permite un implementación computacional mucho más eficiente. Esta afirmación se basa en los datos recogidos en la tabla 5.3, que muestran que el número de sumas y multiplicaciones es prácticamente la mitad en el caso *Lattice*. Más ventajoso es el número de accesos a memoria, cuestión muy importante puesto que la lectura y escritura de los datos, al realizarse de forma masiva, puede convertirse en un cuello de botella que limite el rendimiento de todo el sistema. El único apartado, donde las características de ambas estructuras se igualan, es el número de posiciones de memoria requeridas.

	<i>Lattice</i>	forma directa
accesos/punto	$3L/2 + 4$	$3L + 2$
multiplicaciones/punto	$L + 1$	$2L$
sumas/punto	$L$	$2L - 2$
memoria para un nivel	$3N + 2^j \cdot (L/2 - 1) + (2^j - 1) + L/2 + 1$	$3N + 2L$

Tabla 5.3: *Tabla de comparación de la implementación de la UWT con la estructura Lattice y la estructura directa, por punto del vector de entrada, siendo  $L$  el número de coeficientes de los filtros y  $N$  el número de elementos del vector de entrada.*

## 5.3 Estructura *Lattice* para la implementación del algoritmo de supresión de interferencias propuesto

Las conclusiones sobre la comparación del coste computacional entre la forma directa y la estructura *Lattice*, mostradas en el apartado anterior, aconsejan la utilización de esta última estructura, en la construcción del algoritmo de supresión de interferencias propuesto. Pero hay que introducir algunas modificaciones significativas en la implementación de la UWT, vista en la figura 5.3. La primera modificación invierte el orden de los niveles de la descomposición, tal como se describe en el capítulo 4. La segunda modificación adapta el árbol de descomposición de la UWT al caso UWPT, donde la subbanda que contiene la mayor cantidad de energía de la interferencia es la escogida en la siguiente iteración.

El algoritmo comienza en el nivel de resolución frecuencial más alto, que corresponde con el nivel  $j = J$ . En este nivel la matriz  $T(z^{2^{J-1}})$  representa el banco de

análisis. Una vez determinada la subbanda con la que continuar la descomposición, después de considerar todos los desplazamientos en frecuencia posibles aplicados sobre el vector original, se continúa su descomposición en el siguiente nivel  $j = J - 1$ , con la matriz  $T(z^{2^{J-2}})$ . De forma iterativa el algoritmo llega hasta el nivel de resolución  $j = 1$ , donde la matriz utilizada es  $T(z)$ .

En cada uno de los niveles hay que calcular la energía de las subbandas obtenidas. Este proceso se realiza en paralelo con la descomposición, ya que los valores obtenidos pueden elevarse al cuadrado y acumularse durante el filtrado. La figura 5.4 muestra un ejemplo de aplicación de la estructura descrita, donde el proceso de análisis supone que el nivel inicial es  $J = 4$  y las subbandas elegidas en cada uno de los niveles son:  $W_{4,1,D(4)}$ ,  $W_{3,0,D(3)}$ ,  $W_{2,1,D(2)}$  y  $W_{1,0,D(1)}$ .

Figura 5.4: *Ejemplo de aplicación del proceso de análisis en el algoritmo de supresión de interferencias con  $J=4$ , mediante estructuras *Lattice*. Donde la aplicación del algoritmo ha determinado que las subbandas elegidas en cada nivel sean:  $W_{4,1,D(4)}$ ,  $W_{3,0,D(3)}$ ,  $W_{2,1,D(2)}$  y  $W_{1,0,D(1)}$  y los desplazamientos para centrar la interferencia sean:  $D(4)$ ,  $D(3)$ ,  $D(2)$  y  $D(1)$ .*

En el proceso de síntesis la matriz que se utiliza para reconstruir la señal es  $T^{-1}(z^{2^{j-1}})$ , definida anteriormente. También es necesario cancelar el desfase y desplazar la señal en frecuencia hasta su posición original, anulando el desplazamiento introducido en el análisis.

La figura 5.5 muestra el proceso síntesis del algoritmo de supresión. En el ejemplo considerado, el algoritmo comienza la reconstrucción a partir de las dos subbandas  $W_{1,0,D(1)}$ ,  $W_{1,1,D(1)}$ , una de las cuales tendrá todos sus coeficientes a cero por ser la subbanda que contiene la interferencia.

Figura 5.5: *Ejemplo de aplicación del proceso de síntesis en el algoritmo de supresión de interferencias con  $J = 4$ , mediante estructuras Lattice. En el proceso de síntesis se combinan las subbandas obtenidas en el análisis y se aplican los desplazamientos:  $-D(4)$ ,  $-D(3)$ ,  $-D(2)$  y  $-D(1)$ , para reconstruir la señal.*

## 5.4 Diseño del algoritmo de supresión de interferencias en Procesador Digital de Señal

En esta sección se modela el algoritmo de supresión de interferencias sobre un microprocesador DSP. La elección de este tipo de microprocesador frente a otros tipos de elementos hardware (FPGA, EPLD, ASIC) se debe a que los DSP permiten un diseño rápido y fácilmente modificable, estando orientados a aplicaciones que requieren una realización masiva de operaciones, y permiten una ejecución rápida de los algoritmos utilizados en el procesamiento de la señal, como los filtros FIR, o la FFT. Este alto rendimiento se consigue mediante la inclusión de hardware específico, con multiplicadores rápidos, unidades aritméticas en punto flotante y unidades de generación de direcciones. Todas estas unidades pueden trabajar en paralelo, siendo los caminos de datos específicos e irregulares. Los dispositivos DSP pueden cargar dos operandos y realizar una multiplicación y una suma, en un sólo periodo de instrucción [25] [39].

En concreto el dispositivo sobre el que se va a trabajar en esta sección es el *ADSP2065L SHARC*. Este DSP de 32-bits está orientado hacia aplicaciones de comunicaciones, audio e instrumentación industrial. El dispositivo como la mayoría de procesadores DSP, presenta arquitectura Harvard y es capaz de realizar operaciones sobre datos en representación en punto flotante. En el anexo B se muestra la estructura interna del dispositivo, que está compuesta por dos buses PM y DM. Ambos permiten el acceso a la memoria, aunque el primero de ellos está indicado para la búsqueda de instrucciones y el segundo de ellos se utiliza para la captura de los datos. La memoria cache integrada en el DSP permite que el bus PM quede liberado en el acceso a las instrucciones, y pueda ser usado para acceder también a datos. En este caso, se consigue el máximo rendimiento del dispositivo, ya que se accede a la instrucción y a dos datos en un sólo ciclo. Esta característica permite la programación eficiente de filtros

digitales, mediante operaciones multiplicación-acumulación.

A la hora de desarrollar un proyecto sobre cualquier microprocesador, es aconsejable la utilización de lenguajes de programación de alto nivel, que reduzcan la complejidad del diseño. Pero el problema aparece a la hora de compilar dichos programas, ya que los compiladores actuales generan un código muy pobre [42] [56]. Existen técnicas de optimización de compiladores para Computadores con Repertorio de Instrucciones Reducido (*Reduced Instruction Set Computer, RISC*), pero los caminos de datos irregulares y el pequeño número de registros internos de los DSP, hacen que la aplicación directa de estas técnicas de optimización de código estén lejos de ser eficientes para los DSP [42]. Por esta razón en el desarrollo del código del algoritmo de supresión, se propone la utilización de un lenguaje de alto nivel, para codificar las subrutinas de carácter general, y lenguaje ensamblador, a la hora de programar las partes críticas como la subrutina que realiza filtrado mediante la estructura *Lattice*.

El contenido de esta sección comienza con la especificación de las ecuaciones, que implementan la estructura *Lattice* para la UWPT, siguiendo el esquema propuesto en el apartado 5.3. Estas ecuaciones se traducen a código ejecutable por el DSP *ADSP2065L SHARC*, atendiendo a sus particularidades y buscando optimizar su rendimiento. Los apartados finales describen la implementación del resto de módulos y la estimación del coste computacional del algoritmo.

### 5.4.1 Implementación de la estructura *Lattice* en Procesador Digital de Señal

En este apartado se especifican las ecuaciones correspondientes a las secciones *Lattice* que calculan la UWPT. Esta estructura, mostrada en la sección 5.2 anterior, necesita para su implementación un número de *buffers* circulares, igual al número de elementos de retraso que aparecen en la estructura, que es a la mitad de coeficientes que tienen los filtros Wavelet. El desarrollo mostrado en este apartado se centra en una estructura con dos secciones, proveniente de un banco de filtros con 4 coeficientes.

Cada uno de estos *buffers* circulares está formado por un número de posiciones de memoria, igual al retraso introducido por el elemento que implementa. Por ejemplo, en el primer nivel de descomposición de la UWPT los elementos de retraso son:  $z^{-1}$  y  $z^{-2}$ , como se observa en la figura 5.1. El primero se compone de un *buffer* con una única posición de memoria y el segundo contiene un *buffer* con dos posiciones. En el segundo nivel, los elementos de retraso son:  $z^{-2}$  y  $z^{-4}$ . En este caso los *buffers* correspondientes contienen 2 y 4 posiciones de memoria respectivamente. En los siguientes niveles, se incrementa el número de posiciones de forma correspondiente a los retrasos introducidos en la matriz  $T(z)$ .

Las ecuaciones para realizar el proceso de análisis sobre el vector de entrada son

las siguientes:

$$\begin{aligned}
 w_0(i) &= z_2(m)K_2 + K_1z_1(n) + x(i)A & (5.5) \\
 w_1(i) &= z_2(m) - K_2(K_1z_1(n) + x(i)A) \\
 m &= (m + 1) \bmod(2^j) \\
 Z_2(m) &= Z_1(n) - K_1xA \\
 n &= (n + 1) \bmod(2^{j-1}) \\
 Z_1(n) &= xA
 \end{aligned}$$

donde  $Z_1$  y  $Z_2$  representan los *buffers* de retraso, y los índices  $n$  y  $m$  sirven para acceder a los elementos de cada uno de los *buffers*. Estos índices tienen sus valores restringidos al tamaño del buffer, por lo que es necesario realizar la operación módulo después de incrementarlos.  $A$ ,  $K_1$  y  $K_2$  representan los coeficientes de la estructura,  $x$  es el elemento del vector de entrada y  $w_0$ ,  $w_1$  las subbandas obtenidas.

Las ecuaciones correspondientes al proceso de síntesis para cada punto del vector de entrada son:

$$\begin{aligned}
 \hat{x}(i) &= A(z_1(n) + w_1(i)K_2 + w_0(i) + K_1z_2) & (5.11) \\
 n &= (n + 1) \bmod(2^{j-1}) \\
 z_1(n) &= z_2(m) - K_1(w_0(i) + K_2w_1(i)) \\
 m &= (m + 1) \bmod(2^j) \\
 z_2(m) &= -K_2w_0 + w_1
 \end{aligned}$$

cada uno de los términos anteriores tiene el mismo significado que en las ecuaciones de análisis, y el vector  $\hat{x}(i)$  representa la señal reconstruida.

### Subrutina para el cálculo de la estructura *Lattice* sobre el DSP *ADSP2065L*

Siguiendo la metodología descrita en el primer apartado de la sección, se va a desarrollar la subrutina que implementa la estructura *Lattice* directamente en lenguaje ensamblador, ya que es la parte más crítica dentro del algoritmo de supresión de interferencias. Este lenguaje permite optimizar el código hasta niveles a los que no puede llegar el compilador de C, y se consigue todo el rendimiento que es capaz de proporcionar el dispositivo DSP elegido.

El código realiza el filtrado en un nivel de resolución  $J$ , correspondiente a la transformada con filtros ortonormales de orden 4  $H_0$  y  $G_0$ . Un ejemplo de este tipo de banco de filtros son los filtros de Daubechies ortonormales con 4 coeficientes.

## Capítulo 5. Implementación del algoritmo de supresión de interferencias mediante estructuras *Lattice* sobre Procesadores Digitales de Señal

---

Los datos con los que se trabaja son del tipo punto flotante de 32 bits. En el *ADSP2065L* este formato sigue la norma IEEE 745/854, con un rango de valores representables:  $[1.2^{-38}, \dots,$

principal de la subrutina, para que el código resultante sea mucho más compacto. Este bucle se especifica en la figura 5.6.

En el código propuesto las escrituras sobre  $z_1$ ,  $z_2$ ,  $h_0$  y  $g_0$  se suponen con post-autoincremento, al igual que las lecturas del vector  $x$ . Los retrasos introducidos por los *buffers*  $z_1$  y  $z_2$  se realizan mediante el direccionamiento post-autoincrementado circular, para que al acceder al último punto del *buffer* la unidad de generación de direcciones automáticamente realice la operación módulo sobre el incremento de la dirección. A partir del pseudocódigo presentado en 5.6, la generación del código ensamblador del algoritmo es automática.

Como conclusión se puede afirmar, que el número de ciclos necesarios para realizar el filtrado del vector de entrada es 6 veces el número de puntos, que contiene dicho vector.

La forma de obtener el código para la subrutina de síntesis es similar, pero en este caso sobre las ecuaciones 5.11. El resultado es un algoritmo, que implementa el proceso de síntesis en 7 instrucciones por punto del vector de entrada. Las dos subrutinas se adjuntan en el anexo D.

### Efecto del tamaño finito del vector de entrada

Al dividir en bloques de tamaño finito el vector de entrada, la transformada se realiza sobre un número finito de puntos, siendo necesario solucionar el problema que aparece al tratar los extremos. Para la implementación de la Transformada Wavelet en forma directa, existen en la bibliografía varias propuestas de cómo operar con los bordes siendo las más comunes [78]:

- Rellenar con ceros: El vector de entrada se extiende mediante puntos con valor cero.
- Extensión simétrica: Implementa el vector de entrada como si fuera un *buffer* circular, cuando se llega a un extremo el algoritmo comienza a tomar los valores del otro extremo.
- Extensión antisimétrica: Es similar a la anterior pero hay un cambio de signo en los valores del vector de entrada al atravesar el límite del vector.

En el caso de la implementación *Lattice* también puede realizarse una extensión simétrica o antisimétrica. Para llevar a cabo la extensión, se comienza la aplicación del filtrado sobre el vector de entrada, considerado también como un *buffer* circular, con un retraso igual al impuesto por la suma de todos los retrasos que contiene la estructura. De esta forma, en vez de comenzar el algoritmo sobre el primer punto

Figura 5.6: *Bucle del algoritmo de filtrado Lattice para el DSP ADSP2065L. En la figura la columna MULT representa el multiplicador del DSP, la columna SUMA el sumador, las columnas DM y PM los buses de acceso a memoria, la columna derecha los registro internos ocupados (en color claro) y los desocupados (con color oscuro).*

del vector, lo hace sobre el punto  $N - R$ , donde  $N$  es la longitud total del vector de entrada y  $R$  es el retraso total impuesto por la estructura *Lattice*.

La salida obtenida durante el filtrado de estos  $R$  primeros puntos se descarta, ya que lo que se persigue en realidad no es el valor de la salida, sino que los *buffers* circulares que implementan el retraso se llenen con los valores adecuados. El primer valor válido se obtiene al llegar el filtrado hasta el primer elemento del vector. Para completar el procesado, todos los valores que componen el *buffer* se leen y se introducen en la estructura. Habrá puntos sobre los que se opere dos veces, la primera vez para realizar la extensión simétrica y la segunda para completar el filtrado del vector. En total el número de puntos sobre los que se realiza el filtrado son  $N + R$ , siendo como antes  $N$  la longitud del vector y  $R$  el retraso total de la estructura. Al introducir la extensión

circular se produce una penalización en el rendimiento del algoritmo, porque hay que filtrar algunos puntos por duplicado.

En la estructura en forma directa no se produce esta penalización computacional al realizar la extensión circular, porque los puntos sobre los que es necesario operar siguen siendo los mismos. Lo que varía es la estructura del vector de entrada, que pasa a ser un *buffer* circular.

### 5.4.2 Implementación del resto de subrutinas que componen el algoritmo de supresión

Una vez que se ha desarrollado la subrutina que calcula la UWPT, el resto de módulos del algoritmo de análisis, tal y como se propone en el pseudocódigo 4.6, construidos en lenguaje C son:

- La subrutina que desplaza en frecuencia la señal mediante la operación:

$$x_{j,d} = x_j(n) \cdot e^{(i2\pi nd)/N} \quad (5.16)$$

donde  $x_j$  es el vector en el nivel  $j$ , con componentes real e imaginaria provenientes de los dos correladores en cuadratura. Este vector se desplaza con el factor  $d$  y  $N$  es el número de elementos en el vector. El término  $e^{(i2\pi nd)/N}$  puede calcularse en tiempo de ejecución o puede almacenarse en memoria una oscilación para cada una de las frecuencias. Esta oscilación sirve para construir la señal  $e^{(i2\pi nd)/N}$ , que se utiliza en el desplazamiento del vector.

- Subrutina para el cálculo de la energía de los vectores paso-bajo y paso-alto. Aunque se puede implementar una subrutina para esta tarea, resulta más rentable computacionalmente realizar este cálculo en la subrutina del filtrado *Lattice*, modificando el *bucle* para que después de calcular el valor de salida para ambas subbandas, obtenga las energías de ambos valores y acumule su diferencia como resultado.
- También es necesaria una subrutina para la selección de la subbanda con mayor energía, que además copie esta subbanda sobre el vector  $x_{j-1}$ , para inicializar la nueva iteración. La subbanda descartada y el desplazamiento óptimo se almacenan de cara a la reconstrucción.

El proceso de síntesis utiliza las mismas subrutinas, aunque en un orden diferente y con desplazamientos inversos.

### 5.4.3 Análisis del coste computacional del algoritmo de supresión

Para calcular el coste computacional de la estructura de análisis que implementa el algoritmo, en función del número de puntos de entrada  $N$ , hay que considerar dos factores:

1. Coste medido en ciclos por punto del vector de entrada.

Es el coste de filtrar un punto del vector de entrada, medido en número de ciclos necesarios para realizar todas las operaciones 5.5, que calculan las subbandas aplicando la estructura *Lattice*. Este valor en la programación realizada sobre el DSP es 6 ciclos de reloj. El número de ciclos para filtrar el vector de entrada es  $6N$ .

2. Coste en número de iteraciones.

El número de iteraciones en el filtrado del vector de entrada es:

$$T(N) = (J - 1) + \lfloor \frac{N/(2^J)}{\Delta d} \rfloor + 1 \quad (5.17)$$

El primer factor se corresponde con los  $(J - 1)$  niveles, en los que sólo se realiza una única iteración de filtrado sobre la señal. El segundo factor se corresponde con el nivel inicial, donde se realizan varias iteraciones con desplazamientos en frecuencia distintos. El número de desplazamientos viene dado por el término  $\lfloor \frac{N/(2^J)}{\Delta d} \rfloor + 1$ , que representa la división entera entre el ancho de los filtros y el paso de resolución en frecuencia, más el incremento en una unidad correspondiente al desplazamiento con valor 0.

Combinando ambos valores, el número de ciclos de reloj para calcular todos los vectores de la descomposición de análisis es:

$$T(N) = 6N[J + \lfloor \frac{N/(2^J)}{\Delta d} \rfloor] \quad (5.18)$$

Para el proceso de síntesis, ya que sólo es necesaria una iteración de filtrado por nivel, el coste es:

$$T(N) = 7 \cdot N \cdot J \quad (5.19)$$

siendo 7 es el coste en ciclos por punto y  $J$  el coste en número de iteraciones.

## 5.5. Conclusiones

---

También hay que tener en cuenta el efecto producido por la extensión circular de la transformada, que incrementa el número de puntos sobre los que se aplica el filtrado hasta  $N + R$ , siendo  $R$  el retraso introducido por la estructura *Lattice*. Este retraso depende del nivel y del número de elementos de retardo incluidos en la estructura. Con esta modificación la ecuación del coste en el análisis es:

$$T(N) = 6(N + \overline{R_a})[J + \lfloor \frac{N/(2^J)}{\Delta d} \rfloor] \quad (5.20)$$

donde  $\overline{R_a}$  denota el valor medio del retraso a través del proceso de análisis. En la síntesis la ecuación es:

$$T(N) = 7(N + \overline{R_s})J \quad (5.21)$$

donde  $\overline{R_s}$  es el valor medio del retraso en el proceso de síntesis.

A la vista de la ecuación 5.20, cuando se aproxima  $J \rightarrow \log_2(N)$  y  $\Delta d \rightarrow 1$ , el coste tiende a  $T(N) \rightarrow (6N) \cdot \log_2(N)$ , cercano al límite teórico  $O(N \log_2(N))$  que impone la transformada.

## 5.5 Conclusiones

La Transformada Wavelet Sobremuestreada presenta un coste computacional de orden  $O(N \log_2(N))$ , debido a que en cada nivel de resolución la señal obtenida no se diezma. Este coste computacional es superior al de la Transformada Wavelet convencional, que es de orden  $O(N)$ . Por esta razón es muy importante reducir el número de operaciones por punto del vector de entrada, en el filtrado que realiza el algoritmo de supresión de interferencias en cada nivel. Dos estructuras que permiten esta reducción son la descomposición Polifásica y las estructuras *Lattice*, aunque al no realizarse diezmo, la estructura Polifásica no puede aplicarse en este caso. La comparación entre las dos formas consideradas, directa y *Lattice*, se ha saldado a favor de ésta última, porque reduce el número de operaciones por punto del vector de entrada en un factor mitad.

La implementación del algoritmo de supresión de interferencias se ha llevado a cabo sobre un dispositivo DSP. Este tipo de dispositivo permite un desarrollo relativamente rápido de los algoritmos de procesamiento de señal. Aunque la codificación de los algoritmos en lenguajes de alto nivel, no permite extraer todo el rendimiento que pueden aportar estos procesadores. La subrutina que aplica las secciones *Lattice* en ensamblador mantiene ocupados, prácticamente en todos los ciclos del bucle principal, la mayoría de las unidades funcionales del dispositivo. Todas las operaciones que realiza la estructura sobre cada punto del vector de entrada en el proceso de análisis,

se han programado en 6 ciclos de instrucción. Este número coincide con el número de multiplicaciones total que se necesitan para calcular las ecuaciones 5.5. La subrutina de síntesis, mostrada en el anexo D, está formada por 7 instrucciones.

Para concluir el capítulo, se ha calculado el coste del algoritmo utilizando filtros de cuatro coeficientes. El coste se ha estimado mediante el computo del número de instrucciones, obtenido a partir de las subrutinas de análisis y síntesis programadas en ensamblador para en el DSP *ADSP2065L*.

## 5.5. Conclusiones

---

# Parte IV

## Resultados



---

La parte de resultados está dividida en dos capítulos. En el primero se recogen las pruebas, que mediante simulación y prototipo en laboratorio, han validado el algoritmo propuesto. El segundo contiene la aplicación del algoritmo a una red de control industrial a través del cable de potencia.

De forma expresa los capítulos son:

- Evaluación de las prestaciones del receptor para modulaciones de Espectro En-sanchado con Saltos de Frecuencia con supresión de interferencias
- Aplicación del algoritmo de supresión propuesto a la transmisión de datos por el cable de potencia

---

# Capítulo 6

## Evaluación de las prestaciones del receptor para modulaciones de Espectro Ensanchado con Saltos de Frecuencia con supresión de interferencias

### 6.1 Introducción

Este capítulo contiene los resultados de la evaluación del rendimiento del receptor FH-SS, con el módulo de supresión de interferencias propuesto en el capítulo 4. Para comprobar el comportamiento del algoritmo y el receptor, se ha optado por utilizar la simulación y la pruebas sobre prototipo.

