



Trabajo de Grado

Algoritmo para el Procesamiento de Señales usando Filtrado por Octavas



Autores

Juan Manuel García
José Roberto Quentrequeo
Héctor Ismael Otero

Director

Ing. Oscar N. Bria

Universidad Nacional de La Plata
Facultad de Ciencias Exactas
La Plata - Argentina

UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMÁTICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar	 DIF-01927
TES 96/3 DIF-01927 SALA	



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

A la memoria de mi viejo. A mi vieja, familia y a Karen.
J.M.G.

A mis padres por su esfuerzo y abnegación. A Laly.
J.R.Q.

A mis padres, mi esposa y mis hijos.
H.I.O.



Agradecimientos

Nuestro más sincero agradecimiento al Ing. Oscar N. Bria y su esposa por el tiempo destinado a la dirección de este trabajo.



PREFACIO

Dado que las tendencias actuales de codificación de señales de audio utilizan un esquema mixto de codificación por análisis multiresolución, preferentemente codificación en sub-bandas, complementado con alguna otra técnica de codificación en cada banda (ATC, DPCM, etc.) y dado el creciente interés de los investigadores en utilizar la Transformada Wavelet, que también realiza un análisis multiresolucional y que representa una alternativa al análisis de Fourier, nos interesamos en avanzar sobre este tema en particular y experimentar utilizando esta transformada en algoritmos de compresión de audio.

Es el propósito de este trabajo estudiar el esquema de descomposición multiresolucional de la Transformada Wavelet y utilizar esta descomposición para experimentar en la codificación de señales de audio.

Nuestra propuesta se extiende desde realizar una breve introducción a los fundamentos del procesamiento de señales digitales hasta el desarrollo de una interfase gráfica para la compresión de audio.

En el Capítulo 1 se presenta un repaso de los fundamentos del procesamiento de señales digitales que incluye: características de las señales discretas en tiempo, sistemas lineales, invariantes, causales y estables, convolución, transformada Z, transformada de Fourier, filtros digitales y muestreo de señales continuas.

En el Capítulo 2 se hace una introducción a las diferentes técnicas de codificaciones de señales de voz, como: técnicas de codificación en el dominio del tiempo, técnicas de codificación en el dominio frecuencial y vocoders.

En el Capítulo 3 nos extendemos sobre un caso en particular de las técnicas de codificación en el dominio frecuencial, la codificación en sub-bandas, que está estrechamente relacionada con la transformada wavelet.

El Capítulo 4 está dedicado a la transformada wavelet. En este capítulo se explican el esquema de descomposición utilizando la transformada wavelet básica y su generalización en el esquema de descomposición usando transformada wavelet empaquetada.

En el Capítulo 5 se explican los diferentes tipos de algoritmos implementados ya sea usando descomposición por transformada wavelet básica o por transformada wavelet empaquetada.

El Capítulo 6 está dedicado a explicar una interfase windows para compresión de audio que implementamos con el objetivo de dar al algoritmo de compresión un soporte visual e interactivo.

En el Capítulo 7 se muestran las evaluaciones realizadas y los resultados obtenidos, así como también las conclusiones que se deducen de esta experiencia.

Además se incluyen dos Anexos en donde se analizan los siguientes temas: evaluación de calidad de la voz y filtros espejo en cuadratura.

Finalmente se incluyen disquetes con una copia de este trabajo, el ejecutable de la interfase windows y algunos archivos en formato WAV.

Este trabajo fue realizado por Juan Manuel García, José Roberto Quentrequeo y Hector Ismael Otero de la Facultad de Ciencias Exactas de la Universidad Nacional de La Plata; dirigidos por el Ing. Oscar N. Bría.



INDICE GENERAL

Prefacio	I
Indice General	II
1 Fundamentos del Procesamiento de Señales Digitales	1
1.1 Introducción	1
1.2 Caracterización de las señales discretas en tiempo	1
1.3 Sistemas lineales invariantes en el tiempo	2
1.3.4 Sistema lineal	2
1.3.5 Sistema invariante al tiempo	2
1.3.6 Convolución	2
1.7 Estabilidad y causalidad	3
1.4.8 Sistema estable	3
1.4.9 Sistema causal	4
1.10 Ecuaciones de diferencias de coeficientes constantes	4
1.5.1 Diagrama de flujo para implementar en paralelo	4
1.6 Transformada Z	5
1.7 Transformada de Fourier	11
1.8 Transformada de Fourier Discreta	12
1.9 Fundamentos de filtros digitales	13
1.9.10 Sistemas FIR	16
1.9.11 Sistemas IIR	17
1.12 Muestreo	18
1.10.13 Teorema del Muestreo	19
1.10.14 Decimación e interpolación de formas de ondas	20
1.10.2.a Decimación	21
1.10.2.b Interpolación	22
2 Codificación de Señales de Voz	25
2.1 Introducción	25
2.2 Códigos de forma de onda en el dominio del tiempo	25
2.2.3 Codificación por modulación de pulsos	25
2.2.4 Codificación por modulación de pulsos diferencial	27
2.2.5 Codificación por modulación adaptativa	29
2.2.6 Modulación delta	29
2.2.7 Modulación delta adaptativa	30
2.8 Códigos de forma de onda en el dominio frecuencial	32
2.3.9 Codificación en sub-bandas	32
2.3.10 Codificación adaptativa por transformadas	32
2.11 Vocoders	33
3 Codificación en Sub-Bandas	35

3.1	Introducción	35
3.2	Bancos de filtro de banda entera	38
3.3	Banco de filtros espejo en cuadratura	40
3.4	Codificación en sub-bandas de señales de voz	44
4	Transformada Wavelet	46
4.1	Introducción	46
4.2	Revisión de la Transformada de Fourier	46
4.3	Transformada Wavelet Rápida	46
4.4	Transformada Wavelet Empaquetada	50
4.4.5	Desde la DWT a la WPT	52
5	Algoritmos de Compresión	56
5.1	Introducción	56
5.2	Algoritmos de compresión	56
5.2.3	Algoritmos de DWT	56
5.2.4	Algoritmos de WPT	65
6	Interfase Windows para Compresión de Audio	72
6.1	Introducción	72
6.2	Barra de menús	72
6.3	Menú archivo	73
6.4	Menú operaciones	75
6.5	Menú ventana	79
7	Evaluación, Resultados y Conclusiones	81
7.1	Introducción	81
7.2	Evaluación y resultados	81
7.3	Conclusiones	86
A	Evaluación de Calidad de la Voz	88
A.1	Introducción	88
A.1.1	La necesidad de evaluar calidad	88
A.1.2	Calidad versus Inteligibilidad	89
A.2	Medidas de calidad subjetivas	89
A.3	Medidas de calidad objetivas	91
A.3.1	Relación Señal-Ruido	91
B	Filtros Espejo en Cuadratura	94
	Bibliografía	100

Capítulo 1

**FUNDAMENTOS DEL
PROCESAMIENTO DE SEÑALES**

FUNDAMENTOS DEL PROCESAMIENTO DE SEÑALES DIGITALES

1.1 Introducción

Una *señal* puede ser definida como una función que lleva información, generalmente acerca del estado o comportamiento de un sistema físico. En todos los casos la información está contenida en un patrón de variación de alguna forma (tiempo o espacio). Matemáticamente puede ser representada como una función de una o más variables independientes (que representan tiempo o espacio).

Una *señal continua en tiempo* es aquella donde la variable independiente toma valores continuos (reales) y una *señal discreta en tiempo* es donde la variable independiente toma valores discretos (enteros). Estas últimas son representadas como secuencias de números.

También, la amplitud de una señal puede ser continua o discreta. Las *señales digitales* son aquellas donde tanto la amplitud como el tiempo son discretos y las *señales analógicas* son donde la amplitud y el tiempo son continuos.

Un *sistema* es la transformación de una señal en otra que es en algún sentido más deseable que la original. Un *sistema continuo en tiempo* es uno donde la entrada y la salida son señales continuas en tiempo y un *sistema discreto en tiempo* es uno donde la entrada y la salida son señales discretas en tiempo. Por el mismo razonamiento, en un *sistema analógico* la entrada y la salida son señales analógicas y en un *sistema digital* la entrada y la salida son señales digitales.

1.2 Caracterización de las señales discretas en tiempo

Una señal discreta en tiempo es una secuencia $x(n)$ de muestras de una señal analógica $x_a(t)$ tomadas periódicamente con un periodo de muestreo T . Es decir,

$$x(n) = x_a(nT)$$

En los sistemas digitales se pueden encontrar secuencias como:

- *Sample unitario o secuencia de impulso unitario*

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & c.c. \end{cases}$$

- *Secuencia de paso unitario*

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & c.c. \end{cases}$$

- *Secuencia exponencial*

$$x(n) = a^n$$

Si a es complejo ($a = re^{jw_0}$), entonces:

$$x(n) = r^n e^{jw_0 n} = r^n (\cos(w_0 n) + j \sen(w_0 n))$$

1.3 Sistemas lineales invariantes en el tiempo

Un *sistema* es definido matemáticamente como una transformación única u operador que mapea una secuencia de entrada $x(n)$ en una secuencia de salida $y(n)$. Esto es denotado como:

$$y(n) = T[x(n)]$$

y muchas veces representado como muestra la *Figura 1.1*:

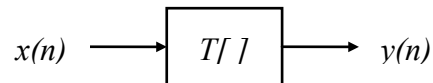


Figura 1.1: Representación de una transformación que mapea una secuencia de entrada $x(n)$ en una de salida $y(n)$.

1.3.1 Sistema lineal

Si $y_1(n)$ e $y_2(n)$ son las respuestas cuando $x_1(n)$ y $x_2(n)$ son las entradas respectivas, entonces un sistema es lineal si y solo si:

$$T(a x_1(n) + b x_2(n)) = a T(x_1(n)) + b T(x_2(n)) = a y_1(n) + b y_2(n)$$

para constantes arbitrarias a y b .

1.3.2 Sistema invariante al tiempo

Un sistema es invariante al cambio si vale: si $y(n)$ es la respuesta a $x(n)$, entonces $y(n-k)$ es la respuesta a $x(n-k)$, donde k es un entero. Cuando el índice n está asociado con el tiempo, que sea invariante al cambio significa que sea invariante al tiempo.

1.3.3 Convolución

Toda entrada en el sistema (excitación) se puede escribir como una combinación de impulsos:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k) \delta(n-k)$$

La salida al sistema será:

$$y(n) = T[x(n)] = T\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right]$$

por propiedad de linealidad:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)T[\delta(n-k)]$$

por propiedad de invariabilidad en el tiempo

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = x(n) * h(n) \quad [1.1.a]$$

haciendo $m=n-k$ obtenemos,

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m) = h(n) * x(n) \quad [1.1.b]$$

Por lo tanto, la salida de un sistema lineal invariante en el tiempo se obtiene como resultado de la convolución de la señal de entrada $x(n)$ con la respuesta al impulso del sistema, $y(n) = x(n) * h(n) = h(n) * x(n)$.

Observaciones:

- $h(n) = T[\delta(n)]$ es llamada la *respuesta al impulso*. Es decir, es la respuesta del operador T al impulso $\delta(n)$. Por la condición de invariabilidad en el tiempo que tiene este operador $h(n-k) = T[\delta(n-k)]$.
- Obtenemos la respuesta del sistema a $x(n)$ conociendo la respuesta al impulso del sistema. Es decir conociendo la función de respuesta al impulso obtenemos cualquier salida del sistema. Hay un $h(n)$ para cada operador T .

Definición: Un operador lineal invariante al tiempo queda unívocamente determinado por su respuesta al impulso.

1.4 Estabilidad y Causalidad

1.4.1 Sistema estable

Un sistema estable es aquel para el cual se cumple que si la entrada es acotada entonces la salida es acotada. Los sistemas invariantes al tiempo lineales son estables si y solo si,

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty \quad [1.2]$$

1.4.2 Sistema causal

Es aquel para el cual la salida para algún $n=n_0$ depende de la entrada para $n \leq n_0$, solamente. Por lo tanto un sistema invariante lineal es causal si y solo si la respuesta al impulso es cero para $n < 0$. La noción de causalidad en este último caso puede entenderse de la siguiente manera: El sistema es causal si y solo si hay respuesta después de la excitación (efecto posterior a la causa). Los sistemas causales son físicamente realizables (pueden ser implementados).

1.5 Ecuaciones de diferencias de coeficientes constantes

En muchas aplicaciones interesan los sistemas lineales invariantes para los cuales la entrada y la salida satisfacen una ecuación de diferencias lineal de orden N y de coeficientes constantes de la forma:

$$\sum_{k=0}^N a_k y(n-k) = \sum_{r=0}^M b_r x(n-r) \quad [1.3]$$

En general, un sistema de este tipo no será necesariamente causal. Si el sistema es causal pueden asumirse condiciones iniciales tales como $x(n)=0, n=n_0$ entonces $y(n)=0, n=n_0$.

De la ecuación [1.3] puede expresarse la salida del sistema en función de la entrada, en la forma:

$$y(n) = - \sum_{k=1}^N \frac{a_k}{a_0} y(n-k) + \sum_{r=0}^M \frac{b_r}{a_0} x(n-r)$$

Observación: La salida n -ésima del sistema puede determinarse en función de las salidas anteriores y, del valor actual y anteriores de la entrada.

$$y(n) = \sum_{k=1}^N \left(\frac{-a_k}{a_0} \right) y(n-k) + \frac{b_0}{a_0} x(n) + \sum_{r=1}^M \frac{b_r}{a_0} x(n-r)$$

1.5.1 Diagrama de flujo para implementar en paralelo (VLSI)

Si conocemos los coeficientes a_m y b_p podemos llegar a una implementación del sistema como se muestra en la *Figura 1.2*:

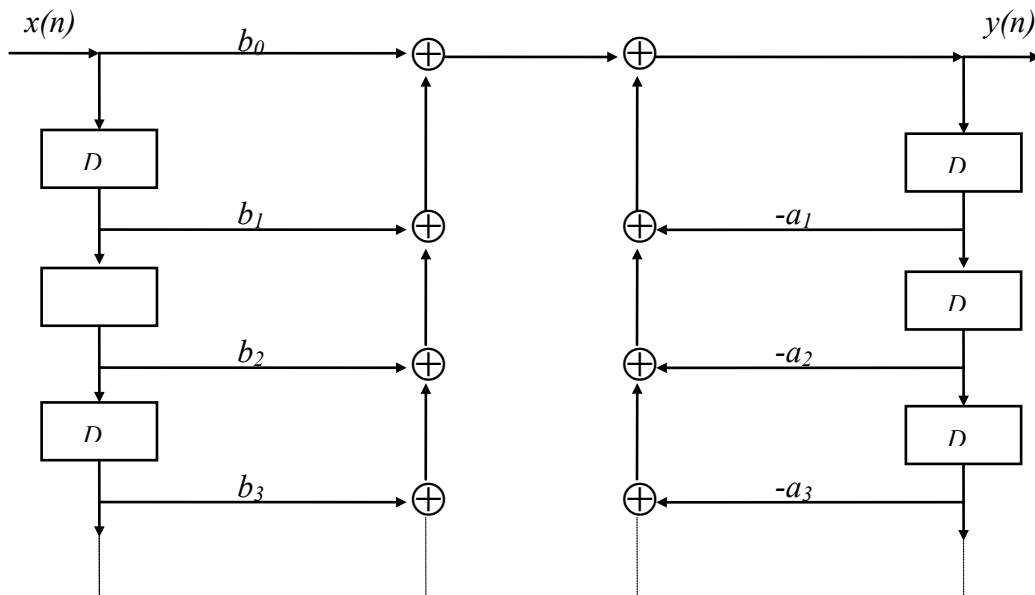


Figura 1.2: Implementación en paralelo de una ecuación de diferencias como la de [1.3].