La simulación juega un papel importante en el análisis, diseño e implementación de sistemas de comunicación y procesamiento de la señal, siendo de utilidad para verificar la funcionalidad del sistema, evaluar el rendimiento y generar las especificaciones que guían el diseño. Hoy en día la simulación se utiliza en el desarrollo de multitud de sistemas, desde los complejos y costosos sistemas de comunicación vía satélite hasta los productos de consumo de bajo coste [70].

Dentro del término simulación se engloban diferentes análisis, que en una primera aproximación, se pueden clasificar como deterministas y estocásticos. En el primer tipo de simulaciones no hay elementos aleatorios o incontrolables, en este caso las entradas y las salidas se consideran exactas porque no hay ninguna incertidumbre asociada a sus valores. En el segundo tipo de simulaciones alguna de las entradas son cantidades aleatorias, cuyos valores exactos en una evaluación individual del sistema no

son conocidos de antemano [33].

En las simulaciones estocásticas, que son las que se van a utilizar en el presente capítulo, la entrada está compuesta por variables aleatorias con unas determinadas distribuciones. Por lo tanto, es necesario utilizar generadores de números aleatorios, que proporcionen la distribución considerada. Contrariamente al caso determinista, dos ejecuciones de la simulación bajo condiciones iguales pueden producir resultados diferentes, si los valores de entrada son independientes. Los valores de salida de una simulación estocástica forman a su vez una variable aleatoria con una distribución desconocida, ya que si no fuera así no sería necesaria la simulación, siendo el objetivo final la estimación de alguno de sus valores estadísticos: la media, la varianza, etc [33].

Dentro de las simulaciones estocásticas, también es posible establecer una clasificación que las divide en estáticas y dinámicas. Las estáticas, al contrario que las dinámicas, no incluyen ninguna dependencia temporal y normalmente la bibliografía las referencia como simulaciones de *Monte Carlo* [30]. Estas simulaciones son relativamente sencillas de realizar, y en muchos casos se aplican en problemas de cálculo matemático que no es tratable analíticamente.

Para llevar a cabo una simulación estocástica estática, el modelo se ejecuta repetidamente bajo las mismas condiciones y parámetros, usando una secuencia de valores de entrada independiente. Las salidas constituyen variables aleatorias independientes e idénticamente distribuidas, a las que se les puede aplicar técnicas estadísticas. La única cuestión que resta por determinar, y que condiciona la fiabilidad de la salida, es el número de repeticiones. El grado de fiabilidad de los resultados se mide mediante el intervalo y el nivel de confianza. El intervalo de confianza especifica el rango de valores en los que está comprendida la salida, con una probabilidad determinada por el nivel de confianza [55]. Estos dos términos se estudian con mayor detalle en el apartado 6.2.2 del capítulo.

Para validar el algoritmo de supresión también se ha construido un prototipo. Sobre él se ha programado el algoritmo usando las estructuras *Lattice*, tal como se detalla en el capítulo 5. El dispositivo usado en esta implementación ha sido el DSP *ADSP2065L*, para guardar la compatibilidad con las ecuaciones de coste calculadas en la sección 5.4.3.

El capítulo se ha dividido en dos secciones. La primera sección 6.2 contiene la especificación del simulador construido y las pruebas realizadas sobre él. Para cada uno de los casos, caracterizados por los parámetros de la modulación empleada y por el tipo de interferencia, se adjuntan y se comentan los resultados obtenidos.

La segunda sección 6.3 expone el prototipo desarrollado, que incluye la implementación del algoritmo de supresión mediante estructura *Lattice*, tal como se propone en el apartado 5.3. Este prototipo se ha diseñado sobre el DSP *ADSP2065L*, lo que ha permitido realizar medidas del coste computacional en número de ciclos de ejecución.

## 6.2 Resultados mediante simulación

El simulador desarrollado está compuesto por un emisor y un receptor FH-SS, modelados sobre el paquete de software matemático Matlab. En el primer apartado 6.2.1 de la sección, se muestra el emisor y el receptor construidos, con el fin de aclarar las condiciones exactas en las que se han llevado a cabo las simulaciones. También se especifica dentro de esta sección, la forma en la que se ha calculado el intervalo y el nivel de confianza de los resultados obtenidos.

Para comprobar la validez del algoritmo se simula la transmisión de una secuencia de bits, a través de un canal sin distorsión y afectado por los siguientes tipos de ruido e interferencias:

- Ruido AWGN: modelado mediante la función *normrnd* de Matlab.
- Interferencia CWT: modelada como un término senoidal continuo con frecuencia aleatoria, pero fija durante la duración de un bloque de entrada al filtro de supresión.
- Interferencia de banda estrecha gaussiana: modelada como AWGN y filtrada para que ocupe sólo una parte del espectro de frecuencias que utiliza un canal de la expansión FH.

Las condiciones generales de la modulación se han mantenido constantes a lo largo de todas las simulaciones. Dichas condiciones básicamente son coincidentes con las consideradas en el capítulo 4, en el diseño del receptor FH-SS con supresión de interferencias. Estas condiciones son:

- Se supone que la modulación empleada es FH-SS combinada con BFSK y un único *chip* por bit transmitido.
- Los canales utilizados por la expansión son ortonormales y se eligen de forma aleatoria, según la secuencia PN. La construcción de este código se explica en el próximo apartado. El número de canales es 5, salvo en el experimento 4.
- Cada uno de estos canales está separado  $2\text{Khz}$ , mientras que las señales BFSK tienen una separación en frecuencias de  $1\text{Khz}$ .
- El periodo de un bit es 1 milisegundo para que los canales sean ortonormales.

Aparte de las condiciones de la modulación, hay que determinar las características del algoritmo de filtrado.

## 6.2. Resultados mediante simulación

---

En primer lugar hay que subrayar, que los filtros usados en todas las simulaciones son los de Daubechies con 4 coeficientes, introducidos en el apartado 2.4.2. También resulta necesario especificar los parámetros del algoritmo definidos en el capítulo 4, donde se incluye: el incremento del desplazamiento en frecuencia  $\Delta d$ , el nivel inicial en la descomposición  $J$  y el número de puntos que compone el vector de entrada  $N$ . En las 5 primeras simulaciones llevadas a efecto, los tres parámetros han permanecido fijos a los siguientes valores:

- El vector de entrada tiene  $2^{15}$  puntos.
- El nivel inicial es  $J = 9$ . Este valor se ha escogido porque es el que mejor resultados produce. En el experimento 6.2.8 se muestran los resultados obtenidos para otros niveles. En este primer nivel el ancho de las zonas paso-banda de las subbandas obtenidas es:  $\frac{2^{15}}{2^9} = 64$  puntos.
- El incremento del desplazamiento es  $\Delta d = 5$ . El número de iteraciones que debe realizar el algoritmo en el nivel  $J = 9$  es 13, puesto que el ancho de banda de las subbandas dividido por  $\Delta d = 5$  es:  $\frac{64}{5} \simeq 12$  y hay que añadir la iteración correspondiente al desplazamiento 0. El número total de iteraciones para el proceso de análisis es  $13 + 8$ , las 13 antes definidas para el primer nivel, más una por cada uno de los restantes niveles. En el proceso de síntesis el número de iteraciones de filtrado es 9.

Las dos últimas simulaciones son las únicas en las que se han modificado los valores anteriores, con el fin de comprobar la dependencia entre el rendimiento del receptor y los valores de estos parámetros.

### 6.2.1 Estructura del simulador

En esta sección se incluye la descripción del simulador construido mediante Matlab, representado en la figura 6.1. Al modelar el simulador sobre computador, las señales están generadas a partir de una determinada frecuencia de muestreo, que se ha escogido bastante elevada para evitar problemas de aliasing. El intervalo de frecuencia utilizado por la expansión FH-SS está en el rango  $(5 - 16)Khz$ , siendo la frecuencia de muestreo elegida  $50Khz$ , salvo en el experimento 4 donde el número de canales se multiplica por 10 y la frecuencia escogida es  $500Khz$ .

Las especificaciones de los módulos que integran el simulador son:

- Emisor:

Figura 6.1: *Diagrama de bloques del simulador construido sobre Matlab. En la figura se representan los módulos y las funciones empleadas en su construcción. El primer módulo es el emisor compuesto del generador de bits aleatorio, el generador de secuencia PN y modulador. El segundo módulo es el canal, que incorpora tres tipos de interferencias. El tercer módulo es el receptor que contiene el demodulador, el filtro paso-bajo, el algoritmo de supresión y el sumador no-coherente. El último módulo es el detector de error.*

- El generador de bits está desarrollado a partir de la función *rand* de Matlab, detallada en [1]. Esta función crea un array de números aleatorios uniformemente distribuidos entre  $[0, 1]$ . Los valores se aproximan al entero más cercano para producir la secuencia de bits, siendo el número de elementos en el array el mismo que el número de bits transmitidos.
  - El generador de la secuencia PN utiliza la misma función *rand* para determinar el canal utilizado en la transmisión. Los valores del array ahora se generan en el intervalo  $[0, m/2 - 1]$ , donde  $m/2$  es el número de canales utilizados. El número de elementos en el array es el mismo que en el caso anterior.
  - El modulador genera un símbolo, por medio de la función *cos*, con una frecuencia determinada por los elementos de los dos vectores anteriores.
- Canal:
    - La señal a la salida del emisor se puede combinar con tres tipos de ruido.

Los modelos AWGN y el gaussiano de banda estrecha parten los dos de la misma función *normrnd*, que crea un vector de elementos de distribución gaussiana, con media y desviación estándar definidos como parámetros de la función. La interferencia gaussiana de banda estrecha se filtra mediante convolución, a partir de la función *conv* y el correspondiente filtro definido con la función *fir1*. La interferencia CWT se construye mediante uno o varios términos cosenoidales, obtenidos a partir de la función *cos* con una frecuencia determinada y fase aleatoria, definida con *rand* en el intervalo  $[0, 2\pi[$ .

- Receptor:
  - La recepción comienza con la multiplicación de la señal recibida con todos los correladores en cuadratura.
  - La salida de los correladores se filtra de forma paso-bajo con la función *conv* y un filtro definido a partir de *fir1*.
  - El siguiente módulo es el filtro de supresión. Este módulo está programado en lenguaje C y se incluye como un objeto de librería. El código de esta función se muestra en el anexo C.
  - El último módulo del receptor suma los puntos pertenecientes a un símbolo, comparando los dos valores que se corresponden a los dos bits de la modulación BFSK, desplazados hasta el canal dado por la secuencia PN.
- Detector de error:
  - Para detectar los errores en los bits transmitidos, existe un módulo que compara los bits del array inicial y los generados en el receptor.

El número de iteraciones que se han llevado a cabo en las simulaciones está en el orden de los  $600 \cdot 10^3$  bits. Un ejemplo del código en Matlab que constituye el simulador se puede ver en el anexo C.

### 6.2.2 Cálculo de los intervalos y niveles de confianza

El objetivo de las simulaciones es la estimación de la tasa de bits erróneos a la salida del receptor. Esta estimación se calcula realizando la media de las variables de salida  $y_i$  obtenidas para cada bit enviado, siendo  $i$  el índice que indica el orden en la transmisión de los bits. Estas variables contienen un valor 1, para los bits recibidos erróneamente, y un 0 para los que se reciben de forma correcta. A partir de la secuencia de valores observados, se puede obtener una estimación de la media definida como:  $\bar{Y}(n)$ , que será una aproximación de la media real del proceso:  $\mu_y$ . El valor de  $\bar{Y}(n)$  representa la tasa de error en la transmisión de los bits y se define:

$$\bar{Y}(n) = \sum_{i=1}^n \frac{y_i}{n} \quad (6.1)$$

donde  $n$  es el número total de bits transmitidos. El desarrollo de esta ecuación y de todo el apartado, se puede consultar en [30] y [54].

Debido a que la media  $\bar{Y}(n)$  es una variable aleatoria, que depende de la secuencia de valores observados, la precisión con la que se estima la media real desconocida  $\mu_y$  puede ser válida con una probabilidad:

$$P(|\bar{Y}(n) - \mu_y| < \Delta_y) = 1 - \nu \quad (6.2)$$

donde  $\Delta_y$  es la mitad del intervalo de confianza de la estimación de la media realizada en 6.1, y  $1 - \nu$  es el nivel de confianza, con  $0 < \nu < 1$ . De esta manera, al calcular el intervalo  $2\Delta_y$  para un nivel de confianza  $1 - \nu$  se puede afirmar que si el experimento se repite  $n$  veces, el valor  $\bar{Y}(n)$  estará comprendido en  $(\mu_y - \Delta_y, \mu_y + \Delta_y)$  en un  $100(1 - \nu)\%$  de los casos.

El cálculo de  $\Delta_y$  se puede realizar a partir de la expresión:

$$\Delta_y = t_{n-1, 1-\nu/2} \hat{\sigma}[\bar{Y}(n)] \quad (6.3)$$

donde  $t$  representa la distribución de las variables de salida y  $t_{n-1, 1-\nu/2}$  es el cuantil de dicha distribución. El valor  $\hat{\sigma}[\bar{Y}(n)]$  es un estimador de la varianza de  $\bar{Y}(n)$  expresado como:

$$\hat{\sigma}^2[\bar{Y}(n)] = \sum_{i=1}^n \frac{|y_i - \bar{Y}(n)|^2}{n(n-1)} \quad (6.4)$$

Para valores de  $n > 30$ , la distribución  $t$  se puede sustituir por una distribución normal, como consecuencia del teorema del límite central. En este caso,  $t_{n-1, 1-\nu/2}$  se puede sustituir por:  $z_{1-\nu/2}$ , que representa el  $1 - \nu/2$  cuantil de la distribución normal [54]. Los valores de  $z_{1-\nu/2}$  están tabulados en [30].

Si las variables  $y_i$  no constituyen una secuencia independiente e idénticamente distribuida, no se puede estimar la varianza mediante 6.4 y es necesario considerar métodos más complejos [30].

### 6.2.3 Experimento 1: variación de la frecuencia de la interferencia

En este primer caso se supone que solamente uno de los cinco canales está afectado por interferencia de banda estrecha, modelada como CWT. El objetivo de este experimento es comprobar el funcionamiento del algoritmo, para distintas posiciones en frecuencia del tono interferente dentro de un canal de la expansión FH, comparando sus resultados con los obtenidos por un receptor sin filtro de supresión y otro receptor con un algoritmo basado en FFT.

*Figura 6.2: Gráfica de resultados experimento 1: variación de la frecuencia de la interferencia CWT. La gráfica representa 3 tipos de receptores: sin filtro, con filtro mediante FFT y con filtro mediante UWPT. La interferencia varía su posición en frecuencias, para situarse en las posiciones que van desde la de una de las portadoras, hasta el punto central del canal de la expansión FH. La potencia de la interferencia CWT es 14dB mayor que la de la señal FH-SS y la SNR del AWGN es de 12dB.*

Para cumplir el objetivo marcado en este experimento, se supone que el emisor para realizar la expansión FH-SS de un símbolo generado por la BFSK, puede elegir entre los 5 canales ortonormales con la misma probabilidad. La interferencia de banda estrecha se modela como un tono CWT, que afecta solamente a uno de los 5 canales, y se considera estacionaria sobre el tamaño del bloque utilizado por la transformada. La potencia de la interferencia se ha supuesto 14dB mayor que la de la señal FH-SS, siendo la relación SNR con el AWGN de 12dB.

En la figura 6.2 se muestra la probabilidad de error en función de la posición en frecuencia ocupada por la interferencia. En dicha figura, la frecuencia de la interferencia

viene dada en función de su desplazamiento, con respecto a la frecuencia de una de las portadoras. Las posiciones en las que se sitúa la interferencia varían desde la ocupada por la portadora (0), hasta el punto central ( $F_c$ ) del canal, siendo la resolución dentro de este rango de valores  $F_c/50$ , con  $F_c = 500Hz$ . La figura 6.2 compara los resultados obtenidos al utilizar la FFT y la UWPT en el algoritmo de supresión. El algoritmo con FFT elimina todos los puntos en el dominio transformado que superen un cierto umbral, tal como se indica en [51].

La probabilidad de error se estima con la media de las variables aleatorias de salida  $y_i$ , que indican si se ha producido error en la transmisión del bit  $i$ . El intervalo de confianza se calcula a partir de las ecuaciones 6.3 y 6.4, suponiendo que la secuencia de valores es independiente e idénticamente distribuida. Un ejemplo de la aplicación de las fórmulas a los datos de la gráfica 6.2 es:

El valor de la probabilidad de error para FFT, con desplazamiento de la frecuencia de la interferencia  $120Hz$ , es  $8.638 \cdot 10^{-4}$ , obtenido mediante la división del número de errores (514) por el número de bits simulados ( $595 \cdot 10^3$ ). En estas condiciones la varianza estimada utilizando la ecuación 6.4 es  $1.450 \cdot 10^{-9}$  y el intervalo para un nivel de confianza del 99% es  $8.935 \cdot 10^{-5}$ . Así pues, el intervalo de confianza presenta un valor con un orden de magnitud inferior a la probabilidad de error. Para todos los demás puntos que componen la gráfica con FFT, este intervalo se mantiene en el mismo orden. En la simulación con supresión mediante UWPT, el intervalo es ligeramente menor, aproximadamente  $6 \cdot 10^{-5}$ , y los resultados sin filtro de supresión están dentro de un intervalo de confianza de 0.002.

Los valores de los intervalos de confianza se mantienen prácticamente constantes en todos los demás experimentos, siendo el intervalo de confianza siempre un orden de magnitud inferior respecto a los datos representados, con un nivel de confianza del 99%.

A partir de la gráfica 6.2, se desprende que el algoritmo mejora sustancialmente la probabilidad de error que presenta el sistema sin filtro de supresión. La diferencia es de dos décadas aproximadamente, manteniéndose constante para todas los desplazamientos, salvo para interferencias con desplazamiento cercano a cero. En este último caso se produce una fuerte degradación del sistema, porque la interferencia está justo sobre una de las portadoras, representado el caso peor. Este resultado coincide con los datos mostrados en [7].

La comparación entre FFT y UWPT refleja que el algoritmo propuesto funciona mejor para la mayoría de frecuencias. Pero la diferencia entre las dos transformadas se diluye al acercarse la interferencia a la posición que ocupa una de las portadoras, porque ambas transformadas distorsionan fuertemente la señal de comunicación. Para el resto de frecuencias la UWPT presenta una probabilidad de error inferior, ya que es capaz de concentrar mejor la energía de la interferencia.

### 6.2.4 Experimento 2: variación del ruido aditivo blanco gaussiano

El objetivo del segundo experimento es comprobar la dependencia del algoritmo con las variaciones del AWGN. También se incluye interferencia CWT situada sobre una de las portadoras de un único canal. En los resultados se compara el receptor sin filtro de supresión, el receptor con un algoritmo basado en FFT o en UWPT y el receptor sin filtro de supresión sobre un canal sin interferencia CWT, que representa el límite mínimo de la probabilidad de error.

Figura 6.3: *Gráfica de resultados experimento 2: variación del ruido AWGN. En la gráfica  $E_b/J_0$ (dB) es el SNR medido en decibelios y se representan 4 tipos de situaciones: receptor sin filtro, receptor con filtro mediante FFT, receptor con filtro mediante UWPT y el receptor sin filtro y sin interferencia CWT. La potencia de la interferencia CWT es 14dB mayor que la potencia de la señal FH-SS.*

Este segundo experimento se ha realizado bajo las mismas condiciones que el apartado anterior. Aunque en este caso se ha fijado la frecuencia de la interferencia para que se sitúe sobre una de las portadoras, de uno de los 5 canales utilizados por la modulación FH-SS. La potencia de la interferencia se ha supuesto 14dB mayor que la de la señal FH-SS. El valor que varía en esta ocasión es la Densidad Espectral de Potencia (*Spectral Power Density, SPD*) del AWGN, definida como  $J_0$ , y el SNR  $E_b/J_0$ . En la gráfica 6.3, se puede comprobar que al variar el nivel del ruido, la probabilidad de error decae de forma exponencial siguiendo la ecuación 3.4.

Cuando hay interferencia de banda estrecha, el receptor sin filtro de supresión

obtiene una probabilidad de error, que decae exponencialmente sólo para niveles muy altos de ruido gaussiano. Al descender el nivel de AWGN, predominan los errores debidos a la interferencia y la probabilidad de error no descende. El valor final tiende hacia el valor obtenido en la gráfica anterior 6.2.

En el caso del receptor con filtro de supresión, aunque la caída de la probabilidad de error no es exponencial, se aproxima mucho a esta función. Para niveles muy bajos de ruido gaussiano, el valor de la probabilidad de error tiende al valor obtenido en la gráfica del experimento 1, con un desplazamiento del tono CWT cercano a cero.

También es interesante resaltar la probabilidad de error conseguida por el receptor con supresión mediante FFT. Para dicho receptor, la probabilidad es mayor que para UWPT, coincidiendo con los datos del experimento 1, y se puede afirmar que el receptor con UWPT supera al receptor con FFT en ambas situaciones.

### 6.2.5 Experimento 3: interferencia multitono

En este apartado se estudia el caso de una interferencia multitono, que sitúa cada uno de sus componentes frecuenciales en uno de los canales de la modulación FH. El objetivo de este experimento es comprobar el rendimiento del receptor, frente a variaciones de la potencia de la interferencia que afecta a todos los canales. En este caso sólo se contempla el receptor con UWPT y el receptor sin filtro de supresión.

La interferencia multitono, como se introdujo en el apartado 3.3.3, degrada fuertemente el sistema, porque los tonos continuos son una forma efectiva de introducir energía en los detectores no-coherentes [7]. La potencia total se divide en  $Q$  componentes de igual potencia con fases aleatorias.

$$j(t) = \sum_{l=1}^Q \sqrt{\frac{2R}{Q}} \cos(\omega_l t + \theta_l) \quad (6.5)$$

donde  $R$  es la potencia total de la interferencia,  $\omega_l$  son las frecuencias ocupadas y  $\theta_l$  es la fase. La relación de la potencia de la señal  $S$  y la potencia de la interferencia es:  $R/Q = S/\alpha$ , donde  $\alpha$  es la relación entre la potencia de la portadora y la interferencia que afecta a un canal, como se definió en la sección 3.3.3.

Los tonos se distribuyen por todo el ancho de banda ocupado por la señal FH-SS. Para situar cada una de los componentes de la interferencia en una frecuencia determinada, se pueden utilizar distintas estrategias. Siempre se va a suponer que los tonos se sitúan sobre frecuencias ocupadas por las portadoras de la FH-SS. Estas condiciones han sido elegidas porque simplifican el análisis y producen resultados más pesimistas.

## 6.2. Resultados mediante simulación

---

En la simulación hay un tono por cada uno de los 5 canales en los cuales se puede expandir la señal, que se superpone a una de las dos portadoras que forman la BFSK. El valor  $\alpha$  varía en el rango  $[0, \dots, 25]$  y la relación SNR con el AWGN es de  $12dB$ .

*Figura 6.4: Gráfica de resultados experimento 3: variación de la potencia de la interferencia multitono. La gráfica representa 2 tipos de receptores: sin filtro y con filtro mediante UWPT. El SNR con respecto al AWGN es  $12dB$ . En la gráfica  $\alpha$  es la relación entre la potencia de la portadora y la interferencia que afecta a un canal.*

La gráfica 6.4 muestra los resultados de esta simulación. La primera conclusión es que el nivel, a partir del cual el algoritmo de interferencias comienza a trabajar, está situado aproximadamente en valores de  $\alpha$  superiores a 10. Es decir, cuando la potencia de los tonos de la interferencia es diez veces menor que la potencia de las portadoras.

La segunda conclusión es que una vez el algoritmo comienza a eliminar la interferencia, aunque aumente su potencia, la probabilidad de error no crece significativamente. La razón de este comportamiento es que la zona del espectro que se elimina siempre es la misma, independientemente de la amplitud de la interferencia y la degradación que sufre la señal FH-SS es la misma.

También hay que resaltar el comportamiento para valores de  $\alpha$  pequeños, que coincide con el que presenta el receptor FH-SS no-coherente sin filtro de supresión.

### 6.2.6 Experimento 4: comparación con FFH para interferencia multitono en el caso peor

El objetivo de este experimento es comparar las prestaciones del receptor con UWPT, frente al receptor FH-SS sin supresión de interferencias y el receptor FFH-SS, bajo interferencia multitono en el caso peor [72].

Como se ha explicado en el apartado 3.3.3, una interferencia multitono distribuye sus componentes entre los canales usados por la expansión FH, pero la forma en la que se reparte la energía de la interferencia puede ser variada. En el caso peor el valor  $\alpha$ , definido anteriormente en 3.3.3, se mantiene constantemente a 1. Bajo estas condiciones, lo que se modifica al variar la potencia de la interferencia es el número de canales afectados, pero en cada uno de ellos la interferencia es capaz de producir errores porque  $\alpha = 1$ .

Definiendo  $J_i = R/W_{ss}$ , donde  $J_i$  es la SPD,  $W_{ss}$  es el ancho de banda ocupado por la modulación y  $R$  es la potencia de la interferencia, la relación entre la energía de los bits  $E_b$  y la SPD  $J_i$  es:

$$\frac{E_b}{J_i} = \frac{m/2\alpha}{QK} \quad (6.6)$$

donde  $m/2$  es el número de canales ocupados por la expansión FH,  $K$  es número de bits codificados en cada símbolo de la transmisión FSK y  $Q$  es el número de componentes de la interferencia multitono.

En la simulación se varía el número de canales afectados por la interferencia, pero no la amplitud de los tonos. Se ha simulado la transmisión con modulación BFSK ( $K = 1$ ) y las expansiones FH-SS y FFH-SS, ésta última con tres *chips* por bit transmitido. El receptor FFH-SS combina de forma no-lineal la energía de los correladores utilizando el esquema auto-normalizador [82]. Al igual que en los experimentos anteriores, el SNR con el AWGN es de  $12dB$ .

La probabilidad de error de los tres receptores simulados se representa en la figura 6.5. El receptor FH-SS sin filtro de supresión presenta un comportamiento que decae de forma lineal con  $E_b/J_i$ . Por otro lado, el receptor FFH-SS supera al receptor FH-SS para ciertos intervalos de  $E_b/J_i$ , siendo este resultado concordante con los valores obtenidos por otros autores, como en [83].

Por su parte, el receptor FH-SS con supresión de interferencias supera las prestaciones de ambos receptores, manteniendo los resultados aproximadamente iguales a los obtenidos en el experimento anterior. Para niveles altos de interferencia, tiende a una probabilidad en el rango  $(5 - 6)10^{-2}$ , al igual que en la gráfica 6.4. Para niveles bajos, decae hacia los valores que se obtendrían en el caso de que no hubiera interferencia.

Figura 6.5: *Gráfica de resultados experimento 4: comparación con FFH para interferencia multitono en el caso peor. En la gráfica  $E_b/J_i$ (dB) representa la relación entre la señal de comunicación y la interferencia, medida en decibelios. Se han considerado 3 tipos de receptores: sin filtro, con filtro mediante UWPT y con modulación FFH-SS. El SNR con respecto al AWGN es 12dB.*

### 6.2.7 Experimento 5: ruido gaussiano de banda estrecha

Este experimento centra su atención en la interferencia gaussiana de banda estrecha, y compara los resultados de los receptores FH con y sin filtro de supresión.

Para generar la interferencia, se parte de una señal modelada como ruido gaussiano blanco. Después la señal se filtra para que afecte a una fracción de cada uno de los canales de la expansión FH-SS. La fracción de cada canal ocupada por la interferencia se denomina  $\mu$  y constituye el parámetro variable de la simulación.

El rango de frecuencias dentro de un canal, que se ve afectado por la interferencia, se centra sobre una de las portadoras. El ancho de banda ocupado por el ruido crece a partir de esta posición central, mientras que la SPD se mantiene constante. Al crecer el ancho de banda  $W_j$  que ocupa y mantenerse la SPD  $J_o$ , crece su potencia  $R = J_o W_j$ . El valor de la SPD  $J_o$  es 6dB superior a  $E_b$  para todo los valores de  $\mu$ .

A partir de la figura 6.6, se concluye que al aumentar el ancho de banda de la interferencia, el algoritmo de supresión deja de ser efectivo. En este caso el error es prácticamente similar al obtenido en el receptor sin filtro. Pero el algoritmo sigue comportándose bien, cuando el ancho de banda de la interferencia se mantiene en

Figura 6.6: *Gráfica de resultados experimento 5: ruido gaussiano de banda estrecha. En la gráfica  $\mu$  representa la fracción del canal FH-SS ocupado por la interferencia. Se comparan 2 tipos de receptor: sin filtro de supresión y con filtro mediante UWPT. El valor de  $J_o$  es 6dB superior a  $E_b$ , para todos los valores de  $\mu$ .*

niveles significativamente inferiores, al ancho de banda asignado a un canal de la expansión FH-SS.

### **6.2.8 Experimento 6: variación del tamaño del bloque y del nivel inicial del algoritmo**

En los apartados anteriores el tamaño del bloque, el incremento del desplazamiento en frecuencia  $\Delta d$  y el número de niveles en la descomposición se han mantenido fijos. En todos los experimentos vistos hasta ahora, el tamaño del bloque es  $2^{15}$ , el incremento 5 y el nivel inicial  $J = 9$ .

Las dos últimas simulaciones, que se van a realizar, estudian la dependencia del algoritmo de supresión UWPT con respecto a estos tres parámetros, que inciden de forma muy importante en el coste computacional del algoritmo. En concreto en este apartado, se consideran variables el tamaño del bloque utilizado en las simulaciones y el nivel inicial de comienzo del algoritmo  $J$ . Tal como se muestra en la figura 6.7, el tamaño del bloque varía entre tres valores 2048, 4096 o 8192 y el nivel inicial está comprendido en el rango  $J \in [6, \dots, 10]$ .

La variación del último parámetro, el incremento del desplazamiento en frecuencia

$\Delta d$ , se estudia en el siguiente apartado, quedando fijado a 1 en este caso.

Los demás términos, dependientes de la interferencia y de la modulación, se han escogido con valores iguales a los utilizados en los apartados anteriores. Concretamente, se ha elegido un receptor FH-SS con filtro de supresión mediante UWPT, con 5 canales de frecuencia para la expansión  $FH$  tomados de forma aleatoria. La interferencia, considerada del tipo CWT, tiene una potencia igual a la de la señal de comunicación y sólo afecta a uno de los cinco canales utilizados. La frecuencia de la interferencia varía aleatoriamente, dentro del rango ocupado por el canal afectado, y el nivel de ruido AWGN presente en el canal tiene un valor SNR de  $12dB$ .

A la vista de la gráfica 6.7, se puede afirmar que cuanto mayor es el tamaño del vector  $N$ , menor es la probabilidad de error que afecta al sistema. Esto es debido a que el aumento de  $N$  produce una mayor resolución frecuencial, permitiendo al algoritmo centrar mejor la interferencia en la zona paso-banda de uno de los filtros.

Otra conclusión, extraída de la gráfica 6.7, es que el nivel de comienzo del algoritmo  $J = 9$  minimiza la probabilidad de error en la transmisión, siendo el óptimo independientemente del tamaño del bloque.

Para confirmar ambas afirmaciones, se incluye la gráfica 6.8, donde aparece el espectro de la interferencia antes y después de aplicar el algoritmo de supresión. Como muestra la gráfica, la interferencia afecta a una zona del espectro estrecha y el algoritmo elimina parte de su energía, independientemente de la frecuencia en la que se sitúa la interferencia.

La gráfica 6.8 permite confrontar la resolución frecuencial y el número de puntos del vector transformado. Al considerar el espectro correspondiente a los dos tamaños del vector de entrada 8192 y 4096, es claro, que el incremento del tamaño tiene una correspondencia con la resolución frecuencial, porque el número de puntos contenido en la gráfica para 8192 es el doble del contenido en 4096, ocupando el mismo intervalo de frecuencias.

En segundo lugar se puede observar la dependencia con el nivel inicial. La zona del espectro eliminada dependiendo del nivel inicial se hace más amplia ( $J = 8$ ) o más estrecha ( $J = 10$ ). Si el nivel de comienzo del algoritmo es  $J = 10$ , la zona del espectro que se elimina no engloba toda la energía de la interferencia, siendo lógico que la probabilidad de error aumente. Si por el contrario el nivel es el inferior  $J = 8$ , se elimina una zona del espectro más ancha de la necesaria y la señal de comunicación FH-SS se ve afectada, aumentando de nuevo la probabilidad de error.

Figura 6.7: *Gráfica de resultados experimento 6: variación del tamaño del bloque y del incremento del desplazamiento en frecuencias. La gráfica representa el receptor con filtro de supresión mediante UWPT, con 5 canales de frecuencia para la expansión FH, con 3 tamaños de bloque de la transformada: 8192, 4096 y 2048. La potencia de la interferencia CWT afecta a un único canal, siendo igual a la señal FH-SS. El SNR es 12dB.*

### 6.2.9 Experimento 7: variación del número de niveles y el incremento del desplazamiento en frecuencia

En este último apartado se comprueba el comportamiento del algoritmo, frente a variaciones del nivel inicial  $J$  y del incremento de desplazamiento  $\Delta d$ . Este incremento se utiliza para modificar los desplazamientos en frecuencia aplicados sobre la señal en el nivel  $J$ , lo que condiciona la precisión conseguida por el algoritmo al centrar la interferencia en la zona paso-banda de uno de los filtros. El tercer parámetro del algoritmo, el tamaño del bloque, se ha fijado a 2048.

Antes de presentar los resultados de la gráfica 6.9, hay que explicar el significado de la variable  $\Delta d'$ . Este parámetro se introduce para poder comparar los desplazamientos en frecuencia, cuando el algoritmo tiene distintos niveles iniciales  $J$ , porque el ancho de banda que presentan las zonas paso-banda de los filtros depende de  $J$ . Este ancho de banda se expresa como:

$$B = N/2^J \tag{6.7}$$

Figura 6.8: *Espectro de la interferencia CWT en la entrada y la salida del algoritmo. En la figura a) se representa la interferencia cuando el bloque de la transformada es 8192, considerando 3 niveles iniciales para el algoritmo UWPT,  $J \in [8, 9, 10]$ . En la figura b) se representa la interferencia cuando el bloque de la transformada es 4096, considerando 3 niveles iniciales para el algoritmo UWPT,  $J \in [8, 9, 10]$*

donde  $N$  es el tamaño del vector de entrada y  $J$  el nivel inicial del algoritmo.

Para poder comparar los resultados obtenidos con distintos niveles iniciales  $J$ , es necesario que el  $\Delta d$  sea distinto y proporcional al ancho de las zonas paso-banda en cada uno de los niveles. Por ejemplo, no representa lo mismo un  $\Delta d = 1$  en el nivel  $J = 6$  que en el nivel  $J = 9$ . En el primer caso este desplazamiento puede ser insignificante, porque el ancho de las zonas paso-banda de los filtros en ese nivel comprende muchos puntos, mientras que para  $J = 9$  puede ser un desplazamiento considerable.

Para evitar el problema, se define  $\Delta d'$  que es un desplazamiento en frecuencia proporcional, dependiente del nivel de resolución inicial  $J$  al que se aplique. Este nuevo parámetro se relaciona con  $\Delta d$  y el nivel  $J$  de la siguiente manera:

$$\Delta d = 0.25 + (N/2^{J+3})(\Delta d' - 1) \quad (6.8)$$

El primer incremento de desplazamiento  $\Delta d$  es 0.25 para cualquier nivel  $J$ . A partir de ahí, cada nivel  $J$  tiene un paso de simulación distinto y los valores de  $\Delta d$  se

expanden más o menos dependiendo del factor  $(N/2^{J+3})$ . Por ejemplo, para  $J = 6$  y  $\Delta d' \in [1, \dots, 8]$  los valores de  $\Delta d$  son:

$$\Delta d = \{0.25, 4.25, 8.25, 12.25, 16.25, 20.25, 24.25, 28.25\}$$

Los resultados mostrados en la gráfica 6.9, permiten confirmar que el nivel inicial  $J = 9$  sigue siendo el óptimo, para cualquier incremento del desplazamiento  $\Delta d$ , y que la probabilidad de error aumenta conforme crece  $\Delta d$ . Este resultado se debe claramente a la disminución de la resolución en los desplazamientos en frecuencia, que no permite al algoritmo centrar la interferencia de forma adecuada en la zona paso-banda de uno de los filtros.

Figura 6.9: Gráfica de resultados experimento 7: variación del número de niveles. La gráfica representa el receptor con filtro de supresión mediante UWPT, con 5 canales de frecuencia para la expansión FH y niveles iniciales  $J \in [6, 7, 8, 9, 10]$ . La potencia de la interferencia CWT afecta a un único canal y es igual a la señal FH-SS. El SNR con respecto al AWGN es 12dB.

### 6.3 Resultados sobre prototipo

En esta sección se comprueba experimentalmente los resultados del coste computacional del algoritmo, aportados en la sección 5.4. Para comprobar dichos resultados se ha traducido el simulador generado sobre Matlab, a lenguaje C para su aplicación

sobre el dispositivo DSP *ADSP2065L*, cuyas características se dan en el anexo B. El objetivo es conseguir medir el tiempo de computo real del algoritmo sobre dicho DSP. Este nuevo simulador contiene la subrutina que realiza la transformada en lenguaje ensamblador, mediante estructura *Lattice*, tal y como se ha especificado en el apartado 5.4. Además de comprobar los resultados de coste, estas nuevas simulaciones permiten verificar las subrutinas realizadas en ensamblador, a través de la confrontación con los resultados anteriores.

El programa contiene los mismos módulos de la figura 6.1, salvo el bloque generador de ruido AWGN. En las simulaciones realizadas no se incluye este tipo de ruido, sólo existe interferencia CWT sobre uno de los canales de la expansión FH. Todos los demás bloques son similares a los utilizados anteriormente y descritos en el apartado 6.2.

La modulación sigue siendo BFSK a nivel de bit, con símbolos de periodo  $1ms$ , y expansión FH-SS con 5 canales ortonormales separados por  $2Khz$ . Estas condiciones son las que normalmente se han utilizado en las simulaciones dentro de la sección 6.2 anterior.

Para medir el coste computacional, se hace uso de las características del DSP y se cuenta el número de ciclos de reloj que tarda en ejecutarse el algoritmo de supresión. En estas condiciones se ha repetido el experimento del apartado 6.2.9 anterior, que explora la dependencia del algoritmo con respecto al valor del incremento de desplazamiento  $\Delta d$  y el nivel inicial  $J$ . Seguidamente, se detalla la prueba realizada y los resultados obtenidos.

#### **6.3.1 Simulación con variación del número de niveles y del incremento del desplazamiento en frecuencia, sobre el DSP ADSP2065L**

Las condiciones en las que se desarrolla la simulación son las mismas que las escogidas en el apartado 6.2.9 anterior. En este experimento, el número de niveles y el incremento de desplazamiento  $\Delta d$  se consideran variables, y el tercer parámetro que define el algoritmo, el tamaño del bloque, se ha fijado a 2048.

Al igual que en la figura 6.9, las gráficas 6.10 y 6.11 de este apartado tienen en el eje de abscisas los valores del incremento de desplazamiento normalizado  $\Delta d'$ , para poder comparar los distintos niveles iniciales  $J$ .

Los resultados del coste computacional de la simulación se exponen en la figura 6.10. Esta gráfica, en lugar de contener la probabilidad de error en el eje de ordenadas, presenta el coste del algoritmo, medido en número de ciclos de instrucción consumidos por el DSP para la descomposición de análisis. El cómputo incluye todas las iteraciones realizadas, abarcando el filtrado de cada uno de uno de los desplazamientos en

frecuencia aplicados sobre la señal de entrada, en todos los niveles de resolución del proceso de análisis.

Figura 6.10: *Número de ciclos en la ejecución de la descomposición de análisis para el filtro de supresión mediante UWPT sobre el DSP ADSP2065L. En la gráfica  $J$  es el nivel inicial de la descomposición y  $\Delta d'$  es el desplazamiento normalizado.*

A partir de los datos mostrados en la figura 6.10, la primera cuestión que salta a la vista es la dependencia entre el coste computacional y el valor de la variable  $\Delta d'$ . El coste para  $\Delta d' = 1$  ( $\Delta d = 0.25$ ) siempre es el máximo, porque los valores pequeños de  $\Delta d'$  incrementan el número de iteraciones del algoritmo en el primer nivel. Por el contrario, cuando el valor de  $\Delta d$  se hace comparable al ancho de las zonas paso-banda de los filtros, el coste se estabiliza porque el número de iteraciones permanece constante.

Por ejemplo, para  $J = 6$  los valores de  $\Delta d$  y el número de iteraciones en el nivel inicial  $J = 6$  son:

$$\Delta d = \{0.25, 4.25, 8.25, 12.25, 16.25, 20.25, 24.25, 28.25\}$$

$$\text{número de iteraciones en el nivel inicial} = \{128, 8, 4, 3, 2, 2, 2, 2\}$$

estos valores de  $\Delta d$  y el número de iteraciones se ven reflejados en la gráfica 6.10, confirmando el hecho de que a partir de  $\Delta d' = 5$ , para  $J = 6$ , el coste se estabiliza. Esta conclusión se puede aplicar a todos los demás valores de  $J$ .

### 6.3. Resultados sobre prototipo

---

La segunda cuestión importante es la dependencia entre el nivel inicial y el coste computacional. El nivel inicial  $J = 6$  es el que exhibe un coste computacional inferior, salvo para el punto  $\Delta d' = 1$  ( $\Delta d = 0.25$ ), que es demasiado pequeño en relación al ancho de las zonas paso-banda de los filtros ( $2^{11}/2^6$ ). Este incremento en el coste para  $\Delta d' = 1$  no tiene ningún beneficio en términos de probabilidad de error, porque, tal como se observa en la figura 6.9, la probabilidad de error es prácticamente similar para todos los valores de  $\Delta d'$ .

En definitiva, está claro que el incremento del coste computacional en el nivel  $J = 6$  y  $\Delta d' = 1$  no sirve para nada, porque la zona del espectro eliminada es demasiado grande y el algoritmo degrada fuertemente la señal FH-SS.

Con el nivel inicial  $J = 10$ , se produce la situación contraria. En este caso, el coste computacional para  $\Delta d' = 1$  es el menor para cualquier  $J$ . Pero conforme  $\Delta d'$  crece, el número de iteraciones se estabiliza, y el coste del algoritmo con  $J = 10$  se convierte en el mayor simplemente porque hay más niveles en la descomposición. La desventaja que supone el coste computacional para  $J = 10$ , no se ve recompensada por una mayor fiabilidad, como cabría esperar. Si se examinan las figuras 6.9 y 6.7, se puede constatar que el nivel que proporciona menor probabilidad de error es  $J = 9$ . Por lo tanto, el nivel 10 tampoco es el más recomendable.

Los otros tres niveles intermedios ( $J = [7, 8, 9]$ ) tienen pequeñas diferencias en su coste computacional, menores de la cuarta parte de su valor. De los tres niveles,  $J = 9$  es el que mayor coste tiene, aunque también es el que presenta la menor probabilidad de error. Si se comparan las probabilidades de error para cada uno de ellos, se advierte una diferencia de una década entre  $J = 9$  y  $J = 7$ , y de más de la mitad de una década entre  $J = 9$  y  $J = 8$ .

La conclusión que se desprende de toda esta discusión es que el nivel  $J = 9$  representa el nivel óptimo, bajo las condiciones expuestas anteriormente, porque el descenso de la probabilidad de error compensa el incremento en el coste.

Para terminar esta sección, a partir de los datos de la figura 6.10, se puede comprobar la validez de la ecuación 5.20, que estima el coste del algoritmo medido en número de ciclos de instrucción sobre el *ADSP2065L*. El número de ciclos predichos por la ecuación 5.20, se aproxima bastante al número de ciclos de la gráfica 6.10 obtenidos experimentalmente.

Por ejemplo, para  $\Delta d' = 8$ , con  $J = 9$  y  $N = 2048$ , el valor experimental es: 279620, mientras que el coste calculado por la ecuación 5.20, sustituyendo  $N = 2048$ ,  $J = 9$ ,  $\overline{R}_a = 231$ ,  $\lfloor \frac{N/(2^J)}{\Delta d} \rfloor = 1$  y considerando el filtrado de los dos correladores en cuadratura,  $X_J = X_{Jr} + jX_{Ji}$ , es:

$$(6 * (2048 + 231) * (9 + 1)) * 2 = 273480 \quad (6.12)$$

los resultados de coste no son exactamente iguales, debido a que la función que implementa la estructura *Lattice*, junto con las instrucciones del bucle de filtrado, contiene instrucciones para inicializar las unidades de generación de direcciones y algunos registros internos, no contempladas en los cálculos de la ecuación 6.12, tal como se puede comprobar en el anexo D.

La segunda figura 6.11 mostrada en este apartado, representa la probabilidad de error obtenida en las simulaciones llevadas a cabo sobre el *ADSP2065L*, para los casos  $J = 10$ ,  $J = 9$  y  $J = 8$ . Es posible comparar dichos resultados con los de la figura 6.9 del apartado 6.2.9. Al comparar ambas gráficas, se observa que la probabilidad de error en este caso es ligeramente inferior a los valores aportados en la figura 6.9, porque en este último caso sólo se ha incluido interferencia CWT, aunque su tendencia es coincidente. Este resultado valida las dos implementaciones realizadas, tanto la llevada a cabo sobre el DSP con estructura *Lattice*, como la efectuada en Matlab mediante la forma directa.

Figura 6.11: *Gráfica de resultados con variación del número de niveles, para el simulador sobre DSP y estructura Lattice. La gráfica representa el receptor con filtro de supresión mediante UWPT, con 5 canales de frecuencia para la expansión FH, para 3 niveles iniciales J distintos. La potencia de la interferencia CWT afecta a un único canal y es igual a la señal FH-SS. No se ha introducido AWGN.*

## 6.4 Conclusiones

En este capítulo se ha verificado el comportamiento del algoritmo de supresión de interferencias mediante simulación. Con estas pruebas se ha estimado la probabilidad de error en los bits transmitidos, con diversos tipos y niveles de interferencia. Para llevar a cabo las simulaciones, se ha construido un modelo del emisor y el receptor BFSK con expansión FH-SS, y se ha emulado la transmisión de la señal a través de un canal sin distorsión, con ruido AWGN e interferencia CWT o gaussiana de banda estrecha.