1.6 Transformada Z

La transformada Z juega un rol importante en el análisis y representación de los sistemas invariantes en el tiempo lineal de tiempo discreto. La transformada Z de la secuencia $x(n)$, denotada con $X(z)$, se define como:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad [1.4]$$

en donde z es una variable compleja. La inversa de la transformada Z se define como:

$$x(n) = \frac{1}{2\pi j} \oint_c X(z)z^{n-1} dz \quad [1.5]$$

La transformada Z o transformada directa de $x(n)$ definida en la ecuación anterior, puede verse como una serie potencia infinita en la variable z^{-1} donde la secuencia de valores, $x(n)$, juegan el rol de coeficientes en la serie. En general estas series potencias convergerán a un valor finito para ciertos valores de z . Una condición suficiente para la convergencia es:

$$\sum_{n=-\infty}^{\infty} |x(n)||z^{-n}| < \infty \quad [1.6]$$

El conjunto de valores para los cuales la serie converge define una región en el plano complejo z llamada *región de convergencia*. En general esta región es de la forma:

$$R_1 < |z| < R_2$$

Una serie potencia de la forma de la ecuación [1.4] es una *serie de Laurent*. Una serie de Laurent y por consiguiente la transformada Z representan una función analítica en todo punto dentro de la región de convergencia, y por lo tanto la transformada Z y todas sus derivadas deben ser funciones continuas de z dentro de la región de convergencia.

Una clase importante de transformadas Z son aquellas para las cuales X(z) es una función racional, esto es, un cociente de polinomios en z. Las raíces del polinomio numerador son aquellos valores de z para los cuales X(z)=0 y son llamados *ceros de X(z)*. Los valores de z para los cuales X(z) es infinito son llamados *polos de X(z)*. Los polos de X(z) para valores finitos de z son las raíces del polinomio denominador.

Para las transformadas Z racionales hay un número de relaciones importantes entre la localización de los polos de X(z) y la región de convergencia de la transformada. Claramente, los polos no pueden localizarse dentro de la región de convergencia dado que la transformada Z no converge en un polo.

Ejemplo:

Considerar la secuencia:

$$x(n) = a^n u(n)$$

La transformada Z esta dada por:

$$X(z) = \sum_{n=-\infty}^{\infty} a^n u(n) z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n$$

la cual converge a:

$$X(z) = \frac{1}{1 - az^{-1}} \quad \text{para } |z| > |a|$$

Reescribiendo X(z) como:

$$X(z) = \frac{1}{1 - a/z} = \frac{1}{1/z(z-a)} = \frac{z}{z-a}$$

vemos que X(z) tiene un cero en z=0 y un polo en z=a. Esto se muestra en el plano z de la Figura 3.

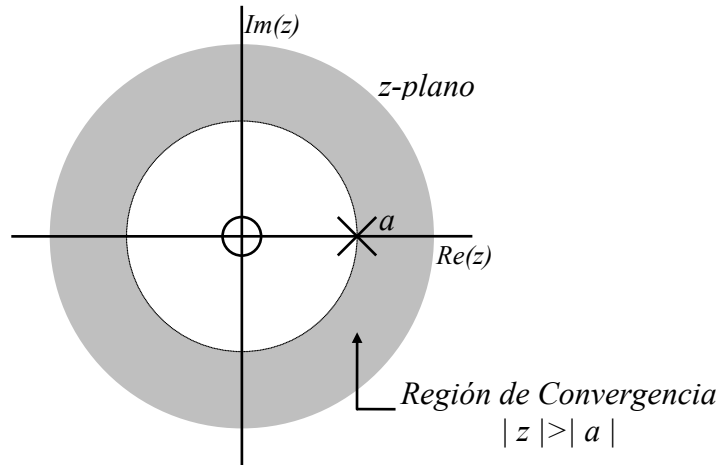


Figura 1.3: Región de convergencia en el z-plano de la función $a^n u(n)$.

Las propiedades de la secuencia $x(n)$ determinan la región de convergencia de $X(z)$. Para ver como la región de convergencia está relacionada con la secuencia planteamos los siguientes casos:

- *Secuencias de Longitud Finita:* Supongamos que solamente un número finito de valores de la secuencia son distintos de cero. Así que:

$$X(z) = \sum_{n=-n_1}^{n_2} x(n)z^{-n} \quad [1.7]$$

n_1 y n_2 dos enteros finitos. La convergencia de esta expresión requiere simplemente que $|x(n)| < \infty$ para $n_1 \leq n \leq n_2$. Entonces z puede tomar todos los valores excepto $z = \infty$ si $n_1 < 0$ y $z = 0$ si $n_2 > 0$. Por lo tanto la región de convergencia para una secuencia de longitud finita es al menos $0 < |z| < \infty$, y esta puede incluir $z = 0$ y $z = \infty$, si $n_2 < 0$ ó $n_1 > 0$, respectivamente.

- *Secuencias a derecha:* Una secuencia a derecha es una para la cual $x(n) = 0$ para $n < n_1$. La transformada Z de tal secuencia es:

$$X(z) = \sum_{n=n_1}^{\infty} x(n)z^{-n} \quad [1.8]$$

La región de convergencia de esta serie es el exterior de un círculo. Para ver esto, supongamos que la serie es absolutamente convergente para $z = z_1$ de modo que:

$$\sum_{n=n_1}^{\infty} |x(n)z_1^{-n}| < \infty \quad [1.9]$$

Considerando la serie [1.9], notemos que si $n_1 \geq 0$, entonces $|z| > |z_1|$, cada término es más chico que en la serie de la ecuación [1.9]. Por lo tanto:

$$\sum_{n=n_1}^{\infty} |x(n)z^{-n}| < \infty \quad \text{para } |z| > |z_1|$$

Si $n_1 < 0$, entonces expresemos la serie como:

$$\sum_{n=n_1}^{\infty} |x(n)z^{-n}| = \sum_{n=n_1}^{-1} |x(n)z^{-n}| + \sum_{n=0}^{\infty} |x(n)z^{-n}| \quad [1.10]$$

La primera serie en el término de la derecha de la ecuación [1.10] es finita para cualquier valor finito de z . La segunda serie converge para $|z| > |z_1|$ por el argumento anterior. Así si R_1 es el valor más chico de $|z|$ para el cual la serie de la ecuación [1.8] converge, entonces la serie converge para: $R_1 < |z|$ con la excepción de $z=\infty$ si $n_1 < 0$. Por lo tanto las secuencias a derecha de [1.10] tiene una región de convergencia que es el exterior de un círculo con radio R_1 . Observemos que si $n_1 \geq 0$, de modo que la secuencia es causal, la transformada Z convergirá a $z=\infty$. Así, si la región de convergencia de la transformada Z es el exterior de un círculo se trata de una secuencia a derecha. Más aún, si esta incluye $z=\infty$, la secuencia es causal.

- *Secuencia a izquierda:* Es una para la cual $x(n) = 0$ para $n > n_2$. La transformada Z es:

$$X(z) = \sum_{n=-\infty}^{n_2} x(n)z^{-n} \quad [1.11]$$

por cambio de variables $n=-m$ obtenemos

$$X(z) = \sum_{m=n_2}^{\infty} x(-m)z^m$$

Por lo tanto los resultados para secuencias a derecha son aplicados a este caso con n reemplazado por $-m$ y z reemplazado por z^{-1} la región de convergencia será el interior de un círculo, $|z| < R_2$, excepto $z = 0$, si $n_2 > 0$. Si la transformada z de una secuencia a izquierda converge en $z=0$, entonces la secuencia es cero para $n \geq 0$.

Ejemplo:

$$x(n) = -b^n u(-n-1)$$

$$X(z) = \sum_{n=-\infty}^{-1} -b^n z^{-n} = \sum_{n=1}^{\infty} -b^{-n} z^n = 1 - \sum_{n=0}^{\infty} -b^{-n} z^n$$

Esta serie converge si $|b^{-1}z| < 1$ ó $|z| < |b|$, en cuyo caso

$$X(z) = 1 - \frac{1}{1 - b^{-1}z} = \frac{z}{z - b} \quad \text{para } |z| < |b|$$

⇒ Si $z = 0$, tenemos un cero de la transformada

⇒ Si $z = b$, tenemos un polo de la transformada

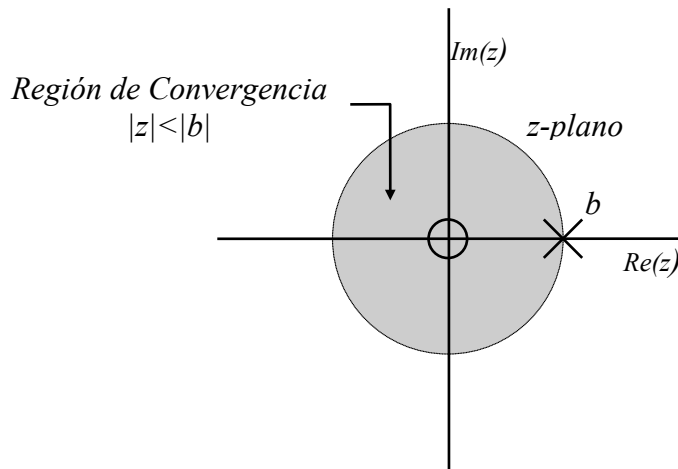


Figura 1.4: Polos y Región de convergencia en el z-plano

- *Secuencia a ambos lados*: Es una que se extiende desde $n = -\infty$ hasta $n = +\infty$, en general puede escribirse:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} = \sum_{n=-\infty}^{-1} x(n)z^{-n} + \sum_{n=0}^{\infty} x(n)z^{-n} \quad [1.12]$$

La primera serie converge para $|z| < R_2$ por [1.9]. La segunda serie converge para $|z| > R_1$ por [1.8]. Si $R_1 < R_2$ hay una región de convergencia en común de la forma

$$R_1 < |z| < R_2$$

Si $R_1 > R_2$ no hay región de convergencia en común y por lo tanto la serie no converge. Si $R_1 < 1 < R_2$ la secuencia tiene transformada de Fourier y transformada Z.

Ejemplo:

$$x(n) = \begin{cases} a^n & n \geq 0 \\ -b^n & n \leq -1 \end{cases}$$

donde $|a| < |b|$

Usando el resultado de los dos ejemplos previos

$$X(z) = \frac{z}{z-a} + \frac{z}{z-b} = \frac{z(2z-a-b)}{(z-a)(z-b)}$$

donde la región de convergencia es:

$$|a| < |z| < |b|$$

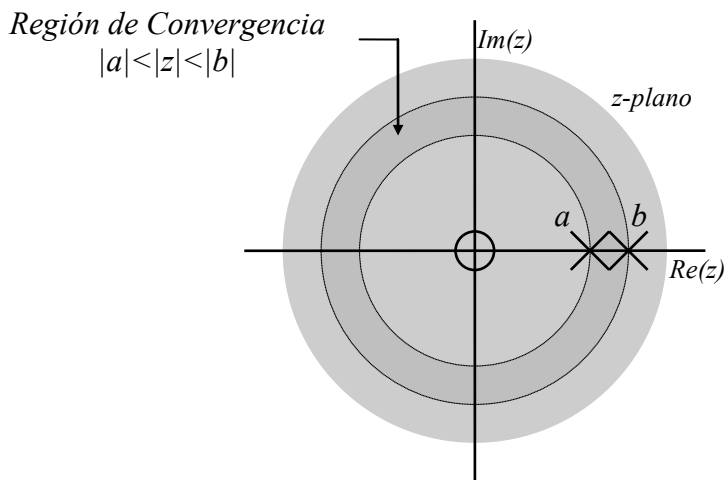


Figura 1.5: Región de convergencia en el z-plano.

A continuación describiremos una lista de teoremas y propiedades de la transformada Z. Estos teoremas pueden ser vistos en forma similar a los teoremas correspondientes a la transformada de Laplace de funciones continuas en tiempo.

Sin embargo, la transformada Z en ningún sentido es una aproximación a la de Laplace. La transformada de Laplace es una representación exacta de una función en tiempo continuo y la transformada Z es una representación exacta de una secuencia. La relación entre representaciones continuas y discretas de una señal se realizará a través del teorema del muestreo.

Las propiedades de lo transformado Z se resumen en la tabla de la *Tabla 1.1*:

Propiedad	Secuencia	Transformada Z
<i>Linealidad</i>	$ax_1(n)+bx_2(n)$	$aX_1(z)+bX_2(z)$
<i>Invariante</i>	$x(n+n_0)$	$z^{n_0} X(z)$
<i>Multiplicación Exponencial</i>	$a^n x(n)$	$X(a^{-1}z)$
<i>Multiplicación lineal</i>	$nx(n)$	$-z \cdot \frac{dX(z)}{dz}$
<i>Tiempo inverso</i>	$x(-n)$	$X(z^{-1})$
<i>Convolución</i>	$x(n)*h(n)$	$X(z).H(z)$
<i>Multiplicación de secuencias</i>	$x(n).w(n)$	$\frac{1}{2\pi j} \oint_c X(v)W(z/v)v^{-1} dv$

Figura 1.1: Propiedades de la transformada Z

1.7 Transformada de Fourier

La representación de la transformada de Fourier de una señal discreta en tiempo esta dada por las ecuaciones:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad \infty < \omega < \infty \quad [1.13.a]$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega \quad \infty < \omega < \infty \quad [1.13.b]$$

Estas ecuaciones pueden ser vistas como un caso especial de las ecuaciones de la transformada Z. Especificamente la representación de Fourier es obtenida restringiendo la transformada Z al círculo unidad del plano z, seteando $z = e^{j\omega}$. La *Figura 1.6* muestra el círculo de unidad del plano z, donde ω es un ángulo en el z plano que indica frecuencia.