El primer resultado logrado muestra la diferencia en el comportamiento entre el receptor con y sin filtro de interferencias. La reducción de la probabilidad de error, que presenta el receptor con supresión de interferencias, es significativa, más de 2 décadas de diferencia frente al convencional. Esta ventaja demuestra la validez del receptor y el algoritmo de supresión propuesto, para cada uno de los niveles y tipos de interferencias especificadas en las simulaciones.

Las comparaciones efectuadas con el algoritmo basado en la transformada FFT han permitido constatar, que la UWTP tiene mejor comportamiento, porque consigue reducir el número de puntos en los que se concentra la interferencia y degrada en menor medida la señal FH-SS. Aunque para frecuencias cercanas a la de la portadora, las diferencias entre ambas transformadas son pequeñas.

Otro aspecto importante, de los resultados ofrecidos, es la comparación entre el receptor para modulación FFH-SS y el receptor para modulación FH-SS, con algoritmo de supresión. Como se introdujo en el capítulo 3, la modulación FFH-SS con recepción no-lineal aumenta la inmunidad del sistema frente a interferencias de banda estrecha. Sin embargo, la modulación FFH incrementa la complejidad del emisor y el receptor, necesita mayor ancho de banda para mantener la misma velocidad de transmisión y produce un aumento de la probabilidad de error en canales con ruido AWGN, debido a la combinación no-coherente de la energía de los *chips*.

El algoritmo propuesto incrementa la inmunidad del sistema frente al receptor FFH-SS, como han demostrado las simulaciones realizadas para el caso  $L = 3$ , siendo  $L$  el número de *chips* por bit. Esta conclusión es importante porque cumple el deseo de obtener una fiabilidad frente a interferencias elevada, sin tener las desventajas que introduce la combinación no-coherente de los *chips* que constituyen un bit.

Para concluir el capítulo se ha traducido el simulador, de la sección 6.2, a lenguaje C para su aplicación sobre el DSP *ADSP2065L*. La simulación que se ha llevado a cabo ha copiado las condiciones del apartado 6.2.9. Esta similitud ha permitido comparar y validar el algoritmo de supresión con estructura *Lattice*, porque los resultados de la probabilidad de error mostrados en la gráfica 6.11 coinciden con los valores de la gráfica 6.9.

El prototipo también ha ayudado a calcular el número de ciclos de instrucción empleado por las subrutinas. Estos resultados han corroborado los suministrados por la ecuación 5.18, propuesta en el capítulo 5, que estima el coste del algoritmo de análisis en función del nivel inicial  $J$ , el tamaño del bloque  $N$  y el incremento del desplazamiento  $\Delta d$ .

## 6.4. Conclusiones

---

# Capítulo 7

## Aplicación del algoritmo de supresión propuesto a la transmisión de datos por el cable de potencia

### 7.1 Introducción

Este capítulo expone la aplicación del algoritmo de supresión, a un sistema de comunicación por cable de potencia integrado en una arquitectura de control industrial distribuida. Para realizar esta aplicación ha sido necesario desarrollar un emisor y un receptor FH-SS, adaptados para la transmisión de datos a través de dicho canal. Además estos elementos se han integrado dentro de una arquitectura computacional orientada al control industrial. Dicha arquitectura tiene como finalidad la gestión y el control de los elementos agrarios, que constituyen una comunidad de regantes y ha sido llevado a cabo por el grupo de Control y Aplicaciones Telemáticas (*CAT*) del *Institut de Robòtica*, dentro del proyecto de investigación PRIFEM. La información referente a la arquitectura mencionada y los elementos que la integran se muestra en el anexo A. Como parte del desarrollo de este proyecto, se han llevado a cabo pruebas de campo en una instalación piloto, que ha permitido constatar la eficacia del algoritmo de supresión de interferencias propuesto. El sistema piloto se ha instalado en el sector *XIV* de la comunidad de regantes "La Providencia", en el término municipal de Sagunto, según las especificaciones del anexo A.

Dentro del sistema, los elementos emisor y receptor constituyen el nivel de comunicación más bajo, que interconecta las unidades remotas encargadas de controlar sensores y actuadores lejanos, con el cabezal de riego donde se hayan las bombas de extracción de agua y abonado. Para alimentar y llevar la señal de comunicación a

cada una de estas unidades, existe un único cable de potencia que recorre toda la comunidad. La ventaja que presenta la utilización de esta línea, como medio de transmisión de la información, es sobre todo económica, ya que no son necesarios nuevos cables para el envío de los datos, ni son necesarias fuentes externas para alimentar las unidades remotas, que se pueden conectar en cualquier punto de la red con total flexibilidad [57].

Pero el cable de potencia del sistema constituye un canal de transmisión muy hostil. Los mayores problemas que presenta su utilización son: ancho de banda limitado, características variables y altos niveles de ruido. Asimismo, las señales transmitidas se ven afectadas por una gran distorsión causada por las largas distancias de cable, la baja impedancia de entrada y las pérdidas en las conexiones.

Algunos de estos inconvenientes se deben al tipo de cargas que normalmente se conecta a la línea, como por ejemplo: motores, equipos de iluminación, electrodomésticos, etc. Estos equipos modifican las condiciones del canal y afectan a la señal transmitida [67], incorporando en muchos casos cargas inductivas, por lo que su impedancia depende de la frecuencia [62]. Otra característica importante del canal es la variación de la impedancia, en función del periodo de la señal alterna de potencia. Típicamente, cuando dicha señal se encuentra en los pasos por cero, la impedancia es mayor que cuando se sitúa en un intervalo con voltaje elevado [62], debido a que los rectificadores de las fuentes se conectan en los instantes de voltaje alto. Los tipos de interferencias que aparecen más comúnmente en este medio son: impulsional, generado sobre todo en tubos fluorescentes, ruido de onda continua, proveniente de motores y fuentes conmutadas, y ruido de fondo.

El capítulo comienza con el estudio del medio. En concreto, la sección 7.2, indica sus propiedades más destacables, haciendo hincapié en los tipos y características de las interferencias presentes y justificando la aplicación del algoritmo de supresión de interferencias propuesto en el capítulo 4. A continuación, la sección 7.3 se centra en la construcción del emisor y el receptor FH-SS, éste último basado en el mostrado en la sección 4.2. Para cada uno de los dos elementos se especifica exactamente sus propiedades, incluyendo la modulación, la estructura de bloques y la señal que se transmite. Para concluir el capítulo, la sección 7.4 muestra los resultados de la probabilidad de error en la transmisión de datos a través del canal especificado, obtenidos sobre la instalación prototipo.

## 7.2 Análisis del canal de comunicación

La línea de potencia utilizada como canal de comunicación en el sistema propuesto tiene una serie de particularidades, que la convierten en un medio de transmisión bastante adverso. En primer lugar aparece el problema de la gran cantidad de cable

que forma la red. Esta cuestión es ineludible ya que la extensión de terreno sobre la que se distribuyen los elementos remotos, hace necesaria largas distancias de cableado para su interconexión. El recorrido del cable se adapta a la división del terreno y presenta una topología totalmente irregular, característica de cada instalación. A modo de ejemplo, en el anexo A se puede observar el esquema del cableado de la instalación sobre la que se han realizado las pruebas.

La tensión de alimentación presente en la línea, con la que funcionan las unidades remotas, se ha reducido a  $24Vac$  por razones de seguridad. Para reducir el voltaje existe un transformador, que conecta el tramo de red en el que se sitúan las unidades remotas, frente a la fase de  $220Vac$  tomada de la instalación.

La impedancia de entrada a la línea no puede saberse con absoluta certeza y varía en cada punto del cable del sistema, ya que su longitud y su topología no son previsibles. Aun así, las mediciones realizadas indican que esta impedancia es pequeña, con valores entorno a las decenas de ohmio. La atenuación sufrida por la señal también depende de las condiciones del cable. Considerando los valores medidos en la instalación de pruebas, se puede afirmar que la atenuación sufrida por encima de  $100Khz$  es bastante severa, siendo el rango  $(0 - 30)Khz$  la zona del espectro con menor atenuación.

Figura 7.1: *Representación temporal de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento. Esta interferencia es del tipo CWT y está formada por multiples armónicos.*

Como se ha introducido al comienzo del capítulo, las mayores fuentes de ruido inyectado en la línea son los motores, las fuentes conmutadas y las lámparas fluorescentes de iluminación. Los tipos de ruido más normales son: ruido impulsional, ruido

de onda continua CWT y ruido de fondo AWGN [67] [6] [62]. La diferencia más significativa entre el ruido impulsional y la interferencia CWT es su representación en el dominio de la frecuencia y en el dominio temporal. La energía de la interferencia impulsional está expandida en un amplio rango de frecuencias y concentrada en un pequeño intervalo temporal, mientras que contrariamente la energía de la onda continua se concentra en frecuencias y se expande en el tiempo. Esta diferencia aconseja el tratamiento de ambas interferencias en dominios distintos.

Figura 7.2: *Espectro de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento. En la gráfica se observa que los componentes frecuenciales de la interferencia, la mayoría provenientes del variador y la bomba están concentrados sobre todo entre las frecuencias (3–7)KHz. Aunque también hay armónicos cercanos a frecuencia 0Hz provenientes de la señal de potencia del cable.*

De toda la variedad de ruido que puede contener el cable de potencia, la interferencia CWT es la más importante en el sistema contemplado. Dicha interferencia aparece generada por los motores eléctricos inductivos, bombas de agua y abonado, que se conectan a las tomas de potencia y se controlan mediante equipos variadores de frecuencia [91]. Este dispositivo genera una onda con voltaje y frecuencia variable, mediante la que se controla la velocidad del motor. La interferencia creada tiene un potencia elevada y está constituida por una secuencia de armónicos, centrados en frecuencias múltiplos de la frecuencia fundamental de conmutación del variador [91] [76].

Este análisis ha sido confirmado por medio de mediciones realizadas en el cable

Figura 7.3: *Detalle del espectro de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento. En la gráfica se observa los componentes frecuenciales de la interferencia CWT, para frecuencias (3 – 7)KHz, con armónicos situados en posiciones múltiplos de 250Hz.*

de potencia del sistema piloto, con el variador y la bomba en funcionamiento. En concreto la figura 7.1 representa en el dominio temporal la interferencia justo después del filtro de entrada del receptor, que elimina la señal de potencia de  $50Hz$ . A simple vista, esta señal se puede clasificar como una secuencia periódica de amplitud menor de  $0,4V$ .

La representación de esta misma señal en el dominio de la frecuencia se puede ver en las figuras 7.2 y 7.3. La primera de ellas representa todo el espectro de frecuencias. Al inspeccionar dicho espectro se aprecia que existe una zona que concentra los armónicos de mayor energía, esta zona comprende el rango de frecuencias ( $3KHz - 7KHz$ ). También aparecen armónicos de elevada energía en posiciones cercanas a  $50Hz$ , provenientes de la tensión de potencia que transmite el cable.

Para determinar con mayor claridad la composición de la interferencia de onda continua, la segunda gráfica 7.3 representa en detalle la zona del espectro ( $3KHz - 7KHz$ ). En esta gráfica se aclara la distribución de los armónicos generados. En concreto, se puede apreciar su distribución en intervalos fijos, con un desplazamiento de frecuencia de  $250Hz$ .

De forma resumida las propiedades del canal considerado son:

- El canal de comunicación es un cable bifilar de potencia, en el que se comparte

la administración de la alimentación y la transmisión de datos, siendo tarea del emisor y el receptor distinguir entre ambas señales.

- La topología del cable puede ser cualquiera y su longitud elevada, por lo tanto la impedancia de entrada será baja e impredecible a priori. El ancho de banda utilizable está en el rango de frecuencias  $(0 - 30)Khz$ , según las mediciones realizadas en la instalación de pruebas.
- La interferencia dominante es del tipo CWT. Esta interferencia se genera en el equipo variador que controla la bomba eléctrica, estando constituida por una serie de armónicos situados a intervalos fijos de frecuencia, que cubren un amplio rango del espectro.

También existe interferencia de tipo impulsional de menor potencia proveniente de fuentes de iluminación, e interferencia CWT generada por la señal de potencia de  $50Hz$ .

- El cable no está adaptado, porque en los extremos no es posible conectar cargas que harían caer la tensión de alimentación de  $50Hz$ . Debido a esta circunstancia, la señal se ve afectada por reflexiones que distorsionan su forma de onda, es decir, la portadora no se recibe como una onda senoidal perfecta.

Como conclusión de esta sección, se puede afirmar que el cable de potencia de la instalación es un medio ruidoso, siendo la interferencia de mayor potencia del tipo CWT. Estas condiciones son las que se propusieron en el diseño del algoritmo de supresión, por lo tanto este sistema constituye un buen banco de pruebas donde validar el algoritmo de forma experimental.

## 7.3 Construcción del emisor y el receptor con modulación FH-SS

En esta sección se muestra el diseño de las unidades emisor y receptor, que forman la aplicación para la transmisión de datos por el canal de potencia. Pero antes de comenzar la descripción de las dos unidades, se indica la modulación aplicada en este capítulo. Esta modulación está influenciada por las condiciones del canal, mostradas en el apartado 7.2 anterior.

### 7.3.1 Características de la modulación

Con el fin de mantener la compatibilidad con las simulaciones del capítulo 6, la modulación elegida es FH-SS combinada con BFSK. Al comenzar a generar la trama, la

modulación BFSK codifica cada bit 0 ó 1 mediante un símbolo con frecuencia  $6Khz$  ó  $7Khz$  respectivamente. Para mantener la ortogonalidad de los dos símbolos el periodo de cada uno de ellos es  $1ms$ , ya que al enventanarlos con pulsos rectangulares de  $1ms$ , el primer paso por cero de su espectro en forma de *sinc* se sitúa en posiciones de frecuencia desplazadas  $1/1ms = 1Khz$ . De este modo los símbolos de  $6Khz$  y  $7Khz$  constituyen señales ortogonales.

La expansión FH-SS sigue manteniendo 5 canales ortonormales, donde situar la portadora mediante desplazamientos en frecuencia. Estos desplazamientos obedecen a un código pseudoaleatorio conocido por emisor y receptor. Cada uno de los canales posee dos frecuencias, una correspondiente al bit '0' y otra para el correspondiente '1'. Los canales en frecuencia elegidos son: Canal(0): (6-7 KHZ), Canal(1): (8-9 KHZ), Canal(2): (10-11 KHZ), Canal(3): (12-13 KHZ), Canal(4): (14-15 KHZ).

La elección de estas frecuencias, además de mantener la compatibilidad con simulaciones anteriores, asegura que la señal FH-SS se mantiene en el rango en el que el canal se comporta mejor.

### 7.3.2 Características del emisor

Debido a la gran cantidad de cable (aprox.  $10Km$ ) y a que por dicho cable se alimentan todas las unidades de la instalación, la corriente que consume la unidad emisora está limitada, siendo menor de  $1mA$ . En estas condiciones la potencia de la señal transmitida está limitada y fijada empíricamente, para asegurar una buena recepción en el cabezal de la instalación de riego.

Para simplificar la construcción del sistema de pruebas, el emisor sólo puede transmitir en respuesta a una petición efectuada desde el elemento receptor FH-SS. Esta petición se realiza mediante una trama enviada en sentido inverso con modulación por Manipulación de Encendido y Apagado (*On Off Keyed, OOK*), esta trama se describe con detalle en el anexo A.

La forma de onda en el dominio temporal de la señal FH-SS generada, se representa en la figura 7.4. En dicha figura se observan los distintos símbolos con frecuencia dependiente del canal seleccionado. Es interesante destacar, que todos los símbolos no tienen la misma amplitud, debido a cuestiones prácticas en la implementación de los filtros de salida, descritos en el anexo A. Aunque este problema no es grave, ya que los símbolo de frecuencias cercanas, que constituyen un canal BFSK, tienen valores de amplitud cercanos. También hay que subrayar, que la trama enviada contiene 20 bits, siendo el periodo de tiempo total necesario para completar su envío  $20ms$ .

La figura 7.5 representa la misma trama en el dominio de frecuencias, en ella se comprueba la zona del espectro ocupada por los canales FH-SS, que coincide básicamente con los 5 canales de la modulación. En el anexo A, se puede obtener información

Figura 7.4: *Señal de comunicación con modulación FH-SS enviada desde el emisor hacia el receptor, donde se comprueba la variación del canal de frecuencia utilizado en cada símbolo transmitido.*

más detallada sobre los elementos que componen el emisor.

### 7.3.3 Características del receptor

En esta sección se especifica el receptor FH-SS desarrollado para la transmisión de datos por el canal del sistema. Este receptor básicamente es similar al mostrado en la sección 4.2, aunque incluye nuevos módulos encargados de la sincronización de los símbolos transmitidos, de adecuar la entrada y la salida de las señales hacia el canal de comunicación, y de la conexión del receptor con los demás elementos del sistema para el intercambio de la información. De todos ellos, el único digno de mención es el módulo de sincronización, que se explica con mayor detalle lo largo del presente apartado.

Los bloques en los que se divide el receptor son:

- En primer lugar hay un módulo de interface con la línea, que filtra la señal de potencia y extrae la FH-SS. Este módulo es totalmente analógico.
- La salida del filtro se muestrea con un convertidor analógico digital de audio. Este convertidor tiene una frecuencia máxima de  $48Khz$ , suficiente para adquirir

Figura 7.5: *Representación en frecuencia de la señal de comunicación con modulación FH-SS, enviada desde el emisor hacia el receptor. La señal queda concentrada en el intervalo de frecuencias (6 – 15)Khz, puesto que se han elegido 5 canales para la expansión FH-SS en ese rango de frecuencias.*

las señales FH-SS transmitidas, que están en el rango ( $6KHz - 15KHz$ ). Seguidamente la trama se almacena en memoria para su posterior procesamiento. El tamaño de este vector almacenado está influenciado por el periodo de muestreo.

- Una vez alcanzado este punto, la señal se multiplica con los correladores en cuadratura, habiendo una pareja de correladores por cada uno de los canales de las frecuencias utilizados en la expansión FH.
- El siguiente bloque del receptor es el filtro paso-bajo, que elimina las componentes de alta frecuencia no deseadas. Para ello, se aplica sobre la señal un filtro IIR paso-bajo.
- El módulo que viene a continuación es el filtro de supresión de interferencias. En este caso el algoritmo programado es el dado en la figura 4.6 del apartado 4.3.3, pero con estructura *Lattice* en el cálculo de la UWPT. El filtrado se realiza si la energía que presenta el vector de entrada al algoritmo es mayor que un cierto umbral, determinado empíricamente en la instalación de pruebas. Este proceso se realiza sobre cada uno de los vectores de entrada, después de haber realizado la correlación y el filtrado.
- Una vez obtenida la señal libre de interferencia es necesario sincronizar la recepción de los símbolos, este proceso se explica en este mismo apartado.

### 7.3. Construcción del emisor y el receptor con modulación FH-SS

---

- Posteriormente, se suman todos los puntos del vector que forman un mismo símbolo.
- La recepción se concluye con la combinación de las salidas de los integradores en cuadratura. El resultado es un vector con 10 elementos, cada uno para una frecuencia dada en la modulación FH (dos frecuencias por cada uno de los 5 canales). Para recuperar los bits, el detector elige el mayor entre los dos valores asociados a las frecuencias BFSK originales, desplazadas hasta las posiciones dictadas por la expansión FH.

De forma esquemática, todos los módulos del receptor se muestran en la figura 7.6.

*Figura 7.6: Módulos que integran el receptor desarrollado dentro del proyecto PRIFEM. El primer módulo es analógico y forma el interface con la línea. El segundo módulo es el conversor analógico/digital, que muestrea la señal. Los correladores y el filtro paso-bajo llevan a banda base la señal. El filtro de supresión elimina las interferencias. El sincronizador determina el instante de comienzo de los símbolos y el último módulo completa la recepción no-coherente.*

#### **Sincronización del emisor y el receptor**

A lo largo de todo el trabajo de investigación, siempre se ha asumido que existe una sincronización perfecta entre emisor y receptor a nivel de símbolo transmitido. Pero en la aplicación desarrollada, esta sincronización debe llevarla a cabo el receptor, siendo una cuestión muy importante de la que depende en gran medida la obtención de una baja probabilidad de error. El proceso de sincronización, tal como se describe en [72], se descompone en dos pasos:

1. Sincronización gruesa: es el proceso por el cual el generador de la secuencia pseudoaleatoria del receptor y del emisor se alinean. Los errores de sincronización después de este paso están acotados por el periodo de un símbolo. De esta manera, el receptor conoce el canal de frecuencia en la modulación FH-SS con cambios definidos por la secuencia PN, en el que se sitúa el último símbolo.

2. Sincronización fina: es la estimación del instante de comienzo y finalización de los símbolos transmitidos.

La solución a la sincronización gruesa ya ha sido considerada en el apartado 7.3.2. Para alcanzar este cometido se han restringido los instantes en los que se puede transmitir una trama: el emisor sólo puede enviar una respuesta a una petición realizada desde el receptor. Esta técnica es similar a los protocolos de acceso al medio del tipo *maestro/esclavo* [79], en los que una unidad esclava sólo puede acceder al canal en respuesta a una petición realizada desde el maestro. El tiempo de respuesta del emisor no es conocido, aunque está formado por un valor fijo sobre el que se aplican pequeños desplazamientos aleatorios con valores menores a  $1ms$ . De esta forma, es posible asegurar el instante en el que se va a recibir la trama, con un error menor al periodo de 1 bit.

En el muestreo de la señal hay que tener en cuenta esta imprecisión. En concreto, el muestreo debe comenzar con un intervalo de anticipación  $T_a$  igual al periodo de un bit o un símbolo  $T_a = T_b$ , ya que la modulación es BFSK. Luego se continúa la captura de la señal durante el periodo de la trama ( $T_{FH-SS}$ ) y se añade un periodo restante ( $T_r$ ), hasta completar un número adecuado de muestras  $N$ . Es deseable, que el valor  $N$  sea potencia de 2, para simplificar la programación de los algoritmos posteriores. El número de puntos del vector  $N$  se relaciona con estos tres intervalos de tiempo a través del periodo de muestreo  $T_s$ :

$$N \cdot T_s = T_a + T_{FH-SS} + T_r \quad (7.1)$$

Pero la trama adquirida puede no estar centrada en el vector muestreado, ya que  $T_a$  y  $T_r$  pueden ser diferentes. Por lo tanto, es necesario sustituir  $T_a$  y  $T_r$ , por dos nuevos periodos  $T'_a$  y  $T'_r$  que sean iguales y cumplan que  $T'_a \geq T_b$ . La forma de elegirlos es:

$$T'_a = T'_r = (N \cdot T_s - T_{FH-SS})/2 \quad (7.2)$$

Existen otras alternativas para la realización de la sincronización gruesa. Un ejemplo es la introducida en [18], que considera la detección de los pasos por cero de la señal de alimentación, para determinar los instantes de comienzo de la transmisión.