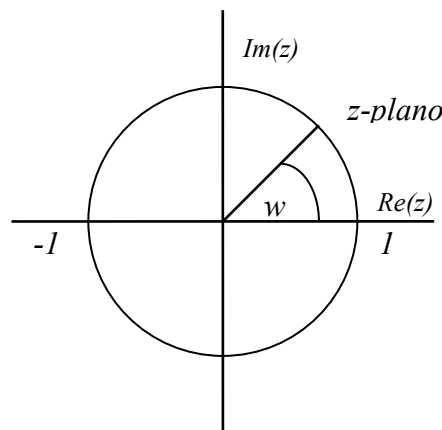


Figura 1.6: Círculo de unidad del z-plano

Una condición suficiente para la existencia de una representación de la transformada de Fourier puede ser obtenido seteando $|z| = 1$ en la ecuación [1.9]

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty \quad [1.14]$$

Una característica importante de la transformada de Fourier de una secuencia es que $X(e^{j\omega})$ es una función periódica de ω con periodo 2π

1.8 Transformada de Fourier discreta

Como en el caso de señales analógicas, si una señal es periódica con período N entonces

$$\bar{X}(k) = \sum_{n=0}^{N-1} \bar{x}(n+rN) \quad -\infty < n < \infty, -\infty < r < \infty \quad [1.15]$$

y alcanza con analizar un solo periodo para determinar el contenido frecuencial de la señal $x(n)$.

Por lo tanto, la representación en Serie de Fourier para una secuencia periódica es:

$$\bar{X}(k) = \sum_{n=0}^{N-1} \bar{x}(n) e^{-j\frac{2\pi}{N}kn} \quad [1.16.a]$$

$$\bar{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \bar{X}(k) e^{j\frac{2\pi}{N}kn} \quad [1.16.b]$$

Consideremos ahora una secuencia de longitud finita $X(n)$ que es distinta de cero en el intervalo $0 \leq n \leq N-1$. Entonces la transformada Z es:

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n} \quad [1.17]$$

Si evaluamos $X(z)$ en N puntos igualmente espaciados del círculo de radio 1,

$$z_k = e^{j\frac{2\pi k}{N}} \quad k = 0 \dots N-1$$

entonces obtenemos

$$X(e^{j\frac{2\pi k}{N}}) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \quad k = 0 \dots N-1 \quad [1.18]$$

Si construimos una secuencia periódica como una secuencia infinita de replicas de $x(n)$

$$\bar{x}(n) = \sum_{r=-\infty}^{\infty} x(n+rN) \quad [1.19]$$

Entonces, de las ecuaciones [1.16.a] y [1.18] podemos ver que las muestras $X(e^{j2\pi k/N})$ son los coeficientes de Fourier de una secuencia periódica $\bar{x}(n)$ como la ecuación [1.19]. Por lo tanto, una secuencia de longitud N puede representarse exactamente por una transformada de Fourier discreta de la forma:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad k = 0 \dots N-1 \quad [1.20.a]$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \quad n = 0 \dots N-1 \quad [1.20.b]$$

Todas las secuencias se comportan como si fueran periódicas cuando son representados por una DFT.

Una notación alternativa a la de la ecuación [1.19] es

$$\bar{x}(n) = x((n))_N$$

Las propiedades de la transformada de Fourier discreta se resumen en la *Tabla 1.2*:

Propiedad	Secuencia	DFT de N-puntos
<i>Linealidad</i>	$ax_1(n) + bx_2(n)$	$aX_1(k) + bX_2(k)$
<i>Invariante</i>	$x((n+n_0))_N$	$e^{j\frac{2\pi}{N}kn_0} X(k)$
<i>Tiempo inverso</i>	$x((-n))_N$	$X^*(k)$
<i>Convolución</i>		$X(k)H(k)$
<i>Multiplicación de secuencias</i>	$x(n)w(n)$	$\frac{1}{N} \sum_{r=0}^{N-1} X(r)W((k-r))_N$

Tabla 1.2: Propiedades de la transformada de Fourier discreta

1.9 Fundamentos de filtros digitales

Un filtro digital es un sistema discreto en tiempo lineal e invariante. Recordemos que para tal sistema la entrada y la salida están relacionadas por la expresión suma de convolución

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad [1.21]$$

Si aplicamos la transformada Z a la secuencia $y(n)$ obtenemos:

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n)z^{-n}$$

Reemplazando $y(n)$ por [1.21] obtendremos

$$Y(z) = \sum_{n=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x(k)h(n-k) \right] z^{-n}$$

Si intercambiamos el orden de suma

$$Y(z) = \sum_{k=-\infty}^{\infty} x(k) \sum_{n=-\infty}^{\infty} h(n-k)z^{-n}$$

Cambiando el índice de suma en la segunda sumatoria $m = n-k$ obtenemos

$$Y(z) = \sum_{k=-\infty}^{\infty} x(k) \left[\sum_{m=-\infty}^{\infty} h(m)z^{-m} \right] z^{-k}$$

$$Y(z) = \sum_{k=-\infty}^{\infty} x(k)z^{-k} \sum_{m=-\infty}^{\infty} h(m)z^{-m}$$

$$Y(z) = X(z)H(z) \quad [1.22]$$

De este modo, obtenemos la transformada Z de la salida como el producto de la transformada Z de la entrada y la transformada Z de la respuesta al impulso. La transformada Z de la respuesta al impulso, $H(z)$, es llamada función del sistema.

La transformada de Fourier de la respuesta al impulso $H(e^{j\omega})$ es llamada la respuesta frecuencial. $H(e^{j\omega})$ es en general, una función compleja de ω la cual puede ser expresada en términos de parte real y parte imaginaria como

$$H(e^{j\omega}) = H_r(e^{j\omega}) + j H_i(e^{j\omega}) \quad [1.23]$$

o en términos de magnitud y ángulo fase como:

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j \arg[H(e^{j\omega})]} \quad [1.24]$$

Un sistema lineal invariante al tiempo *causal* es uno para el cual $h(n) = 0$ para $n < 0$. Un sistema estable es uno para el cual toda entrada limitado produce una salida limitada. Una condición necesaria y suficiente para que un sistema invariante al tiempo lineal sea estable es:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \quad [1.25]$$

Esta condición es idéntica a la ecuación [1.14] y por lo tanto es suficiente para la existencia de $H(e^{j\omega})$.

Además de la expresión suma de convolución, todo sistema invariante al tiempo lineal tiene la propiedad que la entrada y la salida satisfacen una ecuación de diferencias lineal de la forma:

$$y(n) = \sum_{r=0}^M b_r x(n-r) - \sum_{k=1}^N a_k y(n-k) \quad [1.26]$$

Aplicando la transformada Z a ambos lados de la igualdad

$$Z\{y(n)\} = Z\left\{\sum_{r=0}^M b_r x(n-r) - \sum_{k=1}^N a_k y(n-k)\right\}$$

$$Y(z) = \sum_{r=0}^M b_r Z\{x(n-r)\} - \sum_{k=1}^N a_k Z\{y(n-k)\}$$

Por *Propiedad 2* de la Transformada Z

$$Y(z) = \sum_{r=0}^M b_r z^{-r} X(z) - \sum_{k=1}^N a_k z^{-k} Y(z)$$

$$Y(z) + \sum_{k=1}^N a_k z^{-k} Y(z) = \sum_{r=0}^M b_r z^{-r} X(z)$$

$$Y(z) \left[1 + \sum_{k=1}^N a_k z^{-k} \right] = X(z) \sum_{r=0}^M b_r z^{-r}$$

$$\Rightarrow \frac{Y(z)}{X(z)} = \frac{\sum_{r=0}^M b_r z^{-r}}{1 + \sum_{k=1}^N a_k z^{-k}} = H(z)$$

$$\therefore \frac{Y(z)}{X(z)} = Z\{h(n)\} = H(z) \quad [1.27]$$

Por lo tanto, podemos obtener $H(z)$ a partir de la ecuación de diferencias identificando los coeficientes b_r y a_k .

La función del sistema $H(z)$, es en general una función racional de z^{-1} . Como tal está caracterizada por la localización de sus polos y ceros en el plano z .

Específicamente $H(z)$ puede ser expresada, factorizándola en función de sus raíces, como:

$$H(z) = \frac{A \prod_{r=1}^M (1 - c_r z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \quad [1.28]$$

Como vimos un sistema causal tendrá un a región de convergencia de la forma $|z| > R_I$. Si el sistema es también estable $R_I < 1$, de modo que la región de convergencia contenga al círculo de unidad. Por lo tanto los polos de $H(z)$ deben estar dentro del círculo de radio 1 para un sistema estable y causal.

Hay dos clases de sistemas invariantes al tiempo lineal: aquellos de repuesta de impulso de duración finita (FIR) y los de respuesta de impulso de duración infinita (IIR).

1.9.1 Sistemas FIR

Si todos los coeficientes a_k en la ecuación [1.26] son ceros, la ecuación de diferencia da:

$$y(n) = \sum_{r=0}^M b_r x(n-r) \quad [1.29]$$

ahora de la ecuación de suma de convolución tenemos

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

Comparando ambas ecuaciones observamos que

$$h(n) = \begin{cases} b_n & 0 \leq n \leq M \\ 0 & c. c. \end{cases}$$

Los sistemas FIR tienen un número de propiedades importantes:

- $H(z)$ es una polinomial en z^{-1} y por lo tanto no tiene polos distintos de ceros, solamente ceros.
- Los sistemas FIR pueden tener exactamente fase lineal. Si $h(n)$ satisface la relación

$$h(n) = \pm h(M-n) \quad [1.30]$$

entonces $H(e^{j\omega})$ tiene la forma:

$$H(e^{j\omega}) = A(e^{j\omega}) e^{-j\omega \frac{M}{2}}$$

donde $A(e^{j\omega})$ es real o imaginaria dependiendo de si la ecuación [1.30] se satisface con + o -, respectivamente.

La posibilidad de tener “exactamente” fase lineal es muy utilizada en aquellas aplicaciones de procesamiento de señales de voz donde la alineación en tiempo es esencial.

La desventaja al diseñar filtros con respuesta de fase lineal es que se requiere una respuesta de impulso muy grande para aproximar adecuadamente el corte abrupto (cutoff) de los filtros.

Basados en las propiedades de los filtros de fase lineal FIR se han desarrollado tres métodos de diseño que aproximar un conjunto arbitrario de especificaciones. Estos métodos son:

- 1 - Diseño windows.
- 2 - Diseño de frecuencias de muestreo.
- 3 - Diseño óptimo.

Un filtro FIR puede ser representado a través de un diagrama de bloques como se ve en *Figura 1.7*:

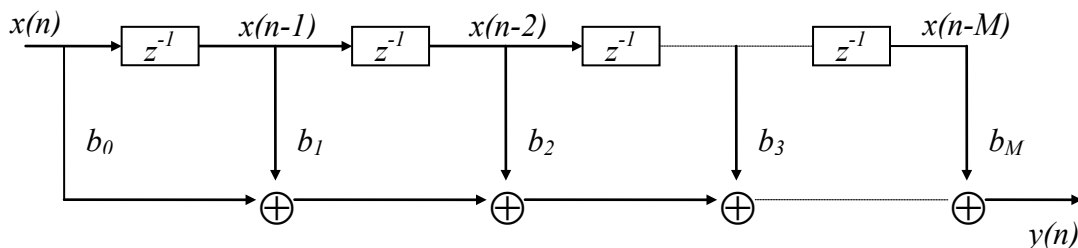


Figura 1.7: Diagrama digital para sistemas FIR

1.9.2 Sistemas IIR

Si la función del sistema de la ecuación [1.28] tiene polos y ceros, entonces la ecuación de diferencias [1.26] puede ser escrita como:

$$y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{r=0}^M b_r x(n-r) \quad [1.31]$$

Esta ecuación es una fórmula de recurrencia que puede usarse secuencialmente para calcular los valores de la secuencia de salida a partir de los N valores anteriores de salida, el valor actual de la entrada y los M valores anteriores de entrada. Si $M < N$ en la ecuación [1.28], $H(z)$ puede expandirse en una suma parcial de fracciones como en:

$$H(z) = \sum_{k=1}^N \frac{A_k}{1 - d_k z^{-1}}$$

Para un sistema causal se ve fácilmente que:

$$h(n) = \sum_{k=1}^N A_k d_k^n u(n)$$

Por lo tanto vemos que $h(n)$ tiene duración infinita. Sin embargo, dada la ecuación de recurrencia de la ecuación [1.31], es posible implementar un filtro IIR que aproxime a un conjunto dado de especificaciones más eficientemente (usando menos cálculos) que con un sistema FIR. Esto es particularmente cierto para filtros con cortes abruptos en frecuencia (cutoff).

La mayor diferencia entre filtros FIR e IIR es que:

- Los IIR no pueden ser diseñados para tener una fase lineal exacta, en cambio los filtros FIR pueden tener esta propiedad.
- Los filtros IIR son más eficientes para realizar cortes abruptos que los filtros FIR.

Hay una considerable flexibilidad para las implementaciones de los sistemas IIR. La implementación más común es la que se describe a continuación.

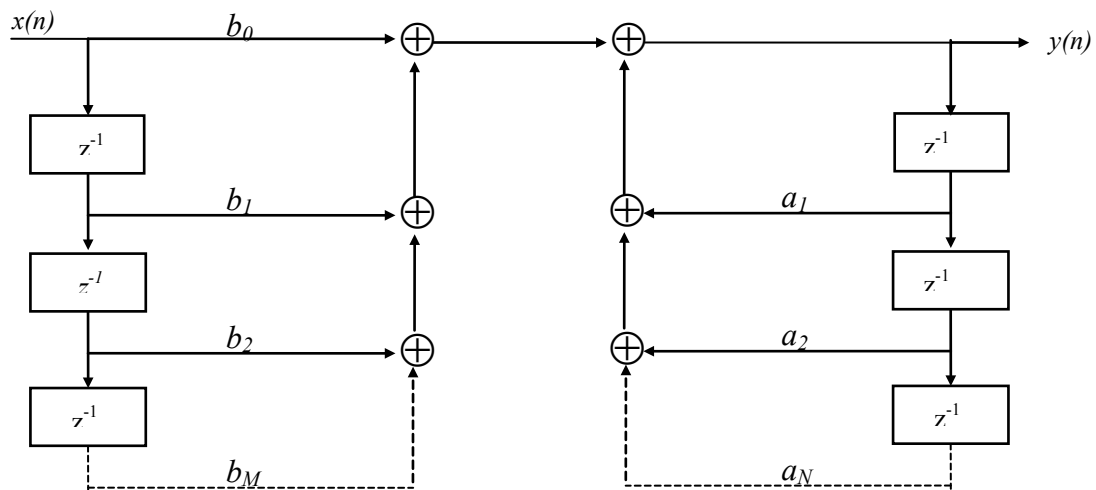


Figura 1.8: Diagrama digital para la implementación de un IIR.