El establecimiento de la sincronización fina se ha desarrollado mediante un método basado en [16], [72] y [26]. Para implementar este método, la trama debe contener una cabecera con un número determinado de bits, cuyos valores sean conocidos de antemano por emisor y receptor. Esta cabecera puede tener un número mayor o menor de bits, dependiendo de las condiciones del canal. Durante la sincronización, el receptor debe buscar el patrón formado por dicha secuencia de bits dentro del

### 7.3. Construcción del emisor y el receptor con modulación FH-SS

---

vector muestreado. El proceso de búsqueda se ve asistido por la siguiente función de sincronización:

$$D(j) = \sum_{c \in C} \sum_{n=j+M \cdot p(c)}^{j+M \cdot (1+p(c))} |x_{cr}(n)|^2 + |x_{ci}(n)|^2 \quad (7.3)$$

donde  $D(j)$  representa la función cuyo valor hay que maximizar dependiendo de  $j$ , que indica el punto de inicio dentro del vector muestreado, y  $C$ , que representa el conjunto de todas las frecuencias correspondientes a los símbolos que forman la cabecera. El valor  $p(c)$  es la posición que ocupa cada frecuencia en dicha cabecera y  $M$  es el número de puntos que forman un símbolo. En último lugar,  $x_{cr}(n)$  y  $x_{ci}(n)$  son las secuencias de salida de los correladores en cuadratura para la frecuencia  $c$ . En la ecuación 7.3, se supone que una frecuencia determinada sólo puede aparecer una única vez en la cabecera.

Para determinar el valor  $j$ , que maximiza la función de sincronización, hay que calcular los valores de la función en un intervalo de periodo  $2ms$ , definido como el periodo de precisión de la sincronización gruesa. El valor de  $j$  que maximiza la función constituye el punto de inicio de la cabecera y de la trama. En el sistema considerado, al ser la frecuencia de muestreo de  $48Khz$ , el número total de puntos en los que se debe calcular la función es 96.

Gráficamente el proceso de sincronización fina se muestra en la figura 7.7. En dicha figura se representa una secuencia de 5 bits que supelementalmente compone la cabecera de la trama. Sobre esta cabecera se desplazan los sumadores, representados en los intervalos inferiores, que calculan un valor de la función  $D(j)$  para cada valor de inicio temporal  $j$ . El valor  $j$ , que mejor ajuste los intervalos inferiores con los periodos de los símbolos, constituye el punto de inicio de la trama.

Al definir la función de sincronización 7.3, no se ha ponderado la contribución de cada símbolo al sumatorio final. Esta situación tiene un inconveniente, ya que la forma de onda de la señal FH-SS enviada por el receptor, tal como se muestra en la figura 7.4, no tiene la misma amplitud para todos los símbolos transmitidos. La solución a este problema pasa por la introducción de unos pesos, que igualen la contribución de todos los símbolos en la ecuación 7.3. Además estos pesos dependerán de las características del cableado, ya que la atenuación sufrida por cada símbolo transmitido dependerá de la frecuencia que ocupe:

$$D(j) = \sum_{c \in C} W_c \sum_{n=j+M \cdot p(c)}^{j+M \cdot (1+p(c))} |x_{cr}(n)|^2 + |x_{ci}(n)|^2 \quad (7.4)$$

Figura 7.7: *Proceso de sincronización realizado en el receptor. Sobre la cabecera formada por 5 bits se aplica la función de coste 7.3,  $D(j)$ , con valores de  $j$  en un intervalo de periodo 2ms, en total 96 puntos con una frecuencia de muestreo de 48KHz. El valor  $j$  que maximiza  $D(j)$  es el punto de inicio de la trama. En la figura se representan los dos niveles de los sumatorios, primero el sumatorio de la señal de salida de los correladores y posteriormente el sumatorio de todos los símbolos de la cabecera.*

Esta ecuación es similar a 7.3, salvo en los pesos  $W_c$ . Con ellos se pretende igualar la contribución de todos los símbolos en el sumatorio final, sin dependencias con la frecuencia del símbolo transmitido.

El anexo A trata con mayor profundidad la construcción física del receptor, mostrando los dispositivos que lo componen y el software desarrollado.

## 7.4 Resultados de campo

El experimento llevado a cabo en la instalación de prueba compara la recepción de tramas a través del cable de potencia. Para comprobar las prestaciones del sistema propuesto, se ha medido la tasa de bits erróneos en el receptor con y sin módulo de supresión. En las pruebas se ha contado con un emisor situado en un punto fijo del cableado de la instalación, un receptor conectado en el cabezal y se han tomado los siguientes parámetros:

## 7.4. Resultados de campo

---

- El módulo de sincronización se sitúa dentro del receptor, justo a la salida del filtro de supresión, es decir, primero se realiza el filtrado de las interferencias siempre que sea necesario.
- Tal como se ha indicado en el apartado 7.3.2, la trama contiene  $20\text{bits}$ , cada uno de los cuales tiene un periodo de  $1\text{ms}$ . Al muestrear dicha señal con una frecuencia de  $48\text{KHz}$ , se obtienen 48 muestras por bit, siendo el número total de puntos necesario para completar la trama 960.
- Junto con los 960 puntos que componen la trama, se adquieren otros 1088 para completar 2048 puntos, siendo éste el tamaño del vector de entrada al filtro de supresión. Los 1088 puntos se dividen en dos intervalos de 544. El primer intervalo de puntos añadido antes de la trama forma el periodo  $T'_a$ , y el segundo intervalo del mismo tamaño constituye el periodo  $T'_r$ . Estos intervalos se han ajustado empíricamente sobre la instalación de prueba.
- El umbral de energía, a partir del cual se considera que la salida de los correladores está afectada por interferencia, se ha determinado mediante mediciones empíricas. Para obtener dicho valor se ha medido la energía de símbolo a la salida de los correladores y se ha multiplicado por 20. Como la trama está compuesta por 20 bits, este umbral asegura que la energía de la señal es mayor que la máxima energía, que puede tener una trama de símbolos transmitiendo el mismo bit por el mismo canal.
- También ha sido necesario realizar una pequeña modificación en el algoritmo de supresión considerado hasta el momento. Esta modificación se ha introducido debido a que la interferencia, como se ha mostrado en el apartado 7.2, sitúa los tonos en múltiplos de frecuencias  $250\text{Hz}$ . En un canal puede haber más de un tono de interferencia porque los canales tienen un ancho de  $1\text{KHz}$ . Ante esta situación se ha optado por permitir la repetición del filtrado tantas veces como sea necesario, para eliminar las distintas zonas del canal que hayan sido afectadas. Es decir, si después de completar el filtrado de supresión el vector de salida sigue conteniendo un nivel de interferencia que supera el umbral, el vector se vuelve a filtrar.

El número de iteraciones llevadas a cabo para obtener los resultados mostrados en la gráfica 7.8, incluye el envío y la recepción de un número de bits en el orden de  $5 \cdot 10^5$ . Este número de tramas es suficiente, para que los resultados estadísticos estén en un intervalo de confianza con orden de magnitud menor, que los datos de la probabilidad de error presentados en la figura 7.8.

En la gráfica 7.8, la variable dependiente ha sido el número de bits utilizados en la sincronización fina de las tramas, este parámetro se ha explicado en el apartado 7.3.3 al introducir el receptor.

A la vista de los resultados de la gráfica 7.8 se puede afirmar, que la interferencia generada por el variador y la bomba es capaz de degradar bastante el rendimiento del sistema, puesto que las estimaciones de la probabilidad de error, cuando el sistema no se ve afectado por la interferencia, están en el orden de  $10^{-5}$ . Por otro lado, los resultados obtenidos con el filtro de supresión indican, que el receptor con supresión realmente consigue reducir la tasa de errores. Aunque la diferencia en la tasa de error no es muy elevada, porque la potencia de la interferencia es pequeña en comparación con la potencia de la portadora. Se obtendrían mayores diferencias, si la potencia de la interferencia aumentara, como se puede observar en la gráfica 6.2.5, donde se simula el aumento gradual de la potencia de la interferencia.

Otra punto a destacar en los resultados de la gráfica 7.8, es el aumento de la probabilidad de error cuando el número de bits utilizado en la sincronización es pequeño. Esta degradación se debe a que no sólo se producen errores por la interferencia, sino que la sincronización también es un factor a tener en cuenta.

Figura 7.8: *Espectro de la interferencia medida a la entrada del receptor FH-SS, en la instalación de prueba, con la bomba y el variador en funcionamiento.*

## 7.5 Conclusiones

En este capítulo se ha aplicado el algoritmo de supresión de interferencias en un sistema de comunicación por cable de potencia. Los elementos emisor y receptor se han integrado en una arquitectura para la gestión y control de un sistema de riego localizado, formando el nivel de comunicación más bajo que interconecta los elementos remotos

## 7.5. Conclusiones

---

distribuidos en una amplia superficie de terreno, tal como se describe en el anexo A. El estudio del canal ha demostrado, que esta aplicación constituye un buen escenario donde aplicar el algoritmo de supresión desarrollado, puesto que la interferencia más importante encontrada es del tipo CWT y está formada por varios armónicos.

En el capítulo se ha mostrado la construcción de los elementos emisor y receptor. Este último, está basado en el receptor FH-SS mostrado en capítulos anteriores pero con dos modificaciones. La primera se refiere al módulo de sincronización, que determina el instante de comienzo y finalización de cada bit de la trama. La segunda modificación es necesaria para adaptar el algoritmo de supresión, al caso en el que varios tonos de interferencia actúan sobre el mismo canal. Este obstáculo se ha salvado aplicando el algoritmo de filtrado tantas veces como sea necesario, hasta que la señal de salida no supere el umbral de energía. Esta solución aunque es costosa computacionalmente es efectiva, tal como han demostrado los resultados obtenidos.

Para terminar es interesante resaltar, que el sistema PRIFEM expuesto ha sido totalmente implementado y sus especificaciones han sido validadas en la instalación piloto. Esta instalación, mostrada en el anexo A, ha permitido validar con resultados experimentales del receptor FH-SS propuesto.

## Parte V

### Conclusiones y trabajo futuro



# Capítulo 8

## Conclusiones y trabajo futuro

### 8.1 Introducción

Tras haber analizado y diseñado el algoritmo de supresión de interferencias, en modulaciones de Espectro Ensanchado con Saltos de Frecuencia, que constituye el objetivo de la presente tesis, se han realizado una serie de experimentos, tanto simulados como sobre prototipo, cuyos resultados han permitido evaluar el algoritmo y analizar sus prestaciones.

Dichos resultados han permitido afianzar los objetivos marcados al comienzo de la investigación, es decir, conseguir una baja probabilidad de error en la transmisión de información a través de canales interferentes. En este capítulo se resaltan los pasos realizados y las conclusiones a las que da lugar el trabajo realizado.

Dentro de este capítulo, en primer lugar se han enumerado las aportaciones relevantes y novedosas contenidas en la investigación realizada. A continuación, la sección 8.3 contiene las conclusiones que se derivan de los resultados obtenidos en la tesis. Para concluir, en la sección 8.4 se exponen las líneas que han quedado abiertas y que constituyen el trabajo futuro que se realizará como continuación de la presente tesis.

### 8.2 Aportaciones

La línea de investigación básica, en la que se ha centrado el trabajo, ha sido el análisis, diseño e implementación del receptor para modulaciones en Espectro Ensanchado con Saltos en Frecuencia (*Frequency Hopping Spread Spectrum, FH-SS*), junto con el algoritmo de supresión de interferencias basado en la Transformada Sobremuestreada en Paquetes Wavelet (*Undecimated Wavelet Packet Transforms, UWPT*). Las aportaciones de la tesis se enmarcan dentro de esta línea, y han ido surgiendo en cada

uno de los pasos en los que se ha dividido la investigación. De forma resumida dichas aportaciones son:

1. Propuesta de un nuevo receptor para modulaciones FH-SS.

Para incluir el algoritmo de supresión de interferencias ha sido necesario modificar el receptor FH-SS no-coherente. En el nuevo receptor propuesto, se añade un módulo de supresión que se sitúa a la salida de cada uno de los correladores integrados en el receptor.

Es interesante destacar, que el receptor propuesto es independiente del algoritmo de filtrado que finalmente se aplique, ya que es posible utilizar algoritmos basados en transformada o en filtros adaptativos temporales, como los introducidos en la sección 3.4.

2. Análisis y diseño del algoritmo de supresión.

Este algoritmo constituye la pieza fundamental de la investigación realizada. Las demás aportaciones se han ido introduciendo, conforme se ha completado el proceso de diseño e implementación de dicho algoritmo.

La premisa básica, con la que se comenzó la fase de análisis, fue la necesidad de concentrar al máximo la energía de la interferencia en una única subbanda, y eliminar el menor intervalo de frecuencias posible para no degradar la señal FH-SS, en el momento de la extracción de la interferencia. Con este objetivo, el algoritmo realiza una descomposición invertida mediante UWPT. Dicha transformación comienza desde el nivel de mayor resolución frecuencial hasta el de menor resolución.

La propiedad que permite al algoritmo concentrar la interferencia, sin disparar el coste computacional, se fundamenta en el alineado de las zonas de corte y paso-banda de los filtros de diferentes niveles. Así, en el primer nivel de resolución del algoritmo, se alinea la interferencia con la zona paso-banda de uno de los filtros, mediante desplazamientos en frecuencia de la señal. Pero en los demás niveles, debido al alineado del espectro de los filtros, ya no es necesario comprobar varios desplazamientos en frecuencia de la señal, puesto que el alineado en el primer nivel asegura que la señal seguirá estándolo en los posteriores, como se ha explicado en el capítulo 4.

Para concluir el proceso, antes de realizar la reconstrucción, la subbanda obtenida en el último nivel se anula, pues contiene la mayor parte de la interferencia.

3. Implementación de la transformada UWPT mediante estructuras *Lattice*

La implementación del algoritmo propuesta se ha fundamentado en las estructuras de filtros *Lattice*. Dichas estructuras permiten una implementación eficiente de las Transformadas Wavelets como se ha demostrado en la sección 2.7. El trabajo aportado en la tesis ha consistido en proponer una implementación correspondiente al algoritmo de filtrado mediante UWPT, a base de secciones *Lattice*.

Esta estructura permite una construcción modular de los filtros incorporando secciones adicionales, y un coste computacional reducido medido mediante el número de sumas, multiplicaciones, accesos y posiciones de memoria.

4. Construcción del algoritmo propuesto sobre Procesador Digital de Señal.

Para obtener de forma analítica el coste computacional de la implementación real del algoritmo, sobre un dispositivo del tipo Procesador Digital de Señal (*Digital Signal Processor, DSP*), se ha propuesto una subrutina con estructura *Lattice* desarrollada en el lenguaje ensamblador del DSP *ADSP2065L SHARC*, que mantiene ocupados en cada ciclo de instrucción todas las unidades internas del dispositivo.

El resultado de esta implementación se ha restringido a una estructura formada por dos secciones *Lattice*, aunque fácilmente puede extenderse a casos con un número mayor de secciones.

5. Aplicación a una red de transmisión de datos sobre línea de potencia eléctrica.

El proyecto PRIFEM, mostrado en el anexo A, ha constituido un buen banco de pruebas donde comprobar la validez experimental del algoritmo propuesto. Además, dicho algoritmo junto con las modulaciones FH-SS han servido para construir una red de control industrial distribuido, que conecta a través del cable de potencia de la instalación los elementos remotos y la unidad central que genera las órdenes de control. Dentro de dicho desarrollo se han diseñado las unidades de emisión y de recepción, que generan la señal con modulación FH-SS, en el sentido emisor hacia receptor, y la señal con modulación mediante Manipulación por Encendido y Apagado (*On Off Keyed, OOK*) en sentido receptor a emisor.

Por último es interesante resaltar, que este desarrollo es totalmente novedoso y puede aplicarse a otros sistemas de control, que presenten similitudes en el cableado y los requisitos temporales de la red.

## 8.3 Conclusiones

Los resultados recabados en los capítulos 6 y 7 han sido obtenidos de forma simulada sobre Matlab, y en una instalación real mediante un prototipo.

En el caso de los resultados alcanzados mediante simulación, en primer lugar hay que destacar, que el algoritmo de supresión de interferencias presenta mejores prestaciones que el receptor FH-SS no-coherente sin supresión, para interferencias del tipo Onda con Tono Continuo (*Continuos Wave Tone, CWT*), como cabría esperar. La reducción de la probabilidad de error que presenta el receptor con filtro de supresión es significativa, y justifica el incremento en la complejidad del receptor.

### 8.3. Conclusiones

---

También supera al receptor con supresión de interferencias con algoritmo basado en Transformada de Fourier Rápida (*Fast Fourier Transform, FFT*), ya que para todas las posiciones de frecuencia, en las que se puede situar la interferencia dentro de un canal de la expansión FH-SS, el algoritmo con UWPT consigue una mayor reducción de la probabilidad de error.

También se extrae como conclusión de las simulaciones llevadas a cabo, que la expansión con Saltos Rápidos (*Fast-Hopping, FFH-SS*), que genera varios *chips* por bit transmitido, para niveles altos de la potencia de la interferencia, no consigue ofrecer las mismas prestaciones que el receptor con FH-SS y filtro de supresión. Este resultado es importante porque la degradación sufrida por el receptor FH-SS, puede evitarse al considerar expansiones FFH-SS, como se demostró en la sección 3.3, pero este cambio en la modulación introduce dos inconvenientes. El primero es la necesidad de mayor ancho de banda para obtener la misma velocidad de transmisión. El segundo inconveniente es la degradación de la probabilidad de error en escenarios sin interferencia debido a la combinación no-coherente de la energía de los bits. De este modo, el receptor FH-SS con supresión de interferencias consigue aumentar las prestaciones sin introducir estas limitaciones.

Al considerar otros tipos de interferencia, caso del tipo gaussiano de banda estrecha, se puede constatar que el filtro de supresión es efectivo, sólo si el ancho de banda ocupado por la interferencia es pequeño en comparación al intervalo ocupado por un canal de la expansión FH-SS. Cuando la zona del espectro que ocupa la interferencia aumenta, también lo hace la probabilidad de error, llegando a un punto en el que la degradación sufrida en el receptor sin filtro de supresión es similar al caso del receptor con filtro.

Desde el punto de vista del coste computacional, existen tres parámetros clave con los que se controla el coste y las prestaciones del algoritmo. Estos tres parámetros son: el tamaño del bloque  $N$ , el incremento del desplazamiento en frecuencia  $\Delta d$  y el nivel inicial de la descomposición  $J$ , que determina el número total de niveles. Las simulaciones realizadas con interferencia del tipo CWT, para diversos valores del bloque  $N$ , muestran que el incremento de este parámetro aumenta la resolución frecuencial de la transformada y por lo tanto las prestaciones del sistema.

Al variar el segundo parámetro, el incremento del desplazamiento en frecuencia  $\Delta d$ , varía el número de iteraciones que se realiza en el primer nivel de la descomposición. Cuanto menor es  $\Delta d$ , mayor es el número de iteraciones necesario para cubrir todo el rango de frecuencias, definido a partir de la zona paso-banda de uno de los filtros. Los resultados obtenidos demuestran que cuanto menor es  $\Delta d$  mejores prestaciones se obtienen, porque la interferencia se centra en la zona paso-banda de uno de los filtros con mayor resolución.

El último parámetro  $J$  define el número de niveles de resolución comprendidos en la descomposición. Las simulaciones llevadas a cabo variando este parámetro aconsejan

tomar el valor  $J = 9$  como el óptimo. Esto es así porque para niveles iniciales inferiores ( $J = 8$ ), el rango de frecuencias del espectro eliminado es más amplio de lo necesario, y la señal FH-SS se ve degradada. Por otro lado, para valores iniciales superiores ( $J = 10$ ), el rango de frecuencias es demasiado estrecho, y una parte de la energía de la interferencia permanece tras el filtrado. Sin embargo, este valor va muy ligado al tipo de interferencia y el tamaño del bloque  $N$  escogido.

Las pruebas simuladas también se han repetido con el dispositivo DSP *ADSP2065L*. El objetivo de estas nuevas simulaciones ha sido la comprobación de la concordancia, de los resultados obtenidos mediante implementación en forma directa sobre Matlab, frente a los alcanzados mediante estructura *Lattice* sobre DSP. Estas simulaciones han validado la estructura propuesta, ya que han dado como resultado una probabilidad de error coincidente con la lograda anteriormente, y también han constatado la validez de la fórmula para el cálculo del coste computacional, que presenta el algoritmo con estructura *Lattice*.

En el caso de los resultados obtenidos sobre prototipo, lo primero que es necesario resaltar es que el nivel de interferencia del tipo CWT, con la bomba y el variador en funcionamiento contenido en la red es alto, pero no lo suficiente como para hacer imposible la comunicación. La probabilidad de error está en el orden de  $(4.5) * 10^{-4}$ , y mediante la detección de los errores a partir de algún código de detección y el reenvío de las tramas corrompidas, se podría mantener un nivel de prestaciones adecuado a las demandas de la aplicación. Por tanto, la modulación FH-SS ofrece un nivel de inmunidad suficiente en este caso, aunque existe la posibilidad de que en diferentes instalaciones el nivel de la interferencia sea mayor.

## 8.4 Trabajo futuro

En el desarrollo del trabajo se han ido tomando ciertas decisiones, que han dejado de lado algunos aspectos interesantes. Estas líneas que han quedado abiertas merecen ser señaladas, ya que constituyen vías por lo que se puede continuar la investigación emprendida. Para organizar la exposición, las nuevas alternativas se van a agrupar en diferentes contextos:

### Variaciones del algoritmo

En primer lugar es interesante considerar posibles modificaciones del algoritmo de supresión de interferencia, ya que sus prestaciones son muy importantes de cara a la obtención de un buen sistema de filtrado de interferencias. Las variaciones que se plantean son:

- En las pruebas realizadas en la tesis, se han utilizado los filtros de Daubechies 4, por ser ortonormales, permitir una implementación eficiente mediante filtrado *Lattice* y presentar una buena caída en frecuencias. Pero parece lógico que el primer punto del trabajo futuro sea la aplicación de otros filtros que permitan aumentar el rendimiento del receptor.

### Aplicaciones sobre modulaciones SS

En segundo lugar se puede introducir nuevas aplicaciones del algoritmo relacionadas con las modulaciones SS y nuevos tipos de interferencias. En la memoria se han tratado algunos casos de modulaciones SS, pero han quedado casos sin considerar que pueden resultar interesantes:

- El primer caso que se puede considerar es la adaptación del algoritmo de supresión a las modulaciones FFH-SS. Es lógico pensar, que esta aplicación debe resultar beneficiosa, puesto que el fundamento de la modualción FFH-SS es similar a la del caso FH-SS. Un resultado interesante que se podría obtener de dicha aplicación es la comparación con los receptores con combinación no-lineal de la energía de los chips. Estos receptores introducidos en la sección 3.3, son por lo general más simples de construir, pero el aumento en el rendimiento con respecto al caso FH-SS no es muy elevado, como se puede ver en la gráfica 6.5. De esta manera, el incremento en la complejidad del receptor puede compensarse con un aumento en las prestaciones del sistema.

También es interesante observar la comparación de las modulaciones FH-SS y FFH-SS, para saber si es provechoso el incremento en el número de chips por bit al aplicar el algortimo de supresión a FFH-SS.

- El segundo caso estudia la posible adaptación del algoritmo para su utilización con otras modulaciones SS. La aplicación a modulaciones DS-SS es importante porque permitiría comparar el algoritmo propuesto, en coste computacional y rendimiento, con otros algoritmos presentados en la bibliografía sólo para modulaciones DS. Entre estos algoritmos destacan los comentados en la sección 3.4, como el algoritmo con supresión Adaptativa en Tiempo-Frecuencia (*Adaptive Time-Frequency, FTA*), los basados en Transformada Wavelet (*Wavelet Transform, WT*) y en bancos de filtros.
- Dentro de este ámbito también es posible plantear escenarios con otros tipos de interferencias. Los casos considerados en las simulaciones realizadas sólo han tenido en cuenta interferencias estacionarias del tipo CWT y gaussiano. Las interferencias no estacionarias se podrían modelar como en [47], mediante una señal gaussiana de banda estrecha con un periodo de encendido y apagado.

### **Nuevas aplicaciones de la investigación**

En cuarto lugar, aparecen las posibles adaptaciones del emisor y el receptor propuestos dentro del proyecto PRIFEM a otras situaciones o aplicaciones. También se pueden considerar otras modulaciones que aumenten las prestaciones y las velocidades de transmisión, y permitan incorporar nuevas funcionalidades al sistema.

#### 8.4. Trabajo futuro

---

# Parte VI

## Anexos



# Apéndice A

## Arquitectura de control PRIFEM

### A.1 Introducción a la arquitectura del sistema

En este anexo se introducen los elementos que componen el sistema PRIFEM, utilizado como banco de pruebas en el capítulo 7. La exposición que se va a realizar se centra sobretodo en los dos elementos que implementan la comunicación por el cable de potencia. Estos elementos se han denominado dentro de la arquitectura propuesta como: **Unidad Remota Autónoma, (URA)**, que se corresponde con el **Emisor** del capítulo 7, y **Unidad de Control Emisora, (UCE)** definido como **Receptor** definido en el capítulo 7.

La arquitectura desarrollada esta compuesta por cuatro niveles, siguiendo el esquema dado por el modelo CIM <sup>1</sup> [68]. Este modelo es utilizado en la mayoría de las aplicaciones industriales. La división en niveles o capas de la arquitectura CIM, permite un diseño modular del sistema e incluye la especificación del sistema, desde las ordenes sobre los sensores situados en el nivel más bajo, hasta la información corporativa orientada a la gestión del sistema, en el más alto nivel. En la figura A.1 se muestra esquemáticamente esta arquitectura con las tecnologías de red asociadas a cada nivel.

#### A.1.1 Arquitectura PRIFEM

A continuación se muestra el esquema básico, de la arquitectura de control desarrollada, dentro del proyecto PRIFEM. Esta arquitectura jerárquica está dividida en los niveles recomendados por el modelo CIM. Cada uno de los niveles, junto con las funciones y requisitos que debe cumplir son:

---

<sup>1</sup>Sistemas de Fabricación Integrados por Computador (*Computer Integrated Manufacturing, CIM*)

Figura A.1: *Esquema de la arquitectura CIM.*

### **Nivel de planificación de la producción y gestión**

Este es el nivel más alto. Representa el nivel donde la información es recogida, almacenada y procesada para obtener informes de producción, costes, gestión, calidad, etc. En concreto este nivel centraliza toda la gestión. Las tareas básicas encomendadas a este nivel son:

- Indicar cual es la cantidad de recursos consumidos detallados para cada usuario.
- Gestionar la contabilidad de cada usuario.
- Gestión de los recursos hídricos en función de la capacidad de la red hidráulica.
- Gestión de alarmas.
- Conexión con el exterior mediante telefonía, tanto de red básica como GSM.

Este nivel esta centralizado y constituido por un único elemento. No existen requisitos temporales en la recogida de la información, ni es necesaria un gran capacidad de almacenamiento de datos. Todas las tareas propuestas en este nivel se han implementando por software sobre un computador tipo PC. Esta unidad ha sido denominada **Unidad Central, UC**.

## Nivel de Supervisión de procesos

Es responsable de de la Fabricación Asistida por Computador (*Computer Aided Manufacturing, CAM*) y la gestión de Calidad Asistida por Computador (*Computer Aided Quality, CAQ*).

Este nivel contiene los elementos encargados de supervisar toda la instalación, coordinando el control de todos los elementos para que vayan acompasados. Se comunica con el nivel superior para obtener la información, de cuándo y cómo actuar sobre los elementos de la instalación. Con el nivel inferior, intercambia las ordenes de sincronización y la información de posibles anomalías. También es el encargado de recoger los datos provenientes de los sensores, y comunicarlos para su almacenamiento en el nivel superior.

Al igual que en el nivel de planificación, las necesidades del sistema sugieren una implementación por software ejecutada sobre un computador personal. Este computador puede ser el mismo que el que ha sido utilizado en el nivel superior. Este nivel por lo tanto, también ha sido integrado en la **UC**.

## Control de Procesos

Contiene los computadores dedicados al control de determinados subsistemas autónomos. Ejemplos de este tipo de elementos dentro de una arquitectura de control son robots, sistemas de transporte, motores, etc.

En la aplicación considerada, dentro de esta categoría se pueden encontrar: grupos de bombeo de agua, elementos inyectoros de abonado y los bancos de filtrado. Cada uno de estos subsistemas requiere de un computador propio, para su control de forma autónoma, y se comunican con el nivel superior de supervisión del control, mediante un bus de campo.

Estos computadores dedicados no requieren una capacidad procesamiento muy grande, ya que el programa de control que deben ejecutar no tiene unas restricciones temporales ni computacionales elevadas. La implementación de estos elementos se ha basado en una unidad autónoma, con una tarjeta que incluye un microcontrolador y un software de control específico. Esta unidad se denomina **Unidad Distribuida Autónoma, UDA**

## Nivel de sensores y actuadores

Es el nivel más bajo. Su interconexión con los elementos del nivel de control de procesos se realiza a través de buses de campo, como son por ejemplo: CAN, Interbus, Profibus, WorldFip, Lonworks, etc, [36], [68] o en domótica CEBUs [62], [20], [21] y Lonworks

## A.1. Introducción a la arquitectura del sistema

---

[68].

Este nivel contiene los sensores y los actuadores que realizan físicamente el control de los elementos de la instalación. Las características específicas de cada uno de los dos tipos de dispositivos son:

- Los sensores se pueden clasificar en analógicos y digitales. Normalmente, la señal de salida de los sensores analógicos es una señal de corriente dentro del rango estándar  $4 - 20mA$ . Los sensores pertenecientes a esta categoría miden presión hidráulica, PH y conductividad del agua (nivel de sales). Por otro lado los sensores digitales son contactos, que se abren o cierran proporcionando impulsos de tensión. Este tipo de sensores se encuentran en caudalímetros de agua y abonado.
- Los actuadores son salidas reguladas provenientes de los computadores de la arquitectura. La mayoría de las estas salidas son digitales, y generan una señal de tensión. Suelen actuar sobre relés que ponen en funcionamiento diferentes elementos del sistema como son: las bombas inyectoras de abonado, los agitadores

La arquitectura especificada pueden contener un gran número de hidrantes, distribuidos por toda la instalación. Por lo tanto, es imposible acceder a cada uno de estos sensores y actuadores mediante un cable individual. Por razones de fiabilidad y de coste económico, todos los hidrantes se alimentan y se comunican a través de un único cable que recorre toda la instalación. El único nexo de unión entre hidrantes y sistema es un cable bifilar de potencia, cuyas características se dan en el apartado 7.2. Este cable es el encargado de llevar la energía de funcionamiento y la señal de comunicación.

Las características y la estructura de la red mostradas en el sección 7.2 y en el gráfico de la instalación piloto A.15 hacen imposible la aplicación de un red de control industrial comercial. Esta ha sido la razón por la que ha sido necesario desarrollar una red de comunicación a medida.

Además las necesidades de alimentación y la salida para cada tipo de sensor es distinta, por lo tanto ha sido necesario desarrollar e implementar una tarjeta, denominada **Unidad Remota Autónoma, (URA)** definida como **Emisor** en el capítulo 7, situada en el hidrante que realiza las siguientes tareas:

- Alimentar de la forma adecuada cada uno de los sensores presentes en el hidrante.
- Incluye un interface para leer la salida de los sensores.
- Incluye el interface para suministrar la señal adecuada al actuador.
- Establece la comunicación con el elemento correspondiente del sistema, para comunicar los valores medidos y recibir las ordenes sobre el actuador.

La comunicación con todos los hidrantes está centralizada. Se realiza desde un elemento situado en la unidad central del sistema que se denomina **Unidad de Control Emisora, (UCE)** definido como **Receptor** en el capítulo 7.

En la figura A.2 se muestra gráficamente todo los 4 elementos introducidos en la arquitectura PRIFEM: UC, UDA, UCE y UDA.

En la próxima sección se realiza el estudio los elementos **URA (Emisor)** y la **UCE (Receptor)** que se comunican a través del cable de potencia.

## A.2 Elementos URA y UCE

En esta sección, a partir de las especificaciones del sistema dadas en el apartado anterior, se resumen las características básicas de los elementos UCE y URA. En primer lugar se repasa los requisitos planteados en su construcción:

Figura A.2: *Esquema de la arquitectura PRIFEM donde se muestran los elementos UC, UDA, UCE y URA. Hay que destacar que los elementos UCE se corresponde con el receptor del capítulo 7 y la URA con el emisor.*

- El canal de comunicación es un cable bifilar de potencia, en el que se comparte la administración de potencia y la transmisión de datos. La topología del canal puede ser cualquiera y las distancias totales de cableado rondan los  $10Km$ .
- No hay grandes restricciones temporales.
- Es necesaria una gran fiabilidad en las comunicaciones, porque un fallo en la transmisión de la información puede dar lugar a daños graves.
- Debido a que los elementos UCE y URA se van a implementar físicamente, debe simplificarse al máximo su diseño.
- Los elementos URA tienen restricciones de consumo, que limitan las características de la comunicación.

A continuación, se estudian con más detalle las decisiones tomadas en el diseño y la implementación de la comunicación por el cable de potencia.

### A.2.1 Modulaciones y tramas

En este apartado, se comienza el diseño de la red, que constituye la comunicación entre los elementos UCE y URA. Debido a que las características y la situación física de

estas unidades son distintas, las modulaciones que mejor se adaptan en cada sentido también son distintas. Así pues, la comunicación va a ser asimétrica. El análisis de las modulaciones se divide en dos subapartados, uno para cada sentido.

### Comunicación en sentido UCE a URA

En el sentido UCE hacia URA, las restricciones que hay que tener en cuenta son:

- El canal utilizado puede contener mucho ruido, proveniente de las bombas de agua y abonado. Este ruido se puede modelar como la suma de AWGN y CWT, como se propone en el apartado 7.2.
- Las características del canal pueden cambiar dependiendo del periodo de la señal de potencia de  $24Vac$  y  $50Hz$ . Así pues, la amplitud de la señal de comunicación emitida puede variar con el tiempo, en función del periodo de la señal de potencia.
- La recepción se realiza a cierta distancia de la unidad central, donde se supone está situado el foco ruidoso. Por lo tanto, la interferencia también se atenúa, en los extremos del cableado, debido a las características resistivas del canal.
- La señal emitida por la UCE puede tener una potencia relativamente grande, porque no hay limitación en la energía consumida.
- No se van a considerar modulaciones en fase, que requieren una recepción compleja con coherencia de fase, para simplificar al máximo el diseño y la implementación de la UCE de la URA.
- Para sincronizar URA y UCE a nivel de los símbolos transmitidos, es necesario algún código sobre la trama de bits transmitidos.

Después de considerar estas limitaciones, la modulación elegida es por Manipulación de Encendido y Apagado (*On Off Keyed, OOK*). Ésta es una modulación muy sencilla, que se puede combinar con codificación Manchester a nivel de bit, para sincronizar la URA y UCE. La portadora se sitúa en  $5KHz$ , dentro del rango  $0 - 15KHz$ . Queda por tanto, alejada suficientemente de los  $50Hz$ , para no complicar demasiado el diseño de los filtros de entrada y salida a la URA. La portadora generada por la UCE tiene una amplitud elevada, del orden de  $5V$ .

En la codificación Manchester, los bit transmitidos representados como '1' se componen de dos símbolos: uno con señal de  $5KHz$ , *On*, y otro sin señal, *Off*. Por el contrario, los bits '0' se componen primero de un símbolo *Off*, y a continuación de un símbolo *On*. Así, siempre hay un cambio entre símbolo *Off* y símbolo *On* o viceversa, a mitad de bit. Esta característica sirve para sincronizar la URA y la UCE, a mitad de los bits transmitidos.

El periodo del símbolo es  $T_c = 2ms$ , produciéndose 10 oscilaciones de la señal de  $5KHz$ . Este símbolo tiene un periodo suficientemente grande, para que la URA pueda detectar fácilmente la presencia de la portadora. Como cada bit necesita dos símbolos, el periodo de un bit es  $T_b = 4ms$ . La máxima velocidad de transmisión de información teórica es:  $V_T = 1/T_b = 250bits/seg$ . Esta velocidad es bastante baja pero suficiente, ya que no hay fuertes requisitos temporales.

### Comunicación en sentido URA a UCE

En este sentido la modulación elegida va a ser diferente, debido a las siguientes limitaciones impuestas por los elementos UCE y URA:

- La señal emitida por la URA no puede tener una gran amplitud, porque su consumo de energía está limitado. Como esta señal puede enmascarse con una interferencia de potencia lo suficientemente elevada, es necesario considerar modulaciones que presenten ciertas características de inmunidad frente a interferencias.
- La UCE es una unidad que puede ser mucho más compleja que la URA, al estar cercana a la entrada de potencia del sistema. Esto permite la detección de la señal mediante técnicas de procesamiento digital y filtrado de interferencias, como las mostradas en los capítulos 3 y 4.

Todos estos requisitos han llevado a elegir la modulación FH-SS. El descarte de las modulaciones con Espectro Ensanchado codificadas en fase, como la DS-SS, o mediante barrido en frecuencia (*chirp*) se debe a que las tramas enviadas son de pequeña duración, y la sincronización de la fase en la recepción necesita un cierto periodo de tiempo, que alargaría la duración de la trama.

La modulación propuesta comienza la codificación de los bits mediante modulación BFSK. Cada bit se codifica mediante un símbolo de frecuencia  $6Khz$ , en el caso de ser un bit '0' y  $7Khz$  en el caso de ser un bit '1'. A continuación, la señal se desplaza en frecuencia y se sitúa en uno de los 5 canales ortonormales, que se van a definir. Cada uno de ellos posee dos frecuencias, una correspondiente al bit '1' y otra para el correspondiente '0'. Los canales en frecuencia elegidos son: Canal(0): (6-7 KHz), Canal(1): (8-9 KHz), Canal(2): (10-11 KHz), Canal(3): (12-13 KHz), Canal(4): (14-15 KHz).

La velocidad de la comunicación es cuatro veces mayor en este sentido, ya que los símbolos transmitidos tienen una duración mitad y no hay codificación Manchester a nivel de bit.

## Análisis del formato de las tramas

El tamaño de los datos que se envían y se reciben es del orden del byte, ya que como se ha comentado en la sección A.1.1, esta información se corresponde con las órdenes sobre los actuadores, la lectura de sensores y las variables distribuidas. Estos datos no necesitan una gran resolución. Por lo tanto las tramas enviadas, como se corresponde con un sistema de comunicación para los niveles más bajos de la jeraquía de sistemas de control CIM, deben tener campos de datos pequeños.

Al igual que en el caso de las modulaciones, el formato de la trama usado en cada sentido será distinto.

### Trama enviada desde UCE a URA

Dado que la modulación en este caso es OOK con codificación Manchester, cada bit transmitido se representa con dos símbolos. La agrupación de dos símbolos forma un bit y la agrupación de varios bits forma un campo de la trama. Esta trama contiene un campo de dirección. También contiene un campo con los datos a transmitir. Los demás campos que se añaden tienen como objetivo, aumentar la fiabilidad en la transmisión de la información. A continuación, se muestran todos los campos añadidos indicando su función:

- Inicialmente hay un preámbulo formado por tres símbolos  $On$ , para sincronizar la URA con la UCE.
- El siguiente bit es el bit de *start* y se codifica como un bit '1'. Este bit busca identificar errores de encuadre.
- El siguiente campo contiene la dirección con el que se identifica la URA destinatario de la información. Este campo se ha codificado mediante  $6bits$  y el número máximo de tarjetas admitidas es 64. Los bits se transmiten de mayor a menor significativo.
- Los siguientes dos bytes son campos de información. En ellos se envía el dato deseado a la URA. Igual que en el caso de la dirección, el bit más significativo es el primero que se transmite.
- El siguiente byte contiene el *CRC* [64]. Este código de protección contra errores se calcula sobre los 3 bytes anteriores, completando la dirección como un byte con los dos bits más altos a cero. El orden en el que se transmiten los bits es el mismo que el los dos casos anteriores.
- Por último se envía un bit de *stop* codificado como un '1', para detectar errores de encuadre.

De forma esquemática, la representación del formato de la trama se muestra en la figura A.3.

Figura A.3: *Trama de bits enviada desde UCE a URA.*

### Trama enviada desde URA a UCE

La trama enviada en este sentido no necesita campo de dirección, ya que sólo hay una unidad UCE. Tampoco es necesario el preámbulo porque la UCE se sincroniza a nivel de símbolo. Al quitar el campo de dirección, el tamaño total de la trama es 26 bits. Estos bits se agrupan en los campos mostrados en la figura A.4. El significado de estos bits coincide con el que se ha especificado para la trama enviada en sentido contrario.

Figura A.4: *Trama de bits enviada desde URA a UCE.*

- Primero se envía un bit de *start*. Este bit siempre es un bit '1'.
- A continuación se envían dos bytes de información.
- El tercer byte corresponde al CRC, calculado de forma similar al caso descrito en el apartado A.2.1. Aunque ahora sólo se calcula sobre los 2 bytes de datos.
- Por último se envía un bit de *stop*, que junto con el bit de *start* sirven para detectar posibles errores de encuadre.

### A.2.2 Implementación de la URA

La URA es un dispositivo que se sitúa a pie de parcela, alejado de la unidad central del sistema. Se conecta únicamente al cable de potencia, del cuál recibe la alimentación y la señal de comunicación. Esta unidad debe tener un consumo de potencia bajo, para que no caiga la tensión de alimentación en los extremos del cable. Está basada en un módulo de procesamiento, construido a partir de un microcontrolador de bajo coste y consumo. En concreto el dispositivo escogido es el microcontrolador *PIC16C73*,

del fabricante Microchip. El diagrama de bloques de la URA se puede observar en la figura A.5, donde se observan los filtros de entrada y salida que se detallan en la sección siguiente, el amplificador de salida y el comaprador de entrada para recuperar la señal cuadrada, el microcontrolador y la fuente de energía, que almacena carga para el disparo de la salida de pulsos.

Figura A.5: *Esquema de la tarjeta URA, con filtros de entrada y salida que se detallan en la sección siguiente, el amplificador de salida y el comaprador de entrada para recuperar la señal cuadrada, el microcontrolador y la fuente de energía, que almacena carga para el disparo de la salida de pulsos.*

La figura A.6 es una fotografía de la tarjeta URA, montada en la instalación prototipo.

## **Generación y recepción de las señales de comunicación**

El problema planteado en la generación y recepción de las señales de comunicación en esta unidad, es que este proceso es analógico. Por lo tanto, de difícil solución mediante un microcontrolador. Este apartado muestra la soluciones adoptadas, para conseguir el envío y de recepción de las ñales.

La generación de la señal FH-SS de salida se realiza mediante un pin del micronrolador. Este pin oscila entre los valores lógicos  $0V$  y  $5V$ , controlado por software para generar una determinada frecuencia. Esta señal cuadrada se amplifica y se filtra de forma pasa-banda, para eliminar los armónicos de mayor frecuencia. Así, se genera una señal anológica, que aunque no está constituida por un único armónico frecuencial, contiene su energía lo suficientemente concentrada sobre este armónico, como para poder despreciar el resto en la recepción de la señal. En las figuras 7.4 y 7.5, se muestra la forma de onda en el dominio temporal y en el dominio frecuencial respectivamente de la señal FH-SS.

El esquema del filtro de salida se pueden ver en la figura A.7 y su respuesta fre-

Figura A.6: *Fotografía de la tarjeta URA.*

cuencial en la figura A.8.

Figura A.7: *Filtro entrada.*

En la recepción, la señal proveniente de la UCE se filtra de forma pasa-banda, para separarla de la señal de alimentación situada en  $50Hz$ . El esquema del filtro de entrada se pueden ver la figura A.9 y su respuesta frecuencial en la figura A.10. A continuación, existe un comparador que la transforma en una señal cuadrada. Por último, la señal rectificada llega hasta el pin de interrupción del microcontrolador. Para evitar que ruido impulsional o interferencia de elevada potencia, pueda producir pulsos de interrupción, el umbral de comparación se sitúa suficientemente elevado.

La señal de comunicación enviada desde la UCE, necesita tener también una amplitud elevada para superar el umbral. Aunque esto no supone un problema grave, dado que la UCE está situado cerca la toma de potencia de todo el sistema, no tiene

Figura A.8: *Espectro del filtro de entrada.*

restricciones de consumo.

Figura A.9: *Filtro salida.*

### A.2.3 Implementación de la UCE

La construcción de la UCE se basa en un elemento de proceso DSP. En concreto se ha utilizado el *ADSP21065L* del fabricante *Analog Devices*, mostrado en el anexo B. El diseño del prototipo de la UCE se basa en la tarjeta de evaluación *ADSP-21065L EZ-LAB*, que integra el DSP *21065L* funcionando a  $60\text{Mhz}$ , junto con  $1\text{Mx}32\text{bits}$  de memoria SDRAM,  $128\text{kx}8\text{bits}$  de memoria EPROM, un interface RS-232 y el Codec de Audio *AD1819A* de 16 bits de resolución.

La frecuencia de muestreo máxima que permite el *Codec* es  $T = 48\text{KHz}$ , frecuencia

Figura A.10: *Espectro del filtro de salida.*

suficiente para muestrear las señales generadas, que están en el rango  $6KHz - 15KHz$ . Este periodo de muestreo es el que dicta el tamaño de los vectores, que contienen la señal de entrada y de salida.

La UCE genera un vector con los símbolos *On/Off* adecuados, para completar todos los campos que forman la trama. Cada símbolo se genera como una señal cosenoidal con frecuencia  $5KHz$ . Los puntos del vector, se envía a intervalos de tiempo dictados por la interrupción del *Codec*.

El receptor FH-SS incluye el módulo de supresión de interferencias introducido en el apartado 4.2. La recepción de la trama FH-SS enviada, comienza justo al concluir el envío de la trama OOK desde UCE hacia la URA. Sólo es posible que una unidad esclava direccionada haga uso del canal, ya que el protocolo de acceso al medio que se ha implementado es de tipo *master/slave* [79]. El esquema de la unidad UCE se muestra en la figura A.11.

### **Generación y recepción de las señales de comunicación**

La generación y la recepción de la señal de comunicación se realiza mediante el *Codec*, y se almacenan en memoria antes de ser procesadas. La señal de comunicación OOK generada por la UCE, se muestra en el tiempo y en frecuencia en las figuras A.12 y A.13.

En la recepción, la señal proveniente de la URA se filtra de forma analógica, en el rango  $3KHz - 25KHz$ , y se muestrea con el convertidor analógico digital. El vec-

Figura A.11: *Esquema de la tarjeta UCE, con filtros de entrada y salida que se detallan en la sección siguiente, el amplificador de salida y el acondicionador de entrada para recuperar la señal y el DSP .*

tor de puntos muestreados forma la entrada FH-SS, cuyos módulos se han explicado anteriormente. Los filtros de entrada y salida son básicamente similares a los de la URA.