1.10 Muestreo

Para utilizar métodos de procesamiento de señales digitales sobre una señal analógica, tal como una *señal de voz*, es necesario representar la señal como una secuencia de números. Esto se realiza comúnmente muestreando periódicamente las señales analógicas ($x_a(t)$) produciendo la secuencia:

$$x(n) = x_a(nT) \quad -\infty < n < \infty \quad n \in \mathbb{Z}$$

donde T es el período de muestreo.

1.10.1 Teorema del Muestreo

Si una señal $x_a(t)$ tiene una transformada de Fourier limitada en banda $X_a(j\Omega)$ tal que $X_a(j\Omega) = 0$ para $|\Omega| \geq 2\pi F_N$, entonces $x_a(t)$ puede ser reconstruida unívocamente desde las muestras igualmente espaciadas $x(n) = x_a(nT)$ $-\infty < n < \infty$ $n \in \mathbb{Z}$ si $1/T > 2F_N$ ¹.

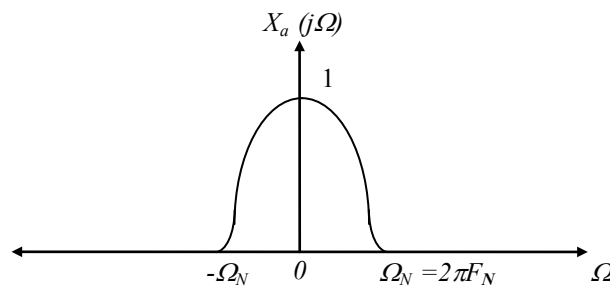
Este teorema se deduce del hecho que si la transformada de Fourier de $x_a(t)$ es definida como

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} dt \quad [1.32]$$

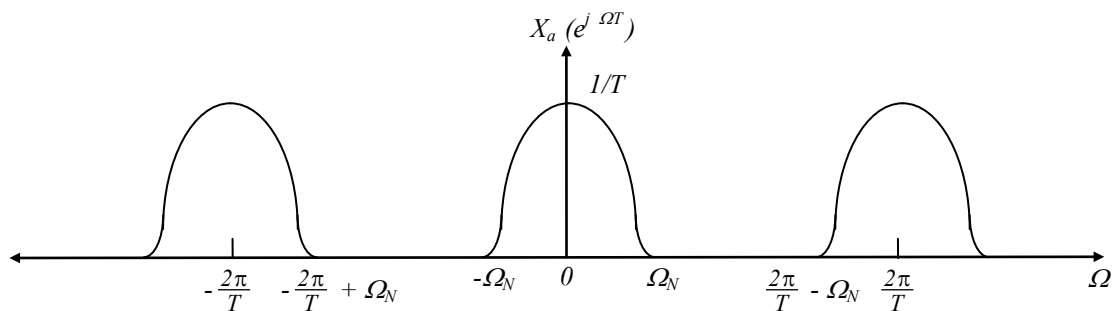
y la transformada de Fourier de la secuencia $x(n)$ es definida como la ecuación [1.13a], entonces si $X(e^{j\omega T})$ es evaluada para frecuencias $\omega = \Omega T$ entonces $X(e^{j\Omega T})$ es relacionada a $X_a(j\Omega)$ por:

$$X(e^{j\Omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a(j\Omega + j\frac{2\pi}{T}k) \quad [1.33]$$

Interpretación

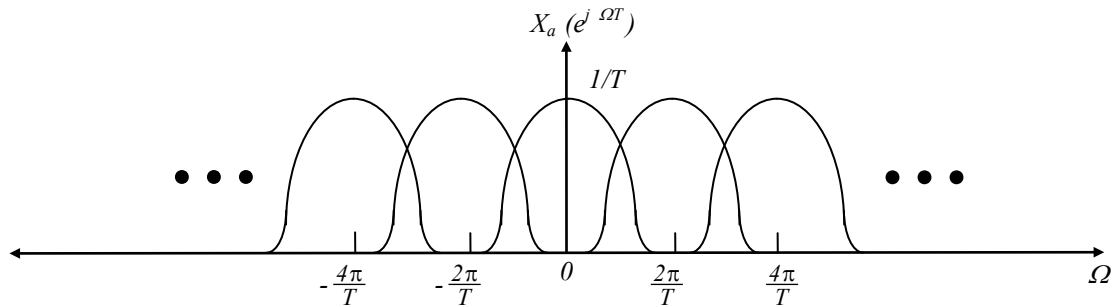


$X_a(j\Omega)$ representa la transformada de Fourier limitada en banda, es decir $X_a(j\Omega) = 0$ para $|\Omega| \geq \Omega_N = 2\pi F_N$



¹ F_N : Frecuencia de Nyquist: Es la frecuencia máxima de $X_a(j\Omega)$

$X(e^{j\Omega T})$ es la suma de un número infinito de replicas de $X_a(j\Omega)$, cada una centrada en un múltiplo entero de $2\pi/T$. Esta figura describe el caso cuando $1/T > 2F_N$, de modo que las imágenes de $X_a(j\Omega)$ no se superponen en la banda base $|\Omega| < 2\pi F_N$



La Figura anterior muestra el caso cuando $1/T < 2F_N$. En este caso la imagen centrada en $2\pi/T$ se superpone en la banda base. Esta condición donde una frecuencia alta aparentemente toma la identidad de una frecuencia baja es llamada *aliasing*. Claramente el *aliasing* puede evitarse si la transformada de Fourier es limitada en banda y si la frecuencia de muestreo es $1/T > 2F_N$.

Bajo la condición que $1/T > 2F_N$, es claro que la T.F de la secuencia de muestras es proporcional a la T.F. de la señal analógica en la banda base,

$$X(e^{j\Omega T}) = \frac{1}{T} X_a(j\Omega) \quad |\Omega| < \frac{\pi}{T}$$

Usando este resultado podemos ver que la señal original puede estar relacionada a la secuencia de muestras por la fórmula de interpolación

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT) \left[\frac{\text{sinc}[\pi(t - nT) / T]}{\pi(t - nT) / T} \right] \quad [1.34]$$

Por lo tanto, dadas las muestras de una señal analógica limitada en banda, tomadas a un rango de muestreo de al menos dos veces la frecuencia de Nyquist, es posible reconstruir la señal analógica original usando la ecuación anterior. Los conversores analógico-digitales buscan aproximar la ecuación anterior.

1.10.2 Decimación e Interpolación de Formas de Ondas Muestreadas

En muchos casos en el procesamiento de señales digitales es necesario cambiar el rango de muestreo de una señal discreta en tiempo. Los procesos para reducir o incrementar el rango de muestreo son llamados decimación e interpolación. Para tratar ambos casos asumiremos que tenemos una secuencia de muestras $x(n) = x_a(nT)$, donde la función tiene una transformada de Fourier limitada en banda. Esto es

$$X_a(j\Omega) = 0 \quad \text{para} \quad |\Omega| > 2\pi F_N$$

Entonces si $\frac{1}{T} > 2F_N$, la transformada de Fourier de $x(n)$ satisficará:

$$X(e^{j\Omega T}) = \frac{1}{T} X_a(j\Omega) \quad |\Omega| < \frac{\pi}{T} \quad [1.35]$$

1.10.2(a) Decimación

Supongamos que queremos reducir el rango de muestreo por un factor M , por ejemplo: queremos computar una nueva secuencia de muestras $x_a(t)$ tomadas con periodo $T' = M \cdot T$.

$$y(n) = x_a(nT') = x_a(nTM)$$

dado que $x(n) = x_a(nT)$, entonces:

$$y(n) = x(Mn) \quad -\infty < n < \infty$$

Es decir $y(n)$ es obtenida reteniendo periódicamente una de cada M muestras. Por el teorema del muestreo vemos que si $1/T > 2F_N$, entonces las muestras de $y(n)$ serán también adecuadas para representar unívocamente la señal original. Las transformadas de Fourier de $x(n)$ e $y(n)$ están relacionadas por la expresión [Schafer y Rabiner]:

$$Y(e^{j\Omega T'}) = \frac{1}{M} \sum_{k=0}^{M-1} X(e^{j(\Omega T' - 2\pi k)/M})$$

De esta ecuación se puede ver que no hay superposición entre las imágenes de $X(e^{j\Omega T})$ si tenemos $\frac{1}{T} > 2F_N$. Si esta condición vale, vemos que:

$$\begin{aligned} Y(e^{j\Omega T'}) &= \frac{1}{M} X(e^{j\Omega T'/M}) \\ &= \frac{1}{M} \frac{1}{T} X_a(j\Omega) \\ &= \frac{1}{T'} X_a(j\Omega) \quad -\frac{\pi}{T'} < \Omega < \frac{\pi}{T'} \end{aligned}$$

Las siguientes figuras muestran un ejemplo de decimación. La *Figura 1.9(b)* muestra la Transformada de Fourier de $x(n) = x_a(nT)$ donde el rango de muestreo $1/T$ es algo más grande que $2F_N$. La *Figura 1.9(c)* muestra el caso de reducción del rango de muestreo por un factor de 3 ($T' = 3T$). Para este caso ocurre *aliasing* porque $1/T' < 2F_N$. Sin embargo, supongamos que $x(n)$ es filtrada con un filtro digital pasa bajo con frecuencia de corte $\pi/T' = \pi/3T$ produciendo una secuencia $w(n)$. Para nuestro ejemplo la T.F. de la salida del filtro pasa-bajo se muestra en la *Figura 1.9(d)*. El *aliasing* no ocurre cuando el rango de muestreo de la señal muestreada es reducido por un factor de 3, como muestra la *Figura 1.9(e)*; sin embargo las muestras de $y(n)$ no representa a $x_a(t)$, sino a una nueva señal $y_s(t)$ la cual es una versión filtrada pasa-bajo de $x_a(t)$.

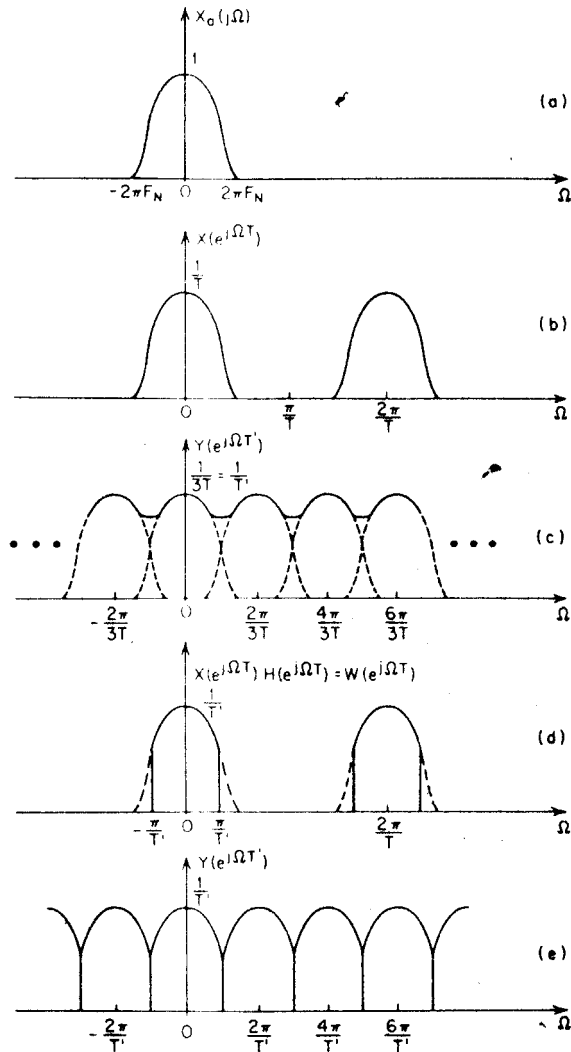


Figura 1.9: Decimación de una señal

La siguiente figura muestra un diagrama en bloques de de un sistema de decimación general:

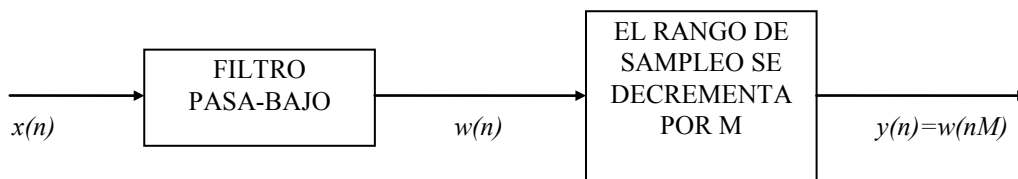


Figura 1.10: Diagrama en bloques de decimación

1.10.2(b) Interpolación

Ahora supongamos que tenemos muestras de una forma de onda analógica $x(n) = x_a(nT)$. Si nosotros deseamos aumentar el rango de muestreo por un factor entero L , debemos computar una secuencia nueva correspondiente a muestras de $x_a(t)$ tomada con periodo $T' = T/L$; p. ej.,

$$y(n) = x_a(nT') = x_a(nT/L) \quad [1.36]$$

Claramente, $y(n) = x(n/L)$ para $n = 0, \pm L, \pm 2L, \dots$ pero debemos llenar las muestras desconocidas para los otros valores de n por un proceso de interpolación [Schafer y Rabiner]. Para ver como se lo puede hacer usando un filtro digital, considerar la secuencia

$$v(n) = \begin{cases} x\left[\frac{n}{L}\right] & n = 0, \pm L, \pm 2L, \dots \\ 0 & \text{C. C.} \end{cases} \quad [1.37]$$

La transformada de Fourier de $v(n)$ es:

$$\begin{aligned} V(e^{j\Omega T'}) &= X(e^{j\Omega T'L}) \\ &= X(e^{j\Omega T}) \end{aligned} \quad [1.38]$$

Así $V(e^{j\Omega T'})$ es periódica con período $2\pi/T = 2\pi/(LT')$, así como también con período $2\pi/T'$ como es el caso en general de las secuencias asociadas con un período muestreo T' . La *Figura 1.11(a)* muestra $V(e^{j\Omega T'})$ [y $X(e^{j\Omega T})$] para el caso $T' = T/3$. A fin de obtener la secuencia

$$y(n) = x_a(nT')$$

desde la secuencia $v(n)$, debemos asegurar que

$$Y(e^{j\Omega T'}) = \frac{1}{T'} X_a(j\Omega) \quad -\frac{\pi}{T'} \leq \Omega \leq \frac{\pi}{T'} \quad [1.39]$$

Asumiendo que

$$X(e^{j\Omega T}) = \frac{1}{T} X_a(j\Omega) \quad -\frac{\pi}{T} \leq \Omega \leq \frac{\pi}{T} \quad [1.40]$$

entonces es claro desde la *Figura 1.11(a)* que se requiere que las imágenes de $X_a(j\Omega)$ en $V(e^{j\Omega T'})$, que sean centrada en $\Omega = 2\pi/T$ y $\Omega = 4\pi/T$, deben ser quitados por un filtro pasa-bajo digital que rechace todos los componentes de frecuencia en la gama $\pi/T \leq \Omega \leq 3\pi/T$. Además, para asegurar que la amplitud es correcta para un intervalo de muestreo T' , la ganancia del filtro debe ser $L = T/T'$. Esto es

$$\begin{aligned} Y(e^{j\Omega T'}) &= H(e^{j\Omega T'}) V(e^{j\Omega T'}) = H(e^{j\Omega T'}) X(e^{j\Omega T}) \\ &= H(e^{j\Omega T'}) \frac{1}{T} X_a(j\Omega) \end{aligned} \quad [1.41]$$

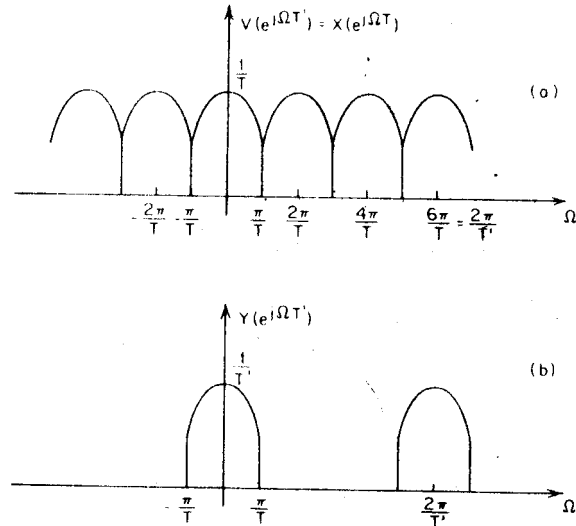


Figura 1.11: Interpolación de una señal

Así, para que $Y(e^{j\Omega T'}) = (1/T')X_a(j\Omega)$ para $\Omega \leq \pi/T'$, requerimos que

$$\begin{aligned} H(e^{j\Omega T'}) &= L & |\Omega| \leq \frac{\pi}{T} \\ &= 0 & \text{c.c.} \end{aligned} \quad [1.42]$$

El sistema general de interpolación se muestra en la Figura 1.12:

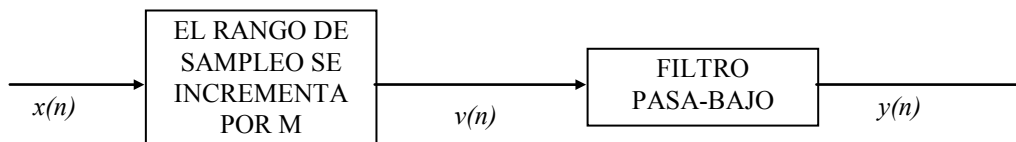


Figura 1.12: Diagrama en bloques de decimación



INFORMATICA
UNIVERSIDAD NACIONAL DE LA PLATA
U.N.L.P.

Capítulo 2

**CODIFICACIÓN DE SEÑALES DE
VOZ**

CODIFICACION DE SEÑALES DE VOZ

2.1 Introducción

Los métodos para representar digitalmente las características espaciales o temporales de la forma de onda de las señales de voz son generalmente llamados códigos de forma de onda o codificación de forma de onda.

Existen varias técnicas de codificación de la forma de onda de las señales de voz, esta codificación puede ser en el dominio del tiempo o en el dominio frecuencial.

2.2 Códigos de formas de onda en el dominio del tiempo

Algunas de las principales técnicas de cuantificación y codificación aplicadas a señales de voz son: codificación por modulación de pulsos (PCM), codificación por modulación de pulso diferencial (DPCM), modulación delta (DM), y las versiones adaptativas de estas técnicas.

2.2.1 - Codificación por modulación de pulsos

Sea $S_a(t)$ una forma de onda de una señal de voz analógica y sea $s(n)$ la representación de las muestras tomadas a una frecuencia de muestreo $f_s \geq 2w$. Donde w es la frecuencia mayor que se presenta en el espectro de la señal analógica $S_a(t)$.

En la técnica (PCM) cada muestra de la señal es cuantizada a uno de los 2^R niveles, donde R es el número de bit usados par representar cada muestra. Así tasa de generación de bits para la fuente es de Rf_s bps.

La forma de onda cuantizada puede ser modelada matemáticamente como:

$$\tilde{s}(n) = s(n) + q(n) \quad [2.1]$$

donde $\tilde{s}(n)$ representa los valores cuantizados de $s(n)$ y $q(n)$ representa el error de la cuantización, el cual tratamos como ruido aditivo.

Asumiendo que se usa un cuantizador uniforme y un número de niveles suficientemente grande, el ruido de cuantización es estadísticamente caracterizado como una realización de un proceso aleatorio estacionario q en el cual cada variable aleatoria $q(n)$ tiene la pdf uniforme.

$$f_{q(n)}(\xi) = \frac{1}{\Delta} \quad -\frac{\Delta}{2} \leq \xi \leq \frac{\Delta}{2} \quad [2.2]$$

donde el paso del cuantizador es $\Delta = 2^{-R}$.

El valor cuadrático medio del error de cuantización es:

$$\xi\{q^2(n)\} = \frac{\Delta^2}{12} = \frac{2^{-2R}}{12} \quad [2.3]$$

para un n arbitrario. Medido en decibeles el valor cuadrático medio del error es:

$$10 \log_{10} \frac{\Delta^2}{12} = 10 \log_{10} \frac{2^{-2R}}{12} = -6R - 10.8 \text{ dB} \quad [2.4]$$

Las señales de voz tienen la característica que las amplitudes pequeñas ocurren más frecuentemente que las amplitudes grandes. Sin embargo, los cuantizadores uniformes proveen el mismo espacio entre sucesivos niveles a través de todo el rango dinámico de la señal. Una mejor aproximación es el uso de un cuantizador no uniforme que provee niveles de espaciado más cortos en amplitudes pequeñas y niveles de espaciado más largos en amplitudes más grandes relativo a una cuantización uniforme. Para un cuantizador no uniforme con R bits, el error de cuantización resultante tiene un valor cuadrado medio que es más pequeño que el dado por [2.3]. Un cuantizador no uniforme característico es usualmente obtenido pasando la señal a través de un dispositivo no lineal que comprima la amplitud, seguido de un cuantizador uniforme. Un compresor logarítmico empleado en Norte América, llamada μ -law, tiene una magnitud de entrada-salida característica de la forma

$$|y| = \frac{\log(1 + \mu|s|)}{\log(1 + \mu)} \quad [2.5]$$

donde $|s|$ es la magnitud de la entrada, $|y|$ es la magnitud de la salida, y μ es un parámetro que es seleccionado para dar la característica de compresión deseada. La *Figura 2.1* ilustra la relación de compresión para varios valores de μ . El valor de $\mu=0$ corresponde a no compresión.

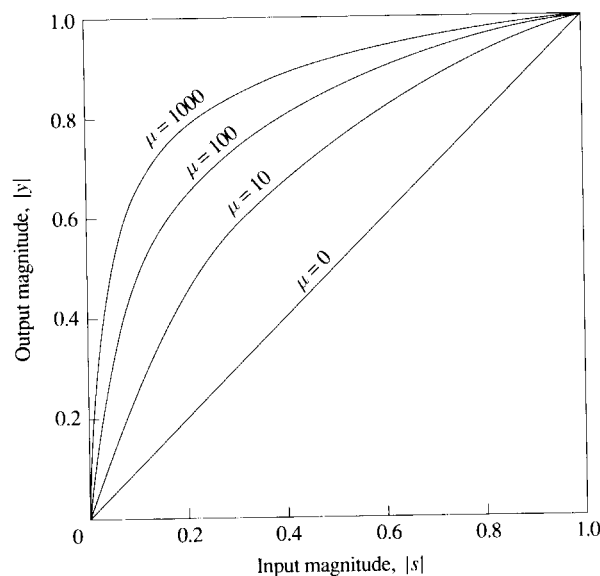


Figura 2.1: Magnitud de entrada-salida característica de un compresor logarítmico

Para un valor de $\mu=255$ dará como resultado una reducción en la potencia del ruido de cuantización de aproximadamente 24-dB relativo a la cuantización uniforme. Consecuentemente, un cuantizador de 7-bit usado en conjunción con un compresor

logarítmico de $\mu = 255$, produce un potencial ruido de cuantización de aproximadamente -77 dB comparado con los -53 dB para una cuantización uniforme.

Un compresor logarítmico usado en Europa llamado *A-law* tiene la forma:

$$|y| = \frac{1 + \log A|n|}{1 + \log A}, \quad [2.6]$$

donde A es elegido como 87.56. Aunque [2.5] y [2.6] son funciones no lineales diferentes, las dos tienen una compresión característica muy similar. La *Figura 2.2* ilustra estas dos funciones de compresión.

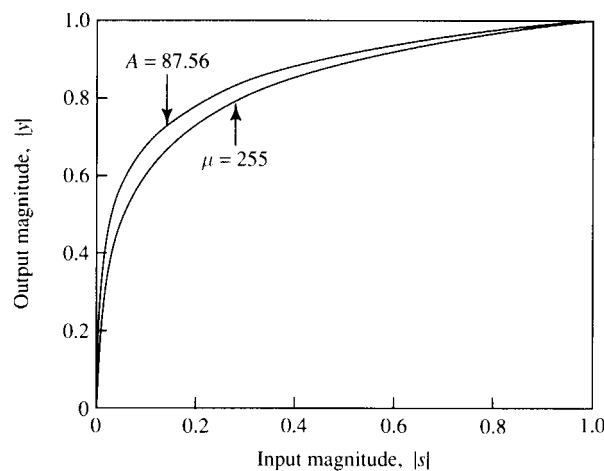


Figura 2.2: Comparación de los compresores logarítmicos μ -law y *A-law*

A un rango de muestreo de 8000 muestras por segundo y 8 bits por muestra (bpm), la tasa de bits resultante para una codificación PCM es de 64.000 bps. Sin el compresor logarítmico, un cuantizador uniforme requiere aproximadamente 12 bpm para obtener el mismo nivel de fidelidad.

En la reconstrucción de la señal a partir de los valores cuantizados, la relación logarítmica inversa es usada para expandir la amplitud de la señal.

2.2.2 Codificación por modulación de pulsos diferencial(DPCM)

En PCM, cada muestra es codificada independientemente. Sin embargo, muchas señales incluidas en el muestreo de voz que cumplan con el teorema del muestreo o lo excedan exhiben una significativa correlación entre muestras sucesivas, es decir el cambio promedio entre muestras sucesivas es relativamente pequeño. Consecuentemente un esquema de codificación que emplee la redundancia de las muestras resultará en una tasa de bits baja para una señal de salida.

Una solución relativamente simple es codificar la diferencia entre sucesivas muestras antes que las muestras en si mismas. La técnica resultante es llamada *codificación por modulación de pulsos diferencial* (DPCM). Dado que las diferencias entre las muestras sucesivas se esperan que sean más pequeñas que las amplitudes de las

muestras, menos bits serán requeridos para representar las diferencias. En este caso se cuantiza la secuencia de diferencias

$$e(n) = s(n) - s(n-1) \quad [2.7]$$

Nótese que el proceso de diferenciación es un simple intento de remover la correlación (redundancia) de la secuencia de voz.

Un refinamiento natural de esta aproximación general es predecir la actual muestra basándose en las M muestras anteriores.

Presumiblemente, este proceso de decorrelación tendrá consecuencias benéficas para el proceso de cuantificación.

Sea $s(n)$ la muestra actual y $\tilde{s}(n)$ el valor pronosticado de $s(n)$, definido como:

$$\tilde{s}(n) = \sum_{i=1}^M \bar{a}(i) s(n-i) \quad [2.8]$$

Así $\tilde{s}(n)$ es una combinación lineal pesada de las M muestras anteriores y de los coeficientes $\hat{a}(n)$ del filtro de predicción lineal (PL).

El error es la diferencia:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{i=1}^M \bar{a}(i) s(n-i) \quad [2.9]$$

El objetivo de la predicción lineal no es hacer una excelente tarea de predicción muestra por muestra, sino remover la correlación en un sentido extenso para una cuantización más eficiente. También los parámetros del PL pueden ser computados sobre una larga variedad de datos de voz y pueden ser construidos en el cuantizador como parámetros estáticos.

El diagrama de bloques de un sistema DPCM práctico es:

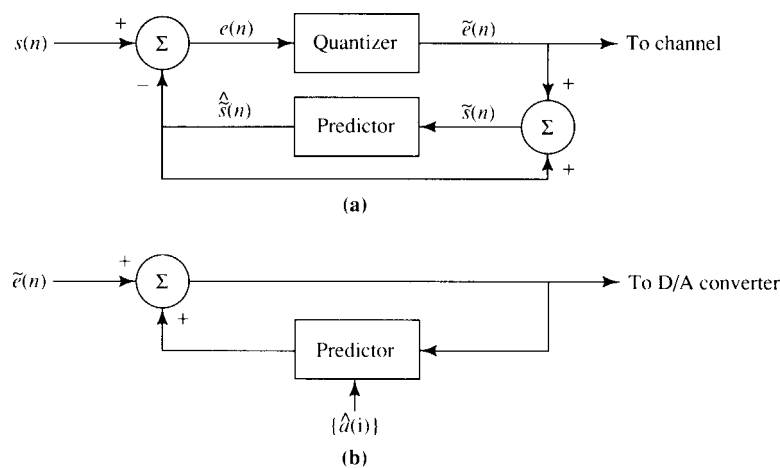


Figura 2.3- (a) diagrama de bloques de un codificador DPCM, (b) decodificador DPCM en el receptor

En esta configuración el predictor es implementado con retroalimentación iterativa alrededor del cuantizador. $\tilde{s}(n)$ denota la entrada al predictor y representa la

muestra $s(n)$ de la señal modificada por el proceso de cuantización, y la salida del predictor es:

$$\bar{s}(n) = \sum_{i=1}^M \bar{a}(i)\tilde{s}(n-i) \quad [2.10]$$

la diferencia:

$$e(n) = s(n) - \bar{s}(n) \quad [2.11]$$

es la entrada para el cuantizador y $\tilde{e}(n)$ denota la salida. Cada valor del error de predicción cuantizado es codificado en una secuencia de dígitos binarios y transmitido sobre el canal al receptor. El error cuantizado $\tilde{e}(n)$ es también sumado al valor predicho de $\bar{s}(n)$ para obtener un nuevo $\tilde{s}(n)$.