### A.3 Estructura de la instalación piloto

El sector que ha servido de banco de pruebas contiene 450 parcelas, con una superficie total de 3244 hectáreas. Distribuidas por todo el sector y situados a pie de parcela, se encuentran 60 URA conectados a través de un único cable, que recorre toda la Comunidad. Para controlar toda la instalación el sistema se compone de una Unidad Central, basada en un ordenador tipo PC y una UCE. Junto a la UC, distribuidas por todo el sector y situados a pie de parcela, se encuentran las URAs conectadas a través de un único cable que recorre todo el sector.

En esta instalación no se han utilizado las unidades autónomas de control, ya que el sistema de abonado y filtrado no es muy complejo y no necesita de elementos propios para su control. El único elemento, que necesita una unidad expresamente para su control, es el banco de filtros de agua, que necesita ser limpiado periódicamente. Pero la unidad instalada ya contaba con su propia unidad de control, que funciona de forma autónoma.

A continuación, se especifican con más detalle las unidades que componen el sistema divididas en: elementos remotos distribuidos por toda el sector, y por otro lado los elementos de la unidad central situados en cabezal.

Figura A.12: *Representación en el tiempo de la señal enviada desde la UCE con modulación OOK y codificación Manchester.*

#### A.3.1 Elementos remotos

A pie de parcela se sitúan los *hidrantes*. Cada uno de ellos contiene una tarjeta URA, una toma de agua de la tubería y una conexión al cable de potencia. La toma de agua está controlada por una electroválvula con un solenoide tipo Latch. Cada hidrante contiene una única electroválvula pero varias salidas de agua, una para cada parcela, con un contador independiente. No se han instalado contadores de agua, por razones de coste, aunque la URA está preparado para realizar esta función. En la figura A.14 se muestra una fotografía de un hidrante, en dicha figura se puede distinguir la tarjeta URA, hasta la cual llega el cable de alimentación, también se muestra el solenoide conectado a las salidas de la URA, y la electroválvula junto con los contadores en la tubería de agua.

El número total de hidrantes para controlar todas las parcelas es 59, estando distribuidas tal como se representa en la figura A.15. En dicha figura también se representa la red de cableado <sup>2</sup> en que recorre toda la instalación junto con la tubería. Como se puede comprobar en la figura, este cableado está muy ramificado con una topología de malla, siendo la distancia desde un extremo del sector hasta el opuesto es de unos  $2.5Km$ , aunque la cantidad total de cable es superior debido a la topología. Las estimaciones realizadas indican que la cantidad total de cable conectada puede ser del orden de los  $10Km$ .

---

<sup>2</sup>El cable es de tipo N05 VV-F PVN del fabricante Pirelli

Figura A.13: *Representación en frecuencia de la señal enviada desde la UCE.*

### **A.3.2 Elementos del cabezal**

En el cabezal está instalada la Unidad Central, constituida por un ordenador de tipo PC, que controla toda la instalación mediante un software específico de control y gestión desarrollado dentro del proyecto.

#### **Software utilizado en las pruebas**

A continuación se dan las características más destacables del software utilizado en las pruebas. En primer lugar se ha utilizado el software PRIFEM UC, que ha sido desarrollado como parte de la arquitectura, dentro de los proyectos que han dado lugar al sistema.

El software PRIFEM UC es capaz de controlar toda la instalación de manera autónoma. En la pantalla principal se muestra el estado de cada uno de los hidrantes: abierto, cerrado o indeterminado, mediante un color. El programa permite modificar el estado de un hidrante de forma manual, sin más que pulsar con el ratón sobre el hidrante deseado. Esta acción produce el envío de la trama adecuada desde la UCC hasta la URA correspondiente. Esta pantalla principal es la mostrada en la figura A.17.

También se puede ver y modificar el estado de cada uno de los elementos situados en el cabezal, incluyendo los de abonado y la bomba de agua. Además de median-

Figura A.14: *Fotografía de un hidrante de la instalación prototipo.*

te funcionamiento manual, el software también es capaz de realizar riegos de forma autónoma, introduciendo en una base de datos interna todos los datos necesarios sobre el riego que incluyen las características comunes a todos los riegos: hidrantes a activar, hora de inicio, periodicidad y condiciones del abonado. El abonado incluye tiempo de pre-riego, cantidad de abonado y tipo de abonado. Las incidencias ocurridas durante el riego se almacenan en un fichero *.log*.

De especial utilidad en la validación de las comunicaciones ha sido un módulo del program PRIFEM UC, dedicado al test de las URAs, que almacena las estadísticas de las tramas erróneas recibidas. Este módulo tiene un interface gráfico, mostrado en la figura A.18. Con él se puede detectar la presencia de la UCE conectada al PC mediante algún puerto de comunicaciones serie (COM), configurable en el software. Este puerto puede ser del tipo RS232/RS485. El módulo permite automatizar el chequeo de las tramas erróneas que se han producido en el proceso de test. En el interface gráfico del modulo se indican los hidrantes a chequear, cada cuanto tiempo se realiza en chequeo y el resultado del proceso.

El error en el proceso de chequeo, puede producirse tanto en la recepción de la trama OOK en la URA, como en la recepción posterior de la trama contestada en la UCE. Para distinguir ambas situaciones, la URA en el proceso de test envía el número de trama correctas, que ha recibido desde que empezó el proceso.

Figura A.15: *Mapa de situación de los hidrantes y esquema del cableado de la instalación.*

Otros programas utilizados han sido el software para depuración, integrado dentro del paquete de software *VisualDSP*, desarrollado por *Analog Devices* y que permite la comunicación directa con la tarjeta *ADSP-21065L EZ-LAB*, la descargar a un fichero de la memoria de la tarjeta y la depuración del software. Este paquete ha permitido tomar muestras de ruido y tramas emitidas en la instalación mediante el convertidor analógico-digital incluido en la UCE, y su posterior descarga para su análisis mediante el software Matlab.

### A.3. Estructura de la instalación piloto

---

Figura A.16: *Unidad Central compuesta por el PC y la UCE.*

Figura A.17: *Imagen de la pantalla principal del programa PRIFEM UC.*

Figura A.18: *Imagen de la pantalla del modo de pruebas del programa Prifem UC.*

### A.3. Estructura de la instalación piloto

---

# Apéndice B

## Características básicas del procesador digital de señal

### B.1 El DSP *ADSP2065L*

El *ADSP2065L SHARC* es un DSP *Digital Signal Processor* de 32-bits diseñado, para aplicaciones de comunicaciones, audio e instrumentación industrial. Este dispositivo como la mayoría de procesadores DSP, presenta arquitectura Harvard y es capaz de realizar operaciones sobre datos en representación en punto flotante.

Las características más importantes son:

- Contiene 3 unidades funcionales, una ALU, un multiplicador y una unidad para realizar desplazamientos. Todas ellas pueden trabajar con datos de hasta 32-bits en coma fija y en formato 32/40-bits IEEE en punto flotante, con un rendimiento de pico de 180 MFLOPS, funcionando a *60Mhz*.
- Las operaciones se ejecutan en un ciclo de reloj, pudiendo realizarse operaciones independientes en la ALU y el multiplicador de forma simultánea.
- Banco con 32 registros internos de 40 bits, agrupados en dos conjuntos de 16, para conseguir un cambio rápido de contexto.
- El procesador está segmentado en tres pasos búsqueda, decodificación y ejecución.
- Dos unidades de generación de direcciones, que implementan *Buffers* circulares por hardware.

## B.1. El DSP *ADSP2065L*

---

- 544 Kbits de memoria SRAM de doble puerto, configurable en dos bancos, el banco0, normalmente usado para instrucciones, contiene 6Kx48 bits y el banco1, usado sólo para datos, 8Kx32 bits.
- 10 canales de DMA.
- 10 temporizadores programables.
- JTAG, puerto serie, y buses de I/O para conexión directa de
- Memoria Cache asociativa por conjuntos de dos entradas, con 16 conjuntos y capaz de almacenar 32 instrucciones. No todas las instrucciones se buscan en la cache, sólo aquellas que interfieran con un acceso de datos a la memoria de programa, producido por la instrucción inmediatamente anterior en el orden de ejecución. En este caso si hay un *miss* en el siguiente ciclo de reloj se busca la instrucción y se almacena en la cache.

La figura B.1 muestra la estructura interna del dispositivo, donde se puede apreciar la estructura Harvard interna, ya que tiene dos buses denominados, PM Y DM, para acceder a la memoria interna, al puerto de conexión externa y a los canales de DMA. El primero de ellos, el PM, está orientado a la búsqueda de instrucciones y el segundo de ellos, el DM, a la captura de los datos. La memoria cache integrada en el dispositivo permite que el bus PM quede liberado en el acceso a las intrucciones y puede ser usado para acceder también a datos. Con esta situación se consigue el máximo rendimiento del dispositivo, ya que se accede a la instrucción y a dos datos en un sólo ciclo, permitiendo la implementación eficiente de filtros digitales, mediante operaciones multiplicación-acumulación.

Figura B.1: *Estructura interna del DSP ADSP2065L.*

B.1. El DSP *ADSP2065L*

---

# Apéndice C

## Programa para Matlab

### C.1 Programa del simulador FH-SS, sobre Matlab

La herramienta utilizada en las simulaciones del sistema formado por el emisor y el receptor FH-SS ha sido Matlab. Son muchos los programas que se han realizado para comprobar cada uno de los casos propuestos en el capítulo de resultados 6, poner el listado de todos los programas sería multiplicar demasiado el tamaño de la tesis, por lo que sólo se va a incluir un ejemplo, que representa a todos los demás.

#### C.1.1 Fichero simuladorfh.m

El programa que se incluye muestra el caso de un emisor FH-SS, que emite una trama de bits sobre la cual se realiza la excisión con UWPT.

```
function simuladorfh()  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DEFINICIONES: de valores iniciales  
ef=[]; %Vector de errores  
ns=40; %N'umero de s'imbolos en una trama  
fm=50000;  
t=1/fm:1/fm:50/fm; %Vector de tiempo frecuencia de muestro  
u=1/fm:1/fm:(ns*50+48)/fm; %Vector de tiempo para una trama  
fp=1000; %Frecuencia de la portadora  
fd=1000; %Frecuencia de desplazamiento  
g=1; %Division del periodo al integrar  
in=1; %N'umero de puntos por periodo  
z=1/2; %Fase de la se~{n}al  
filtro=fir1(200,0.053); %Filtro paso-bajo  
umbral=0.004 %Umbral de energ'ia  
for h=0:9
```

## C.1. Programa del simulador FH-SS, sobre Matlab

---

```
    coseno(h+1,:)=sqrt(2/0.0002)*cos(2*pi*u*(fp+fd*h))*0.000001;
    seno(h+1,:)=sqrt(2/0.0002)*sin(2*pi*u*(fp+fd*h))*0.000001;
    %correladores
end
ins=length(t)/in;          %Indice del sumatorio final

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%INTERFERENCIA SENOIDAL:
veces=23;
veces2=1/10^(veces/10);
ins=length(t)/in; %VALORES DEL RUIDO GAUSSIANO:

for rep=1:1000    %Repetimos la simulaci'on para ns*rep s'imbolos

    %rep
    %z=rand(1)*2;
    ff=0; %rand(1)*50;    %interferencia CWT
    ruidosen=100*cos(2*pi*u*(1000+ff)+pi*z);
    bit=floor(rand(1,ns)*2);
        %Genera ns s'imbolos con valores entre [0:1]

    for inn=1:ns
        indd=(floor(rand(1,1)*5)*2);
        simbolo(inn)=(indd+bit(inn));
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%SE\~{N}AL TRANSMITIDA:
    senal_trans=[];
    error(rep)=0;
    correcto(rep)=0;
    for h=1:ns    %S'imbolo transmitido
        senal_trans=[senal_trans sqrt((2)/0.0002)*cos(2*pi*t*(fp+fd*simbolo(h)))]];
    end

    inicio=length(senal_trans);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%RUIDO:
    media=0;
    desvia=veces2*150e4;
    ruido=normrnd(media,sqrt(desvia),1,length(u));
        %Genera ruido aleatorio de distribuci'on gaussiana
        %de media 0 y varianza desvia

    senal_recep=ruido+ruidosen;
    senal_recep(1:2000)=senal_recep(1:2000)+senal_trans;
        %Se\~{n}al en el receptor suma de se\~{n}al enviada y ruido

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%RECEPTOR:
    for h=0:9
        rc=coseno(h+1,:).*senal_recep;
        rs=seno(h+1,:).*senal_recep;
        rcb=conv(filtro,rc);
        rsb=conv(filtro,rs);
```

```

rcbi=rcb(100:length(rcb)-100);
rsbi=rsb(100:length(rsb)-100);
%retrasos del filtro FIR
rb=rcbi+sqrt(-1)*rsbi;

if (rcbi*rcbi+rsbi*rsbi>umbral)
    xss=algoritmo(rcb);
    xst=real(xss);
    yst=imag(xss);
else
    xst=rcbi;
    yst=rsbi;
end

for i=1:round(inicio/ins)
    x(i)=sum(xst(i*ins-(ins-1):i*ins));
    y(i)=sum(yst(i*ins-(ins-1):i*ins));
end

    %Valor del s'imbolo a la entrada del detector
    %para cada valor de h
for i=1:ns
    s(h+1,i)=sqrt((x(i))^2+(y(i))^2);
end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%COMPROBACION:se compara el s'imbolo transmitido

for i=1:ns
    [m,indice]=max([s(simbolo(i)-bit(i)+1,i) s(simbolo(i)-bit(i)+2,i)]);
    if(indice==bit(i)+1) correcto(rep)=correcto(rep)+1;
    else error(rep)=error(rep)+1;
    end
end
ef(g)=sum(error)/((40-2)*rep)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ERROR:
ef(g)=sum(error)/((ns)*rep);
load s204810;
ef2=[ef ef2];

save s204810 ef2;
g=g+1;

\normalsize

```

## C.1.2 Algoritmo de filtrado algoritmo.C

```

//#pragma hdrstop
//#include <condefs.h>

```

## C.1. Programa del simulador FH-SS, sobre Matlab

---

```
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include
"mex.h"
//-----
//#pragma argsused
#define tam 2048//4096 //8192//1024//2048 //8192 //32787 //
#define pi 3.1416 #define sq2 0.7071 void best2(double
*xssr,double *xssi,double *rcbi,double *rsbi,double pasos) {

    int i,ii,k,j,temp,nn,tamf=4;
    int ret1[]={257,129,65,33,17,9,5,3};//2049,1025,513,257,129,
    int reth[]={129,65,33,17,9,5,3,2};//1025,513,257,129,65,
    double indicef,maxvecen,vecen;
    double paso[11],final[11],vec[11],vecdes[11];
    double pasof,despl_anter,despldef,e1,eh,veces_ener,des;
    double xss2r[tam],xss2i[tam],xss3r[tam],xss3i[tam];
    double lpr[tam],hpr[tam],lprd[tam],hprd[tam];
    double ordenr[11][tam],ordeni[11][tam],
    double lpi[tam],hpi[tam],lpid[tam],hpid[tam];
    double ha[4]={-0.4830, 0.8365, -0.2241, -0.1294};
    double la[4]={-0.1294, 0.2241, 0.8265, 0.4830};
    double lb[4]={0.4830, 0.8365, 0.2241, -0.1294};
    double hb[4]={-0.1294, -0.2241, 0.8365, -0.483};

    nn=8;
    displ_anter=0;
    for (ii=0;ii<nn;ii++){
        if(!(ii)){
            paso[0]=pasos;
            final[0]=(tam/2)/256;//(pow(2,nn-ii))+1;
        }else{
            paso[ii]=(tam/2)/(pow(2,nn-ii));
            //final[ii-1]=(tam/2)*(1/(ldexp(1,nn-ii)))+1;
            final[ii]=(tam/2)/(pow(2,nn-ii))+1;
        }
    }
    //printf("%i",pasos);
    for(j=0;j<tam;j++){
        xss2r[j]=rcbi[j];
        xss2i[j]=rsbi[j];
        //xss3r[j]=rcbi[j];
        //xss3i[j]=rsbi[j];
    }
    pasof=7;
    for (ii=0;ii<nn;ii++){

        for (veces_ener=0;veces_ener<=final[ii];veces_ener+=paso[ii]){

            des=(veces_ener)/tam;
```

```

e1=0;eh=0;
for(k=0;k<tam;k++){
    xssr[k]=xss2r[k]*cos(2*k*pi*des)-xss2i[k]*sin(2*k*pi*des);
    xssi[k]=xss2r[k]*sin(2*k*pi*des)+xss2i[k]*cos(2*k*pi*des);
}

for(k=0;k<tam;k++){
    lpr[k]=0;lpi[k]=0;hpr[k]=0;hpi[k]=0;indicef=0;
    for(j=k;j>k-(1<<(int)pasof)*tamf;j=j-(1<<(int)pasof)){
        if(j<0) temp=tam+(j);
        else temp=j;

        lpr[k]+=xssr[temp]*la[indicef];
        lpi[k]+=xssi[temp]*la[indicef];
        hpr[k]+=xssr[temp]*ha[indicef];
        hpi[k]+=xssi[temp]*ha[indicef++];
    }
}
for(j=0;j<=tam-retl[ii];j++){
    lprd[j]=sq2*lpr[j+retl[ii]-1];
    lpid[j]=sq2*lpi[j+retl[ii]-1];
}
for(j=1;j<retl[ii];j++){
    lprd[j+tam-retl[ii]]=sq2*lpr[j-1];
    lpid[j+tam-retl[ii]]=sq2*lpi[j-1];
}
for(j=0;j<=tam-reth[ii];j++){
    hprd[j]=sq2*hpr[j+reth[ii]-1];
    hpid[j]=sq2*hpi[j+reth[ii]-1];
}
for(j=1;j<reth[ii];j++){
    hprd[j+tam-reth[ii]]=sq2*hpr[j-1];
    hpid[j+tam-reth[ii]]=sq2*hpi[j-1];
}
e1=0;eh=0;
for(j=0;j<tam;j++){
    e1+=(lprd[j]*lprd[j])+(lpid[j]*lpid[j]);
    eh+=(hprd[j]*hprd[j])+(hpid[j]*hpid[j]);
}
vecen=fabs(e1-eh);
// printf("\nener %lf",vecen);
if(vecen_ener==0){

    maxvecen=vecen;despldef=des;
    if(e1>eh){
        for(j=0;j<tam;j++){
            xss3r[j]=lprd[j];
            xss3i[j]=lpid[j];
            ordenr[ii][j]=hprd[j];
            ordeni[ii][j]=hpid[j];
        }
    }
}

```

## C.1. Programa del simulador FH-SS, sobre Matlab

---

```
        }
        vec[ii]=0;
    }
    else{
        for(j=0;j<tam;j++){
            xss3r[j]=hprd[j];
            xss3i[j]=hpid[j];
            ordenr[ii][j]=lprd[j];
            ordeni[ii][j]=lpid[j];
        }
        vec[ii]=1;
    }
}
else
    if(maxvecen<vecen){
        despldef=des;maxvecen=vecen;
        if(e1>eh){
            for(j=0;j<tam;j++){
                xss3r[j]=lprd[j];
                xss3i[j]=lpid[j];
                ordenr[ii][j]=hprd[j];
                ordeni[ii][j]=hpid[j];
            }
            vec[ii]=0;
        }
        else{
            for(j=0;j<tam;j++){
                xss3r[j]=hprd[j];
                xss3i[j]=hpid[j];
                ordenr[ii][j]=lprd[j];
                ordeni[ii][j]=lpid[j];
            }
            vec[ii]=1;
        }
    }
}
for(j=0;j<tam;j++){
    xss2r[j]=xss3r[j];
    xss2i[j]=xss3i[j];
}
despl_anter=despl_anter+despldef;
vecdes[ii]=despldef;
// printf(" vec:%lf",vecdes[ii]);
pasof--;
}

for(j=0;j<tam;j++){
    xssr[j]=0;
    xssi[j]=0;
}

pasof++;
```

## Apéndice C. Programa para Matlab

---

```

for (ii=nn-1;ii>=0;ii--){
    if(vec[ii])
        for(k=0;k<tam;k++){
            lpr[k]=0;lpi[k]=0;hpr[k]=0;hpi[k]=0;indicef=0;
            for(j=k;j>k-(1<<(int)pasof)*tamf;j-=(1<<(int)pasof)){
                if(j<0) temp=tam+(j);
                else temp=j;
                lpr[k]+=ordenr[ii][temp]*lb[indicef];
                lpi[k]+=ordeni[ii][temp]*lb[indicef];
                hpr[k]+=xssr[temp]*hb[indicef];
                hpi[k]+=xssi[temp]*hb[indicef++];
            }
        }
    else
        for(k=0;k<tam;k++){
            lpr[k]=0;lpi[k]=0;hpr[k]=0;hpi[k]=0;indicef=0;
            for(j=k;j>k-(1<<(int)pasof)*tamf;j-=(1<<(int)pasof)){
                if(j<0) temp=tam+(j);
                else temp=j;
                lpr[k]+=xssr[temp]*lb[indicef];
                lpi[k]+=xssi[temp]*lb[indicef];
                hpr[k]+=ordenr[ii][temp]*hb[indicef];
                hpi[k]+=ordeni[ii][temp]*hb[indicef++];
            }
        }
    for(j=0;j<=tam-reth[ii];j++){
        lprd[j]=sq2*lpr[j+reth[ii]-1];
        lpid[j]=sq2*lpi[j+reth[ii]-1];
    }
    for(j=1;j<reth[ii];j++){
        lprd[j+tam-reth[ii]]=sq2*lpr[j-1];
        lpid[j+tam-reth[ii]]=sq2*lpi[j-1];
    }
    for(j=0;j<=tam-retl[ii];j++){
        hprd[j]=sq2*hpr[j+retl[ii]-1];
        hpid[j]=sq2*hpi[j+retl[ii]-1];
    }
    for(j=1;j<retl[ii];j++){
        hprd[j+tam-retl[ii]]=sq2*hpr[j-1];
        hpid[j+tam-retl[ii]]=sq2*hpi[j-1];
    }
    for(k=0;k<tam;k++){
        xss2r[k]=lprd[k]+hprd[k];
        xss2i[k]=lpid[k]+hpid[k];
    }
    for(k=0;k<tam;k++){
        xssr[k]=xss2r[k]*cos(-2*k*pi*vecdes[ii])
            -xss2i[k]*sin(-2*k*pi*vecdes[ii]);
        xssi[k]=xss2r[k]*sin(-2*k*pi*vecdes[ii])
            +xss2i[k]*cos(-2*k*pi*vecdes[ii]);
    }
    pasof++;
}

```

## C.1. Programa del simulador FH-SS, sobre Matlab

---

```
    }  
}  
  
#define cols 1 void mexFunction( int nlhs, mxArray *plhs[], int  
nrhs, const mxArray *prhs[] ) {  
    double pasos,*xr, *xi, *yr, *yi;  
    int mx,nn;  
    mx=2048;//8192;//1024;//2048;//8192;//32787; //  
    nn=11;  
  
    //getchar();  
    //printf("ya");  
    xr=mxGetPr(prhs[0]);  
    xi=mxGetPi(prhs[0]);  
    /*yr=mxCalloc(mx,sizeof(double));  
    yi=mxCalloc(mx,sizeof(double));*/  
    plhs[0]=mxCreateDoubleMatrix(mx,cols,mxCOMPLEX);  
    yr=mxGetPr(plhs[0]);  
    yi=mxGetPi(plhs[0]);  
    pasos=*xr;  
    *xr=xr[1];  
    //printf("%f",*xr);  
    //printf("%f",xr[1]);  
    //printf("%i",pasos);  
  
    best2(yr,yi,xr,xi,pasos);  
    //printf("ya");
```

# Apéndice D

## Programa para el DSP ADSP21065L

### D.1 Funciones en ensamblador para el cálculo de la estructura *Lattice*

En este anexo se muestran las dos subrutinas de cálculo de la estructura *Lattice* para el DSP ADSP21065L. Estas estructuras hacen uso de los registros secundarios del DSP que permiten cambiar el contexto de una función de forma rápida, puesto que el resto del programa, escrito en lenguaje 'C', no hace uso de este banco de registros, por lo tanto no es necesario apilar ni desapilar su contenido.