En el receptor el mismo predictor que fue usado en el final de la transmisión es sintetizado y su salida $\bar{s}(n)$ es sumado a $\tilde{e}(n)$ para producir $\tilde{s}(n)$. La señal $\tilde{s}(n)$ es la excitación deseada para el predictor y también la secuencia de la salida deseada desde la cual la señal reconstruida es obtenida por filtrado, como muestra la *Figura 2.3 (b)*.

Usando un compresor logarítmico y un cuantizador de 4-bit para la secuencia del error de cuantización $e(n)$, DPCM resulta en una señal de voz de alta calidad a una tasa de 32.000 bps.

2.2.3 Codificación por modulación de pulsos adaptativo (APCM) y codificación por modulación de pulsos diferencial adaptativo (ADPCM).

Dada la naturaleza no estacionaria de las señales de voz, la amplitud de la señal puede variar dependiendo del hablador y de si el segmento es vocalizado o no, se puede utilizar otro tipo de cuantizador que aproxime estas características para obtener una mejor relación de bits en la codificación. La cuantización no uniforme permite esta aproximación. La cuantización no uniforme puede realizarse adaptando las propiedades del cuantizador al nivel de la señal de entrada. Este tipo de cuantizadores se denominan cuantizadores adaptativos. Cuando la cuantización adaptativa se usa en PCM surge el APCM y cuando se usa en DPCM surge el ADPCM.

La idea básica del cuantizador adaptativo es que el tamaño de paso varíe de acuerdo a la varianza de la señal de entrada. Esto es, el tamaño de paso se incrementa o decrementa de acuerdo a como se incrementa o decrementa la varianza de la señal de entrada [Deller, 1993].

2.2.4 Modulación delta (DM)

La modulación delta puede ser vista como una forma simplificada del DPCM en el cual un cuantizador de dos niveles es usado en conjunción con un predictor de primer orden determinado.

El diagrama de bloques de la siguiente figura muestra un codificador y un decodificador DM

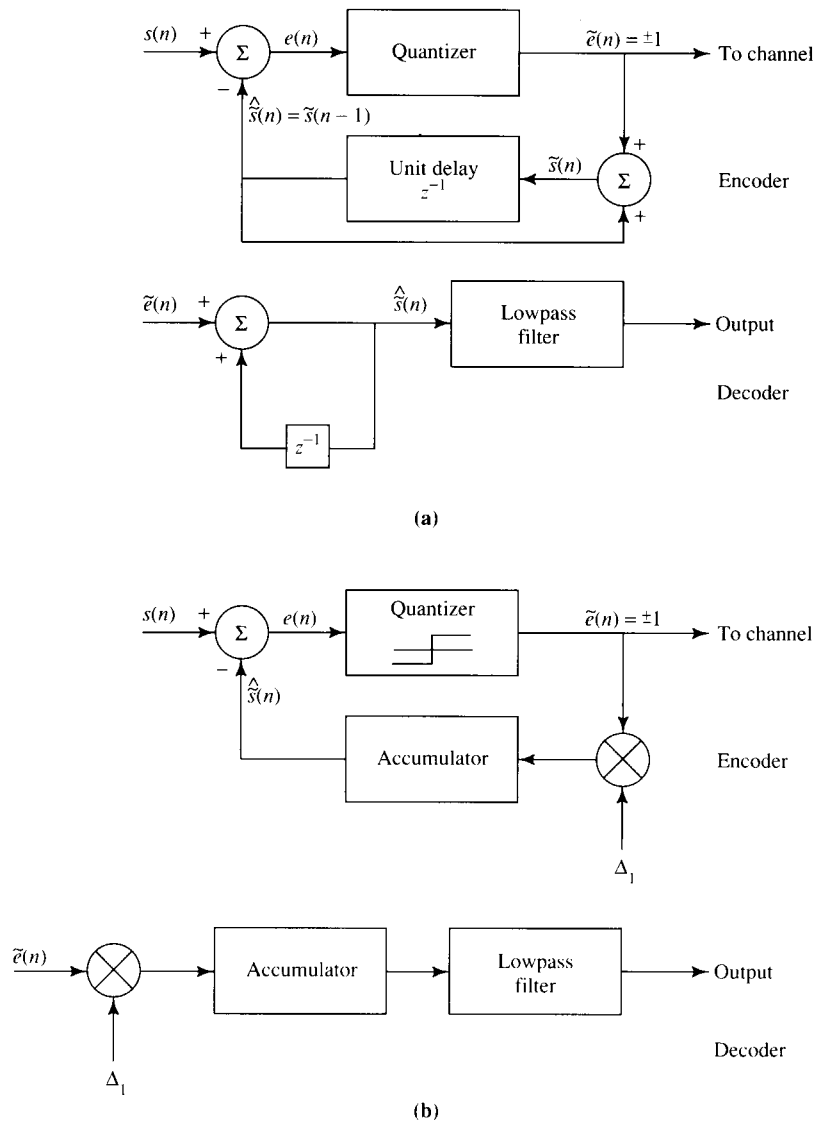


Fig. 2.4.- (a) diagrama de bloques de sistema DM.
 (b) Realización equivalente de un sistema DM

2.2.5 Modulación delta adaptativa (ADM)

Bajo cualquier rango de muestreo dado, la performance del codificador DM esta limitada por dos tipos de distorsiones, como se ve en la siguiente figura:

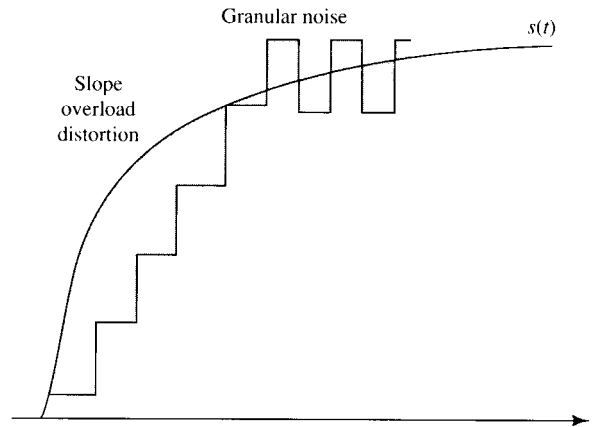


Figura 2.5:-Ejemplo de un distorsión slope overload y de una distorsión granular noise en un codificador DM

La distorsión *slope overload* se debe al uso de un tamaño de paso demasiado pequeño cuando la forma de onda tiene una fuerte pendiente. La distorsión llamada *granular noise* resulta de usar un tamaño de paso que es muy grande donde la forma de onda tiene una pendiente pequeña. La necesidad de minimizar ambas distorsiones es conflictivo para la selección del tamaño del paso Δ_1 . Una solución es seleccionar Δ_1 para minimizar la suma de los valores cuadrados promedios de las dos distorsiones.

Una solución alternativa es usar un tamaño de paso variable que se adapte a las características de la señal fuente. Esto es, el tamaño del paso es incrementado cuando la forma de onda tiene una fuerte pendiente y decrementado cuando la señal tiene una pendiente relativamente pequeña.

Esta característica adaptativa es ilustrada en la siguiente figura:

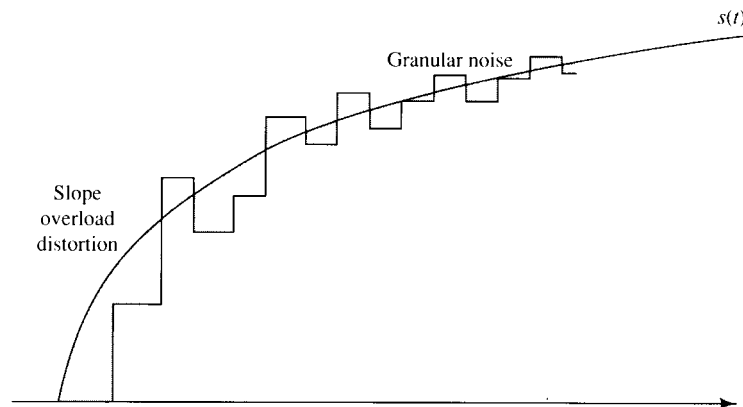


Figura 2.6:- Ejemplo de un código DM de paso variable.

La secuencia del error cuantizado $\tilde{e}(n)$ provee una buena indicación de las características de caída de la forma de onda que esta siendo codificada. Cuando el signo del error de cuantización cambia indica que la caída en la localización de la iteración es relativamente pequeña. Por otro lado cuando la caída es pronunciada, sucesivos valores del error serán esperados con el mismo signo. De este modo es posible imaginar algoritmos que decremen o incrementen el tamaño del paso dependiendo de sucesivos valores del error $\tilde{e}(n)$.

Una regla relativamente simple pensada por Jayant es variar adaptativamente el tamaño del paso acorde a la relación

$$\Delta(n) = \Delta(n-1)K^{\tilde{e}(n)\tilde{e}(n-1)}, \quad n = 1, 2, \dots \quad [2.12]$$

Donde $K \geq 1$ es una constante que es seleccionada para minimizar la distorsión total.

2.3 Códigos de forma de onda en dominio frecuencial.

En esta sección se considerarán los métodos de codificación de formas de onda que filtran la señal de voz en un número de bandas o sub-bandas de frecuencia y luego codifican separadamente la señal en cada sub-banda. La codificación puede ser en el dominio del tiempo o en el dominio frecuencial.

2.3.1 Codificación en Sub-Bandas (SBC)

A través de un banco de filtros se separa la señal de voz en N sub-bandas de frecuencia y se codifica la señal en cada sub-banda separadamente.

En la práctica, la señal es dividida en 4 a 8 sub-bandas y la señal en cada sub-banda se codifica por separado. APCM ha sido usado para codificar las sub-bandas. Dado que las bandas de bajas frecuencias contienen la mayoría de la energía espectral en señales de voz vocalizada, más bits son usados para las bandas bajas y menos bits para las altas.

2.3.2 Codificación Adaptativa por Transformadas (ATC)

En ATC la señal de voz analógica es muestreada, subdividida en frames de N muestras, transformada al dominio espectral para su codificación, y transmitida o almacenada. En el decodificador cada bloque de muestras espectrales es antitransformado al dominio del tiempo y la señal es sintetizada. Para alcanzar un código eficiente asignamos más bits a los coeficientes espectrales más importantes y menos bits a los menos importantes. Además, usando una alocaión dinámica en la asignación del número total de bits, podemos adaptar los cambios estadísticos de las señales de voz.

Un objetivo al seleccionar la transformación desde el dominio del tiempo al dominio frecuencial es alcanzar muestras espectrales decorrelacionadas. En este sentido la transformada de Karhunen-Loève (KLT) es óptima dado que produce valores espectrales que son decorrelacionados. Sin embargo, esta transformada es computacionalmente compleja. Para un frame de N puntos, $O(N^2)$ es necesario para computar la transformada KLT. Además, la matriz de autocorrelación entera del frame debería ser enviada al receptor, como información adicional para que la transformada pueda ser invertida. La Transformada de Fourier Discreta (DFT) y la Transformada Discreta Coseno (DCT) son alternativas viables aunque no sean óptimas. De estas dos la

DCT produce buena performance comparado con la KLT y es usada generalmente en la práctica.

2.4 Vocoders

Las técnicas de codificación descritas en la sección 2.3 están basadas en una representación de señales de voz muestra por muestra o frame por frame, ya sea en el dominio del tiempo o frecuencial. En contraste, los distintos tipos de vocoders están basados en la representación de una señal de voz por un modelo *solo-polos* del sistema. El modelo se muestra en la *Figura 2.7*.

En este modelo el sistema de producción de voz está modelado como un filtro *solo-polos*. Para voz vocalizada, la excitación es un tren de impulsos periódicos con un periodo igual al período tono fundamental de la voz. Para voz no vocalizada, la excitación es una secuencia de ruido blanco. Además, puede haber un parámetro de ganancia estimada incluida en el modelo. Básicamente, los diferentes vocoders estiman los parámetros del modelo desde los frames de la señal de voz (análisis de la voz), codifican y transmiten estos parámetros al receptor sobre una base frame-por-frame, y reconstruye la señal de voz desde el modelo en el receptor (síntesis de voz).

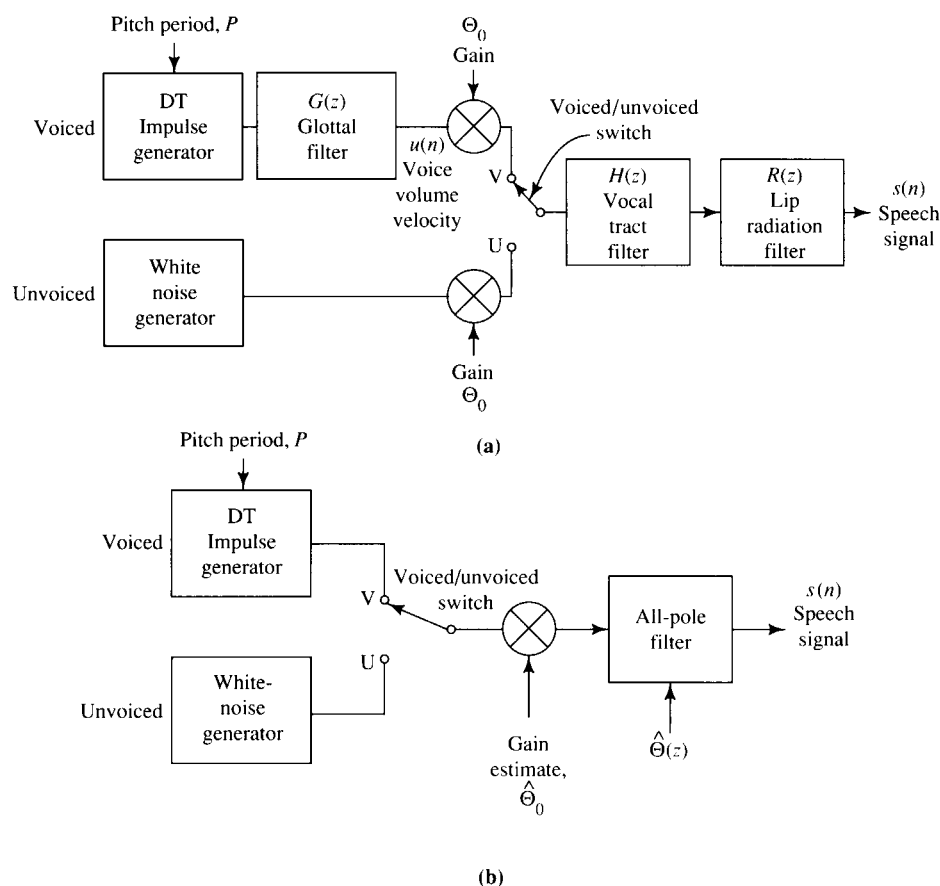


Figura 2.7: modelo de producción de voz

Los vocoders usualmente proveen una mayor compresión que la que se puede lograr con las otras técnicas de codificación. En particular, las otras clases de vocoders

permiten obtener una calidad de comunicación de voz en el rango de 2400 a 9600 bps. Los distintos tipos de vocoders incluyen el vocoder de canal, el vocoder cepstral (homomórfico), el vocoder fase, el vocoder formante y el codificador predictivo lineal, el cual es el más usado actualmente [Deller, 1993].

Capítulo 3

CODIFICACION EN SUBBANDAS

CODIFICACION EN SUB-BANDAS

3.1 Introducción

En las clases de algoritmos de codificación en el *dominio del tiempo*, la señal de entrada es tratada como una señal simple de banda-completa; y en los codificadores predictivos, la redundancia es removida a priori a la codificación por predicción y filtrado inverso. Las mayores diferencias entre estos algoritmos están determinadas por el grado de predicción que estos emplean, y por si los esquemas son adaptativos o no.

Otras clases de *algoritmos* de codificación son aquellos que dividen la señal de entrada en un número arbitrario de componentes de frecuencia y codifican cada una de estas separadamente. Esta división en componentes de frecuencia remueve la redundancia en la entrada y provee un conjunto de entradas decorrelacionadas para el canal. Recordemos que la acción de un codificador DPCM es también similar, si no idéntica. Los codificadores en este caso, cuando tratan con una señal redundante, sacan una secuencia de componentes de error de predicción que tienden a ser decorrelacionadas. Las técnicas de *codificación en el dominio frecuencial* tienen la ventaja que el número de bits usados para codificar cada componente de frecuencia puede ser variable, de modo que la precisión de codificación es siempre puesta donde es necesario en el dominio frecuencial. De hecho, las bandas con poca o sin energía pueden no ser codificadas en su totalidad. La alocaión de bits variable puede en principio proveer formas arbitrarias de ruidos.

Como en el caso de técnicas de codificación en el dominio temporal existen una gran variedad de algoritmos en el dominio frecuencial, desde simples a complejos. Las mayores diferencias entre estos algoritmos están determinadas por la forma en que son modeladas las estadísticas de la fuente, y por el grado en que es explotada su redundancia. Dos ejemplos de técnicas de codificación en el dominio frecuencial son: Codificación en Sub-Bandas (SBC) y Codificación por Transformadas (TC) .

En la codificación en sub-bandas la banda de voz es dividida en cuatro o más sub-bandas por un banco de filtros pasa-bandas. Cada sub-banda es, en efecto, trasladada pasa-bajos a frecuencia cero por un proceso de modulación. Cada sub-banda es entonces muestreada (o remuestreada) a su rango de Nyquist (el doble del ancho de la banda) y codificada digitalmente con un codificador PCM o DPCM . En este proceso cada sub-banda puede ser codificada de acuerdo a criterios de percepción que son específicos a cada banda. En la reconstrucción las señales sub-bandas son decodificadas y remoduladas a sus localizaciones originales; luego son sumadas obteniéndose una réplica de la señal de voz original.

La codificación en sub-bandas ofrece varias ventajas. Alocaando apropiadamente la cantidad de bits en las diferentes bandas y el número de niveles de cuantificación, se puede controlar separadamente la varianza del error de reconstrucción en cada banda y la forma del espectro del error de reconstrucción en función de las frecuencias. En las bandas de frecuencias más bajas, donde el tono fundamental y la estructura formante debe ser preservada con precisión, un gran número de bits por muestra puede ser usado; en cambio en las bandas de frecuencias más altas, donde la fricativa y los sonidos semejantes a ruidos ocurren en la voz, menos bits por muestra pueden ser usados.

La parte más compleja del codificador es el banco de filtros. Con las nuevas tecnologías en filtros tal como filtros CCD y filtros digitales, esta complejidad ha sido rápidamente reducida. También las técnicas de diseño de filtros espejo en cuadratura (QMF) aportan distintas ventajas en la implementación digital de este codificador.

La *Figura 3.1* ilustra un diagrama en bloques básico del codificador sub-banda. El codificador consiste de un banco de M filtros pasa-bandas, seguido por sus respectivos codificadores sub-bandas los cuales son codificadores PCM, y un multiplexor. El receptor tiene los estados inversos de demultiplexión, decodificación y filtrado pasa-banda a priori a la suma de las sub-bandas. El codificador sub-banda en la *Figura 3.1* transmite formas de ondas en tiempo individuales $x_k(t)$; $k=1,2,\dots,M$ y el receptor suma versiones decodificadas $y_k(t)$ sincronizadas en fase para obtener $y(t)$. El codificador sub-banda es por lo tanto un codificador preservador de formas de onda.

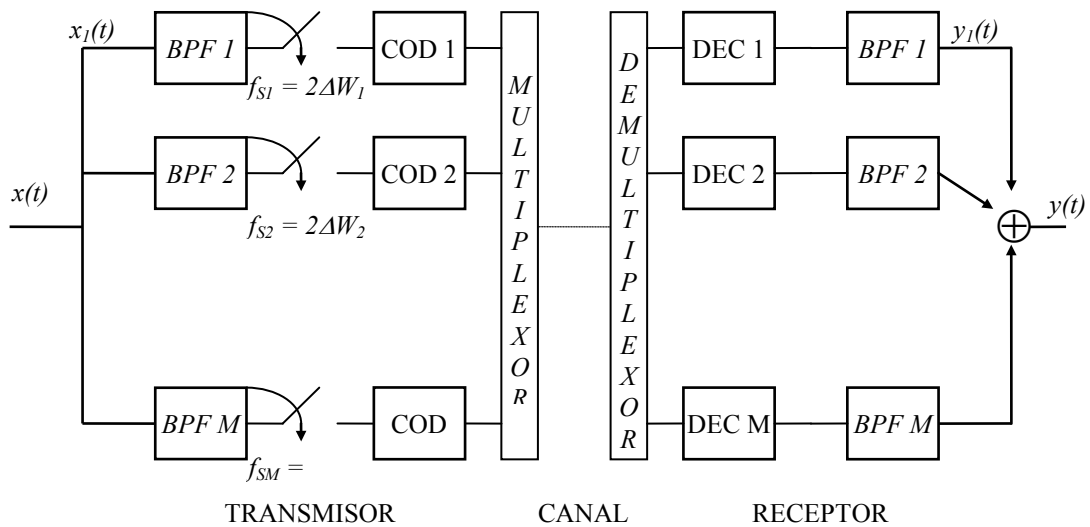


Figura 3.1: Diagrama en Bloques de codificación en sub-bandas (SBC)

En la *Figura 3.1*, el ancho de cada sub-banda ΔW_k será una función de el número de sub-bandas k , implicando sub-bandas de *ancho variable*. El caso especial de sub-bandas de *igual ancho* es importante para la implementación así como también para tratarlo analíticamente. Ambos tipos de división serán considerados por la codificación en sub-bandas. La *Figura 3.2* ilustra las dos clases de división en sub-bandas para el ejemplo de $M=4$. Las regiones pintadas definen el número de sub-banda para $k=3$.

En el caso de sub-bandas de igual ancho:

$$\begin{aligned} \Delta W_k &= \Delta W = W/M; & k &= 1, 2, \dots, M \\ \Delta \Omega &= 2\pi \Delta W = \Omega_w/M = 2\pi W/M \end{aligned} \quad [3.1]$$

donde W y Ω_w representan el ancho de banda total de la entrada en Hz y radianes por segundo, respectivamente. En el caso de sub-bandas de igual ancho, el ancho de éstas está determinado por el crecimiento de k :

$$\Delta W_{k+1} > \Delta W_k; \quad k = 1, 2, \dots, M - 1 \quad [3.2]$$

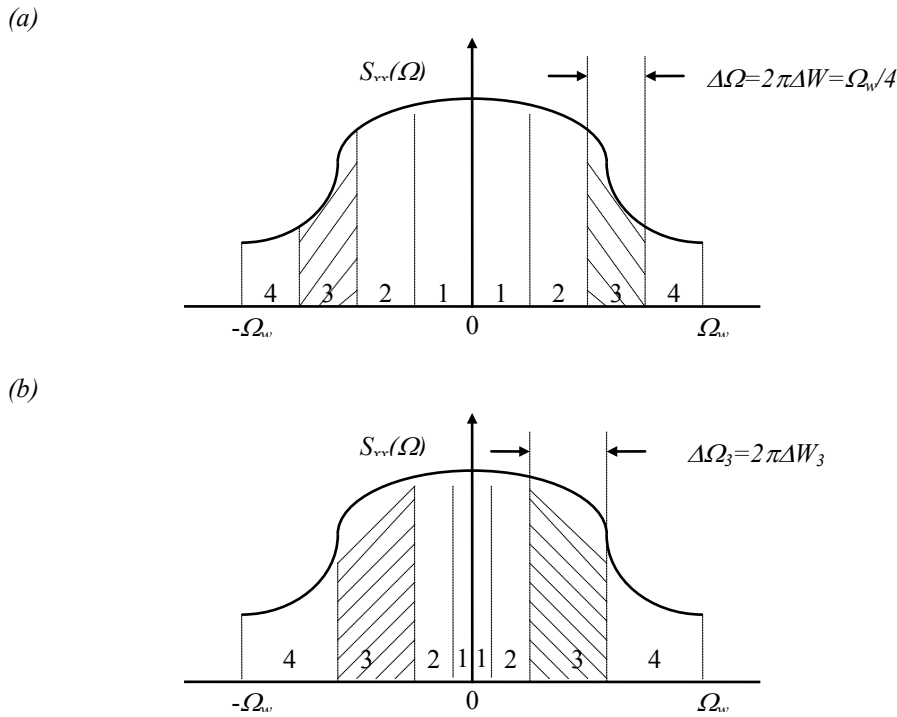


Figura 3.2: División del espectro de entrada en $M=4$ sub-bandas de ancho constante (a) y variable (b)

El diseño en [3.1] sugiere que las frecuencias bajas en la señal de voz son mejor aisladas u observadas que las frecuencias altas.

La *Figura 3.3* muestra respuestas de amplitud de una banco de filtros que pueden ser apropiadas para realizar la separación (splitting) de la banda como muestra la *Figura 3.2*. Las observaciones a ser hechas serán también válidas para los bancos de filtros digitales de la *Sección 3.2*, con $H(j\Omega)$ y Ω_w reemplazados por $H(e^{j\omega})$ y π , respectivamente. Una distinción importante en la *Figura 3.2* esta dada entre (a) filtros de igual ancho y (b) filtros de ancho variable. Otra distinción importante es entre filtros que overlapean como en (a) y que no son contiguos como en (b). La situación de filtros exactamente contiguos es académica dado que las implementaciones prácticas involucran respuestas de amplitud con características roll-off finitas.

La aproximación a la *Figura 3.3(b)* requiere roll-offs de filtros extremadamente rápidos que minimicen los gaps inter-bandas. Esto ofrece la posibilidad de reducir los rangos de muestreo (valores chicos de f_{sk}) y por lo tanto un menor rango de bits de codificación para valores dados de R_k . Los gaps de frecuencias inter-bandas serán distintos de cero en los diseños de filtros prácticos, causando una calidad reverberante en la señal de voz de salida cuando se codifica con un rango bajo de bits, a menos que las sub-bandas puedan adaptativamente tomar regiones de energía de la señal de voz significativa, tal como frecuencias formantes en voz vocalizada.

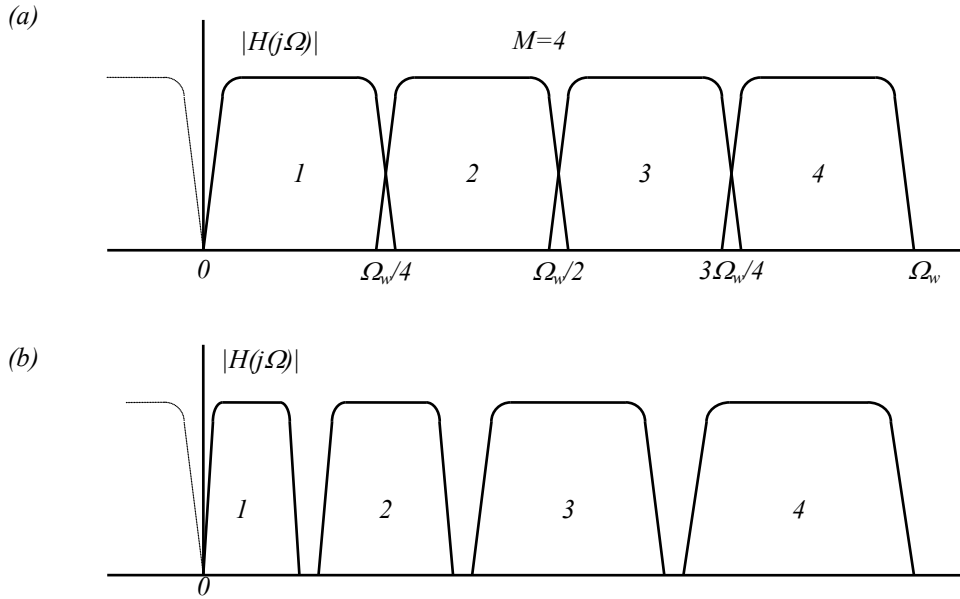


Figura 3.3: Respuestas de amplitud en bancos de filtros consistiendo de cuatro pasa-bandas de igual ancho (a) y de ancho distinto (b)

3.2 Banco de Filtros de Banda Entera

Una característica importante en la Figura 3.1 es que los cutoffs (cortes) de los filtros pasa-bandas pueden elegirse tal que cada banda pueda muestrearse al doble del ancho de banda correspondiente:

$$f_{sk} = 2\Delta W_k; \quad k = 1, 2, \dots, M \quad [3.3]$$

más bien que al doble de la frecuencia más alta presente en la banda completa de la señal.