#### D.1.1 Análisis

```
#include <signal.h> #include <def21065l.h> #include <signal.h>
#include <asm_sprt.h> #include <macros.h>

#define MAX_RBUFFER      8192/*2048*/

.extern count_start; .extern count_end;

.EXTERN    _b_z1; .EXTERN    _b_z2; .EXTERN    _z1; .EXTERN
_z2; .EXTERN    _punt; .EXTERN    _sen_x; .EXTERN    _sen_y;
.EXTERN    _sen_hgx; .EXTERN    _sen_hgy; .EXTERN    _difex;
.EXTERN    _difey; .EXTERN    _p; .EXTERN    _pp; .segment/pm
seg_pmco;

.global _latticex; _latticex:
```

## D.1. Funciones en ensamblador para el cálculo de la estructura *Lattice*

---

```
leaf_entry;

    BIT CLR MODE1 0x00001000; /*bit 12*/
    BIT SET MODE1 0x000004F8; /*bit 3 4 5 6 7 10*/
    nop;

    b0=_b_z1;
    b8=_b_z2;

    r0=dm(_z1);
    l0=r0;
    r3=r0;

    r0=dm(_z2);
    l8=r0;
    r3=r3+r0;

    r2=MAX_RBUFFER;
    l2=r2;
    r5=MAX_RBUFFER;
    r5= lshift r5 by 1;
    l3=r5;

    r1=dm(_p);/*_sen_x/y*/
    b2=r1;
    r1=r1+r2;
    r1=r1-r3;
    /*r1=r1-1;*/
    i2=r1;

    r1=dm(_pp);/*_sen_hgx/y*/
    b3=r1;
    r7=r3+r3;
    r7=r5-r7;
    r7=r1+r7;

    i3=r7;

    r4=MAX_RBUFFER;
    r15=r3+r4;

    m6=1;
    m14=1;
    m4=0;
    m12=0;

    f0=-0.1294/1.4142;
    f1=1.7321;
    f2=3.7326;
```

```

                f11=f10+f14,  f4=dm(i2,m6),  f5=pm(i8,m12);
f6=f0*f4,  f7=f10+f14,  f8=dm(i0,m4);
f12=f2*f7,  f3=f11+f15,  f7=dm(i0,m4);
f12=f1*f6,  f3=f9-f12,  dm(i3,m6)=f3,  f9=pm(i8,m12);
f10=f0*f4,  f13=f8-f12,  dm(i3,m6)=f3;
f14=f1*f7,  dm(i0,m6)=f6,  pm(i8,m14)=f13;
lcntr=r15, do dotx until lce;

                f15=f2*f5,  f11=f10+f14,  f4=dm(i2,m6),  f5=pm(i8,m12);

                f6=f0*f4,  f7=f10+f14,  f8=dm(i0,m4);
f12=f2*f7,  f3=f11+f15,  f7=dm(i0,m4);
f12=f1*f6,  f3=f9-f12,  dm(i3,m6)=f3,  f9=pm(i8,m12);
f10=f0*f4,  f13=f8-f12,  dm(i3,m6)=f3;
dotx:  f14=f1*f7,  dm(i0,m6)=f6,
pm(i8,m14)=f13;

                BIT CLR MODE1 0x000004F8; /*bit 3*/
                BIT SET MODE1 0x00001000; /*bit 12*/
                nop;
leaf_exit;

.endseg;

```

## D.1.2 Síntesis

```

#include <signal.h> #include <def21065l.h> #include <signal.h>
#include <asm_sprt.h> #include <macros.h>

#define MAX_RBUFFER      8192/*2048*/

.EXTERN  _b_z1; .EXTERN  _b_z2; .EXTERN  _z1; .EXTERN
_z2; .EXTERN  _punt; .EXTERN  _sen_x; .EXTERN  _sen_y;
.EXTERN  _sen_hgx; .EXTERN  _sen_hgy; .EXTERN  _difex;
.EXTERN  _difey; .EXTERN  _p; .EXTERN  _pp; .segment/pm
seg_pmco;

.global _latinvx; _latinvx:

leaf_entry;

```

## D.1. Funciones en ensamblador para el cálculo de la estructura *Lattice*

---

```
BIT CLR MODE1 0x00001000; /*bit 12*/
BIT SET MODE1 0x000004F8; /*bit 3 4 5 6 7 10*/
nop;

b0=_b_z1;
b8=_b_z2;

r0=dm(_z1);
l0=r0;
r3=r0;

r0=dm(_z2);
l8=r0;
r3=r3+r0;

r2=MAX_RBUFFER;
l2=r2;
r5=MAX_RBUFFER;
r5= lshift r5 by 1;

l2=r2;
l3=r5;
l4=r5;

r1=dm(_p);/*_sen_x/y*/
b2=r1;
r1=r1+r2;
r1=r1-r3;

i2=r1;

r1=dm(_pp);/*_sen_hgx/y*/
b3=r1;
b4=r1;
r7=r3+r3;
r7=r5-r7;
r7=r1+r7;

i3=r7; /*primero se lee g*/
r7=r7+1;
i4=r7;

r7=r2+r3;
r10=0;
r11=r7+r10;
m6=1;
m14=1;
m4=0;
m12=0;
m5=2;
```

## Apéndice D. Programa para el DSP ADSP21065L

---

```
f1=-0.1294/1.4142;
f2=1.7321;
f3=3.7326;

f8=f1*f5,          f4=dm(i3,m4),  f5=pm(i8,m12);
f8=f3*f4,  f7=f8+f12,  f13=dm(i4,m4),  f10=pm(i8,m12);
f12=f2*f5,  f4=f8+f13,  f6=dm(i4,m5);
f15=f3*f6,  f8=f8+f12,  f9=dm(i3,m5);
f14=f2*f4,  f0=f9-f15,  f5=dm(i0,m4);
f12=f1*f5,  f0=f10-f14, dm(i2,m6)=f7,  pm(i8,m14)=f0;
              f5=f8+f13,  dm(i0,m6)=f0;
lcnt=r11, do dot_loop until lce;
f8=f1*f5,          f4=dm(i3,m4),  f5=pm(i8,m12);
f8=f3*f4, f7=f8+f12,  f13=dm(i4,m4),  f10=pm(i8,m12);
f12=f2*f5,  f4=f8+f13,  f6=dm(i4,m5);
f15=f3*f6,  f8=f8+f12,  f9=dm(i3,m5);
f14=f2*f4,  f0=f9-f15,  f5=dm(i0,m4);
f12=f1*f5,  f0=f10-f14, dm(i2,m6)=f7,  pm(i8,m14)=f0;
dot_loop:      f5=f8+f13,  dm(i0,m6)=f0;

BIT CLR MODE1 0x000004F8; /*bit 3*/
BIT SET MODE1 0x00001000; /*bit 12*/
nop;
leaf_exit;

.endseg;
```



**Parte VII**  
**Bibliografía**



# Bibliografía

- [1] Statistic toolbox, for use with matlab. <http://www.mathworks.com/>, 2002.
- [2] A.N. Akansu, M.V. Tazebay, M.J. Medley, and P.K. Das. Wavelet and subband transforms: Fundamentals and communications applications. *IEEE Communications Magazine*, pages 104–115, Dec. 1997.
- [3] H. Bölcskei, F. Hlawatsch, and G. Feichtinger. Frame-theoretic analysis and desing of oversampled filter-banks. *In Proc. IEEE ISCAS97, Hong-Kong*, pages 409–412, June. 1997.
- [4] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Commun.*, COM-31:532–540, 1983.
- [5] C. Chakrabarti and M. Viswanath. Efficient realizations of the discrete and continuous wavelet transform: From single chip implementations to mappings on simd array computers. *IEEE Trans. On Signal Processing*, 43:759–771, Mar. 1995.
- [6] M. Chan and R. Donaldson. Amplitude width, and interarrival distributions for noise impulses on intrabuilding power line communication networks. *IEEE Transactions on Electromagnetic Compatibility*, 31:320–323, Aug. 1989.
- [7] K. Cheun and E. Stark. Performance of fhss systems employing carrier jitter against one-dimensional tone jamming. *IEEE Trans. on Commun.*, 43:2622–2629, Oct. 1995.
- [8] K. Cheun and E. Stark. Probability of error in frequency-hop spread-spectrum multiple-access communication systems with non-coherent signaling. *IEEE Trans. on Commun.*, 39:1400–1410, Spt. 1991.
- [9] R. Coifman and D.L. Donoho. Translation-invariant de-noising. *Wavelets and Statistics, Springer Lecture Notes. Springer-Verlag*, 103:125–150, 1995.
- [10] R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Trans. on Information Theory*, 38:713–718, 1992.
- [11] Z. Cvetković and M. Vetterli. Oversampled filter banks. *IEEE Trans. on Signal Processing*, 46:1245–1255, May. 1998.

- [12] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.
- [13] I. Daubechies. The wavelet transform, time-freq. localization and signal analysis. *IEEE Trans. Information Theory*, 36:961–1005, Set. 1990.
- [14] S. Davidovici and E. Kanterakis. Narrow-band interference rejection using real-time fourier transforms. *IEEE Trans. on Commun.*, COM-37:713–722, July. 1989.
- [15] S. Davidovici and E. Kanterakis. Radiometric detection of direct-sequence spread-spectrum signals using interference excision. *IEEE Journal on Select. Areas in Commun.*, 7:576–588, May. 1989.
- [16] K. Defly, X. Wang, G. Wu, and M. Lecours. Synchronization in fh-mfsk spread spectrum systems. *IEEE 38th Vehicular Technology Conference.*, pages 385–389, 1988.
- [17] T.C. Denk and K. Parhi. Vlsi architectures for lattice structure based orthonormal discrete wavelet transform. *IEEE Trans. On Circuits and SystemsII: Analog and digital signal processing*, 44:129–132, June. 1997.
- [18] K. M. Dostert. Frequency-hopping spread-spectrum modulation for digital communications over electrical power lines. *IEEE Journal on Select. Areas in Commun.*, 8:700–710, May 1990.
- [19] T. Edwards. Discrete wavelet transforms: Theory and application (draft #2). Technical report, Stanford University, June 1992.
- [20] G. Evans. *The CEBus Communications Standard, Part 1*. Communication Systems Design, Feb. 1996.
- [21] G. Evans. *The CEBus Communications Standard, Part 2*. Communication Systems Design, Mar. 1996.
- [22] J. Fridman and E. Manolakos. Distributed memory and control vlsi architectures for the 1-d discrete wavelet transform. *VLSI signal processing*, pages 388–397, 1994.
- [23] L.M. Garth and H.V. Poor. Narrowband interference suppression in impulsive channels. *IEEE Trans. on Aeros. and Elect. Systems*, 28:15–33, Jan. 1992.
- [24] S. Glisic and B. Vucetic. *Spread-Spectrum CDMA systems for Wireless Communications*. Artech house, 1997.
- [25] G. Goossens, J. Praet, D. Lanneer, and W. Geurts. Embedded software in real-time signal processing systems: Design technologies. *Proceedings of the IEEE*, 85:436–454, March 1997.

- [26] J.R Hampton. An edge detection technique for fine acquisition in frequency hopping radios. *Military Communications Conference, MILCOM '93.*, pages 198–202, 1993.
- [27] S. Haykin. *Adaptive Filter theory*. Prentice Hall, 1991.
- [28] R.A. Iltis and L.B. Milstein. Interference rejection in fhh systems using least squares estimation techniques. *IEEE Trans. on Commun.*, COM-38:2174–2183, Dec. 1990.
- [29] R.A. Iltis and L.B. Milstein. An approximate statistical analysis of the widrow lms algorithm with application to narrow-band interference rejection. *IEEE Trans. on Commun.*, COM-33:121–130, Feb. 1985.
- [30] R. Jain. *The Art of Computer System Performance Analysis*. Wiley, 1991.
- [31] W.W. Jones and K.R. Jones. Narrowband interference suppression using filter bank analysis/synthesis techniques. *Proc. MILCOM'92*, pages 898–902, Oct. 1992.
- [32] C.M. Keller and M.B. Pursley. Clipper diversity combining for channels with partial-band interference- partii: Ratio-statistic combining. *IEEE Trans. on Commun.*, COM-37:145–151, Feb. 1989.
- [33] W. D. Kelton. Simualtion analysis. *Proceedings of the 1983 winter simulation conference*, 1:159–168, 1983.
- [34] G. Knowles. Vlsi architecture for the discrete wavelet transform. *Electronic letters*, 26:1184–1185, July. 1990.
- [35] B. S. Krongold, K. Ramchandran, and D.L. Jones. Frequency-shift-invariant orthonormal wavelet packet representations. *IEEE Trans. on Signal Processing*, 47:2579–2582, 1999.
- [36] W. Lawrenz. *CAN System Engineering*. Springer-Verlag, 1997.
- [37] J.S. Lee, R.H. French, and L.E. Miller. Probability of analyzes of a bfsk frequency-hopping system with diversity under partial-band jamming interference- partii: Performance of square-law linear combining soft decision receiver. *IEEE Trans. on Commun.*, COM-32:645–653, June 1984.
- [38] J.S. Lee, L.E. Miller, and Y.K. Kim. Probability of analyzes of a bfsk frequency-hopping system with diversity under partial-band jamming interference- partii: Performance of square-law non-linear combining soft decision receiver. *IEEE Trans. on Commun.*, COM-32:1243–1250, Dec. 1984.
- [39] Rainer Leupers. Code generation for embedded processors. *IEEE ISSS Conference 2000, Madrid*, 2000.

- [40] C. Li, G. Hu, and M. Liu. Narrow-band interference excision in spread-spectrum systems using self-orthogonalizing transform-domain adaptative filters. *IEEE Journal on Select. Areas in Commun.*, 18:403–406, March 2000.
- [41] J. Liang and T.W. Parks. A translation-invariant wavelet representation algorithm with applications. *IEEE Trans. on Signal Processing*, 44:225–233, Feb. 1996.
- [42] S. Liao, S. Devadas, K. Keutzer, S. Tjiang, and A. Wang. Code optimization techniques for embedded dsp microprocessors. *32nd ACM/IEEE Design Automation Conference*, 1995.
- [43] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 11:674–693, July 1989.
- [44] S. Mallat and W.L.Hwang. Singularity detection and processing with wavelets. *IEEE Trans. Inform. Theory*, 38:617–643, March 1992.
- [45] S.G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Pr, 1998.
- [46] M.J. Medley and Ali.Ñ. Akansu. *Wavelet, Subband, and Block Transforms in Communications and Multimedia*. Kluwer International Series in Engineering and Computer Science, No 504, 1999.
- [47] M.J. Medley, G.J. Saulnier, and P.K. Das. Applications of the wavelet transform in spread-spectrum communications systems. *Proc. Thrid NJIT Symp. Applicat. Subbands Wavelets*, Mar. 1994.
- [48] M.J. Medley, G.J. Saulnier, and P.K. Das. Narrow-band interference excision in spread-spectrum systems using lapped transforms. *IEEE Trans. on Commun.*, COM-45:1444–1455, Nov. 1997.
- [49] L.E. Miller and P.K. Das. An analysis of a real-time transform domain filtering digital communications system- parti: Narrow-band interference rejection. *IEEE Trans. on Commun.*, COM-28:816–824, June 1980.
- [50] L.E. Miller, J.S. Lee, and A.P. Kadrichu. Probability of analyzes of a bfsk frequency-hopping system with diversity under partial-band jamming interference- partiii: Performance of square-law self-normalizing soft decision receiver. *IEEE Trans. on Commun.*, COM-34:669–675, July 1986.
- [51] L.B. Milstein. Interference rejection techniques in spread spectrum communications. *Proce. of the IEEE*, 76:657–671, June 1988.
- [52] G.Ñason and B. W. Silverman. The stationary wavelet transform and some statistical applications. *Wavelets and Statistics, Springer Lecture Notes. Springer-Verlag*, 103:281–299, 1995.

- [53] K.K. Parhi and T. Nishitani. Vlsi architectures for the discrete wavelet transform. *IEEE Trans. VLSI Syst.*, 1:191–202, June. 1993.
- [54] K. Pawlikowski. Steady-state simulation of queueing processes: A survey of problems and solutions. *ACM computing surveys*, 22:123–170, Jan. 1990.
- [55] K. Pawlikowski, H.J. Jeong, and J. R. Lee. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, pages 159–168, Jan. 2002.
- [56] A. Pegatoquet, E. Gresset, M. Auguin, and L. Bianco. Rapid development of optimized dsp code from a high level description through software estimations. *36th ACM/IEEE Design Automation Conference*, 1999.
- [57] J.J. Pérez, S. Felici, and J. Costa. A specific communication study for wide irrigation systems with integrated power transmission. *Proceedings IEEE ETFA '99, Barcelona*, 1999.
- [58] J.C. Pesquet, H. Krim, and H. Carfantan. Time-invariant orthonormal wavelet representations. *IEEE Trans. on Signal Processing*, 44:1964–1970, Aug. 1996.
- [59] J.G. Proakis. *Digital communications*. McGraw-Hill, 1995.
- [60] J.G. Proakis. Interference suppression in spread spectrum systems. *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications Proceedings*, 1:259–266, 1996.
- [61] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice Hall, 1995.
- [62] D. Radford. Spread-spectrum data leap through ac power wiring. *IEEE Spectrum*, pages 48–53, Nov. 1996.
- [63] A. Ranheim. Narrowband interference rejection in direct-sequence spread-spectrum system using time-frequency decomposition. *IEEE Proc. Commun.*, 142:393–400, Dec. 1995.
- [64] R. Rifkin. Comments on narrow-band interference rejection using real-time fourier transforms. *IEEE Trans. on Commun.*, COM-39:1292–1294, Sept. 1991.
- [65] O. Rioul and M. Vetterli. Wavelet and signal processing. *IEEE Signal Processing Magazine*, 1:14–37, Oct. 1991.
- [66] G.J. Saulnier. Supression of narrow-band jammers in spread-spectrum systems receiver using transform-domain adaptive filtering. *IEEE Journal on Select. Areas in Commun.*, 10:742–749, May 1992.

- [67] G. Schickhuber and O. McCarthy. Control using power lines - a european view. *IEE: Computing and Control Engineering Journal*, pages 180–184, Aug. 1997.
- [68] G. Schickhuber and O. McCarthy. Distributed fieldbus and control network systems. *IEE: Computing and Control Engineering Journal*, pages 21–32, Feb. 1997.
- [69] D. Schilling, L. Milstein, R. Pichholtz, M. Kullback, and F. Miller. Spread-spectrum for commercial communications. *IEEE Communications Magazine*, pages 66–79, Apr. 1991.
- [70] K.S. Shanmugan. Simulation and implementations tools for signal processing and communications systems. *IEEE Communications Magazine*, pages 36–40, July 1994.
- [71] M. Shensa. The discrete wavelet transform: Wedding the trous and mallat algorithms. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 11:674–693, July. 1989.
- [72] M.K. Simon. *Spread-Spectrum Communications Handbook*. McGraw-Hill, 1994.
- [73] S. Simon, P. Rieder, and J.A. Nossek. Cordic-based architectures for the efficient implementation of discrete wavelet transforms. *ISCAS '96*, 4:77–80, 1996.
- [74] S. Simon, P. Rieder, and J.A. Nossek. Efficient vlsi suited architectures for discrete wavelet transforms. *Workshop on VLSI Signal Processing, IX*, pages 388–397, 1996.
- [75] T. Sinnwell, S. Sinnwell, and K. Becker. Efficient implementation of the discrete wavelet transform on digital signal processors. *Applied Signal Processing*, 5 Issue 2:96–110, 1998.
- [76] K. Smith and L. Ran. Pwm dirves: Voltage-type harmonic sources in power systems. *IEE Proc-Gener.Transm. Distrib*, 145:293–299, May 1998.
- [77] M. Smith. Code optimization techniques for dsp applications. *9th IEEE DSP Workshop, Hunt, Texas*, Oct. 2000.
- [78] G. Strang and T.Ñguyen. *Wavelets and Filters Banks*. Wellesley-Cambridge Press, 1996.
- [79] A. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.
- [80] M.V. Tazebay and A.N. Akansu. Adaptive subband transforms in time-frequecy excisers for dsss communications systems. *IEEE Trans. on Signal Processing*, 43:2776–2781, Nov. 1995.

- [81] K.C. Teh, A.C. Kot, and K.H. Li. Performance analysis of an ffh/bfsk linear-combining receiver against multitone jamming. *IEEE Commun. Lett.*, 2:205–207, Aug. 1998.
- [82] K.C. Teh, A.C. Kot, and K.H. Li. Performance analysis of an ffh/bfsk self-normalising receiver under multitone jamming. *IEE Proceedings Communications*, 145:431–436, Dec. 1998.
- [83] K.C. Teh, A.C. Kot, and K.H. Li. Error probabilities and performance comparisons of ffh/bfsk receivers with multitone jamming and awgn. *IEEE International Conference on Acoustics, Speech, and Signal Processing, Phoenix, USA*, 5:2547–2550, Mar. 1999.
- [84] K.C. Teh, A.C. Kot, and K.H. Li. Partial-band jammer suppression in ffh spread-spectrum system using fft. *IEEE Trans. on Vehicular Tech.*, 48:478–486, March 1999.
- [85] P.P. Vaidyanathan and P.Q. Hoang. Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction qmf banks. *IEEE Trans. On Acoustics Speech and Signal Processing*, 36:81–93, Janu. 1988.
- [86] M. Vetterli and C. Herley. Wavelets and filter banks: Theory and design. *IEEE Trans. on Signal Processing*, 40:2207–2232, Setp. 1992.
- [87] R. Vijayan and H.V. Poor. Nonlinear techniques for interference supression in spread-spectrum systems. *IEEE Trans. on Commun.*, COM-38:1060–1065, July. 1990.
- [88] M. Vishwanath, R. M. Owens, and M.J. Irwin. Vlis architectures for the discrete wavelet transform. *IEEE Trans. On Circuits and SystemsII: Analog and digital signal processing*, 42:305–316, May. 1995.
- [89] A.T. Walden and A.C. Cristan. The phase-corrected undecimated discrete wavelet packet transform and the recurrence og high latitude interplanetary shock waves. *Technical Report TR-97-03*, Feb. 1997.
- [90] M.V. Wickerhauser. Inria lectures on wavelet packet algorithms. *In odelettes et Paquets dÓndeletes, Roquencourt, France*, pages 31–99, June. 17-21,1991.
- [91] R. Yacamini. Power system harmonics. *Power Engineering Journal*, pages 193–198, Aug. 1994.