Sea el ancho de una sub-banda escrita en la forma

$$\Delta W_k = \frac{W}{\zeta_k}; \quad k = 1, 2, \dots, M \quad [3.4]$$

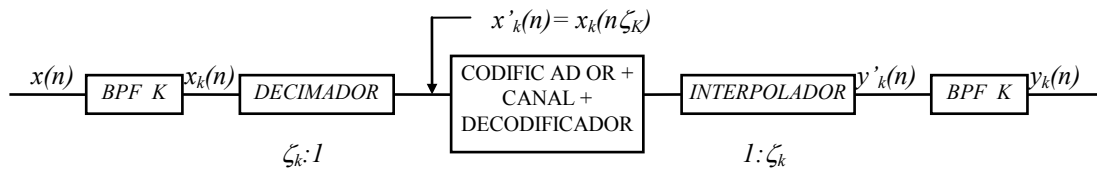
$\zeta_k=M$ para todo k con sub-bandas de igual ancho. Con la partición en sub-bandas de las Figuras 3.2(a) y 3.3(a), $\zeta_k=4$ para todo k .

La Figura 3.4 muestra la secuencia de operaciones de filtrado y codificación en SBC, usando el ejemplo genérico para una sub-banda k en la Figura 3.4(a) y el caso especial de $k=3$ y $\zeta_k=4$ en la Figura 3.4(b).

El decimador $\zeta_k:1$ sub-muestra la salida pasa-banda $x_k(n)$ por un factor ζ_k , implicando un rango de muestreo de $f_{sk}=2W/\zeta_k$ para la sub-banda k . Esta decimación implica un rango de repetición de su espectro que es más grande que el espectro de la banda-completa por un factor de ζ_k . Como resultado de esto, los ejes de las x en la Figura 3.4(b) difieren por un factor de $\zeta_k=4$. Una de las repeticiones del espectro será en la banda-base, de modo que la decimación efectivamente traslada la frecuencias más baja de la señal pasa-banda a frecuencia cero. El interpolador $1:\zeta_k$ llena con $(\zeta_k - 1)$ ceros entre cada par de muestras de la señal pasa-bajo. El k -esimo armónico de la banda-

base interpolada es así efectivamente trasladado pasa-banda a la región pasa-banda inicial apropiada. Los procesos de modulación explícitamente mencionado en la Sección 3.1 son por lo tanto reemplazados por procesos discretos en tiempo simples de decimación e interpolación. Se asume que el proceso de interpolación incluye un factor de escaleo de amplitud de ζ_k . Esto mantiene los valores originales de la varianza a pesar de los valores de amplitud cero que son introducidos en el proceso de interpolación.

(a)



(b)

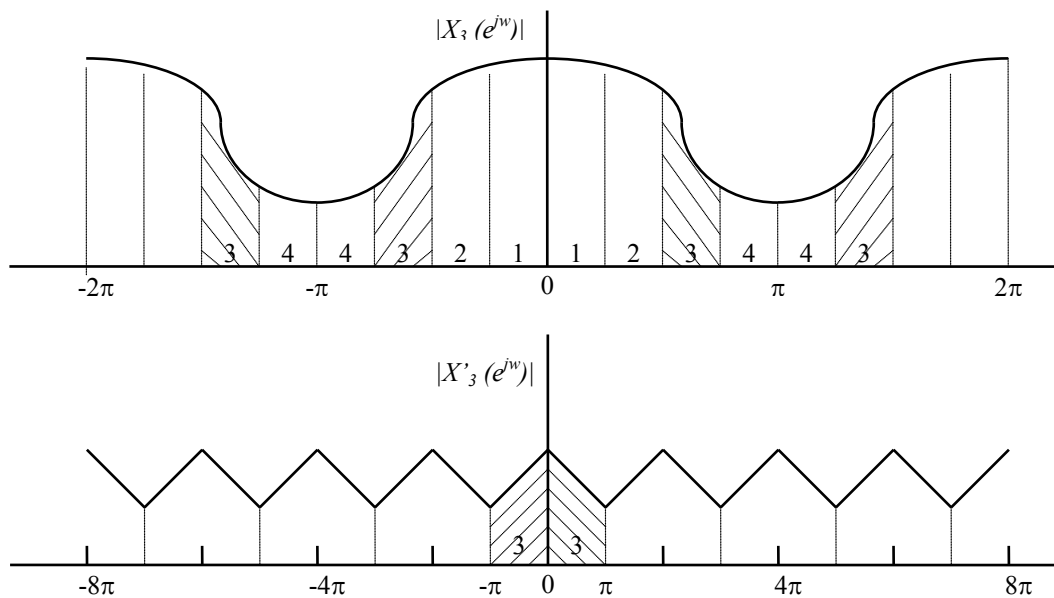


Figura 3.4: Realización del muestreo de bandas enteras con una entrada discreta: (a) diagrama en bloque de codificación SBC para la sub-banda k ; y (b) espectro original y espectro re-muestreado después de la decimación, para la sub-banda $k=3$, con $\zeta_3=4$. El espectro de la banda base resultante de la decimación es movido hacia atrás al rango de frecuencia original de la sub-banda k después de la interpolación por un factor ζ_k .

La amplitud espectral $|X_3(e^{jw})|$ y $|X'_3(e^{jw})|$ en la Figura 3.4(b) se refiere a la sub-banda $k=3$, con ancho $\zeta_k=4$. Note que el espectro de la secuencia decimada tiene su propio escala de frecuencias. Si el procedimiento de los últimos dos párrafos es reemplazado por un número par de sub-bandas ($k=2$ o $k=4$), puede verse que el espectro obtenido es invertido en el proceso de traslación pasa-bajo a la banda-base. Esto es, sin embargo, neutralizado por una subsiguiente inversión en el proceso de interpolación por el valor par k .

3.3 Bancos de Filtros Espejo en Cuadratura

La situación de sub-bandas overlapeadas en la *Figura 3.3(a)* sugiere que los efectos de *aliasing* pueden ocurrir si las sub-bandas son muestreadas a $f_{sk}=2W/M=\Omega_w/\pi M$. Este problema es elegantemente tratado por los bancos de filtros espejo en cuadratura (QMFB), como se muestra en la *Figura 3.5*. Esta figura muestra la división de la banda-completa de la señal, de frecuencia máxima en radianes π , en dos de igual ancho usando un par restringido de filtros pasa-bajos y pasa-altos. En la notación de [3.4], $\zeta_1=\zeta_2=M=2$. Por repetidas sub-divisiones de las sub-bandas, usando bancos de filtros QMF, se puede realizar un banco de filtros SBC con M dado por una potencia de 2, tal como $M=4$ como se muestra en la *Figura 3.3(a)*. Los valores de M que no son potencias de 2 pueden también ser realizados simplemente ignorando las ramas de la sub-bandas apropiadas en el árbol QMF.

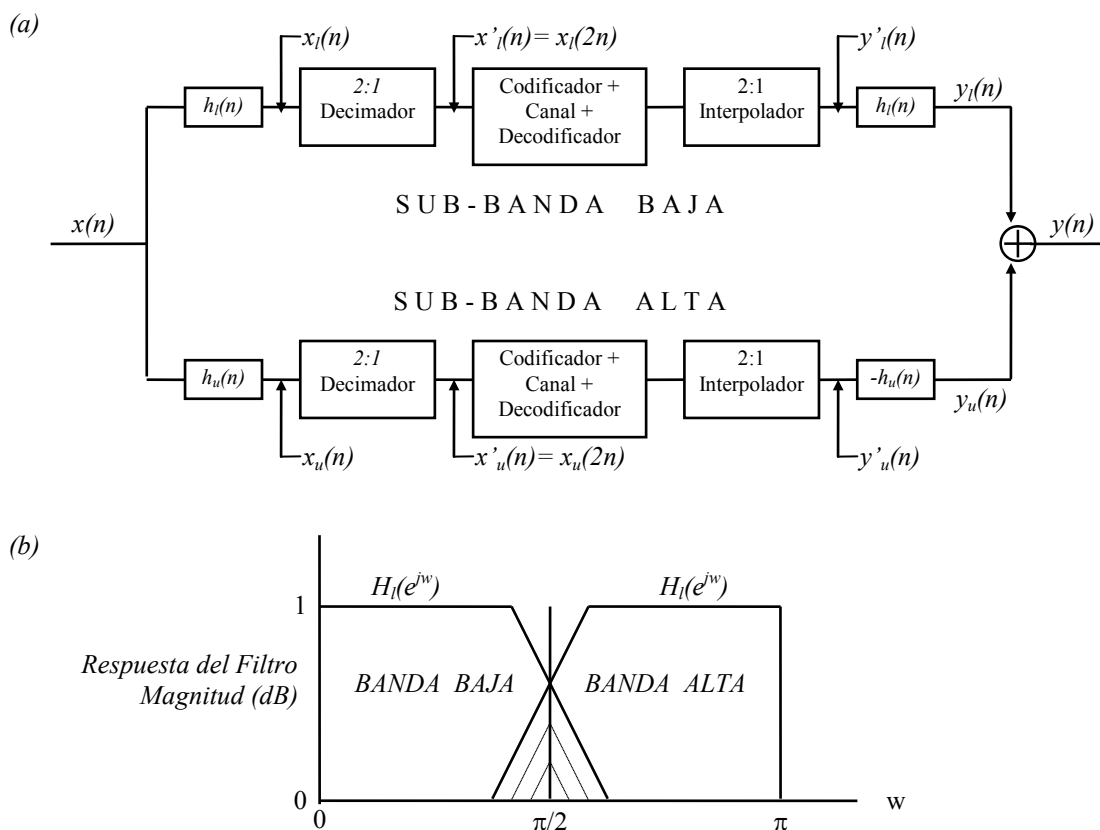


Figura 3.5: Filtrado espejo en cuadratura para dividir una entrada en dos sub-bandas de igual ancho: (a) implementación; y (b) ilustración cualitativa de un banco de la respuesta de un banco de filtros que provee cancelación de imagen de aliasing.

El resto de la siguiente discusión se refiere al primer estado de tal banco de filtros, involucrando dos sub-bandas como muestra la *Figura 3.5*. Cuando más estados de particionamiento de bandas son introducidos, cada una de las ramas en la *Figura 3.5(a)* se dividirá en más ramas, y la frecuencia de muestreo será reducida por un factor de dos en cada estado. El resultado de la *Figura 3.5(b)* se aplicará repetidamente con redefiniciones apropiadas de las frecuencias absolutas representadas por $0, \pi$ y $\pi/2$.

Cada una de las señales sub-bandas $x_l(n)$ y $x_u(n)$ es re-muestreada por un factor 2:1. Esta reducción del rango de muestreo es necesaria para mantener una tasa mínima

total de bits en la codificación de esta señal. Esta reducción del rango de muestreo introduce términos de *aliasing* en cada una de las señales sub-banda dado el rango finito de roll-off en la respuesta del filtro. Por ejemplo, en la banda baja la energía de la señal en el rango de frecuencias superior a $\pi/2$ es plegado hacia abajo en el rango de 0 a $\pi/2$ y aparece como *distorsión de aliasing* en esta señal, en el rango de frecuencias compuesto por la región sombreada en la mitad izquierda de la *Figura 3.5(b)*. El *aliasing* también ocurre en la banda alta en un sentido similar; toda energía de la señal en el rango de frecuencias inferior a $\pi/2$, es plegada hacia arriba en su banda Nyquist $\pi/2$ a π , esto causa *aliasing* en el rango de frecuencias compuesto por el área sombreada en la mitad derecha de la *Figura 3.5(b)*. Este aliasing mutuo de la energía de la señal entre las sub-bandas baja y alta es a veces denominado *leakage interband*. La cantidad de leakage que exista entre sub-bandas es directamente dependiente del grado con el que los filtros $h_l(n)$ y $h_u(n)$ aproximen a filtros pasa-bajos y pasa-altos ideales, respectivamente.

En el proceso de reconstrucción, los rangos de muestreo de las sub-bandas son incrementados por un factor de $1:2$ poniendo valores cero entre cada par de muestras. Esto introduce una repetición periódica del espectro de la señal en las sub-bandas. Por ejemplo, en la banda baja la energía de la señal desde 0 a $\pi/2$ está simétricamente plegada cerca de la frecuencia $\pi/2$ en el rango de la banda alta. Esta energía de la señal no deseada, referida como *imagen* es principalmente filtrada fuera por el filtro pasa-bajos $h_l(n)$ en el receptor. Esta operación de filtrado efectivamente interpola los valores ceros insertados entre cada par de muestras de las señales sub-bandas que representan apropiadamente la señal deseada [Crochiere y Rabiner, 1983]. Similarmente, en la sub-banda alta una imagen es reflejada en la sub-banda baja y filtrada fuera por el filtro $-h_u(n)$.

El grado con el cual las imágenes de arriba son removidas por los filtros $h_l(n)$ y $-h_u(n)$ esta determinado por el grado con el cual ellos aproximen a filtros pasa-bajos y pasa-altos ideales. Dada la relación especial de las señales sub-bandas en el banco de filtros QMF, las componentes restantes de las imágenes pueden ser canceladas por términos de *aliasing* introducidos en el análisis. Esta cancelación ocurre después de sumar las dos señales sub-bandas interpoladas $y_l(n)$ y $y_u(n)$, y la cancelación es exacta en ausencia de errores de codificación. En presencia de codificación, esta cancelación es obtenida en el nivel de ruido de cuantificación.

Una manera de obtener esta cancelación apropiadamente en el banco de filtros QMF es usar filtros $h_l(n)$ y $h_u(n)$, los cuales son *filtros de respuesta al impulso finita* (FIR), simétrico y antisimétrico, respectivamente; diseñados con un número par de elementos:

$$h_l(n)=h_u(n)=0 \quad \text{para } 0 > n \geq N; \quad [3.5]$$

$$h_l(n)= h_l(N-1-n), \quad n=0,1,\dots,N/2-1 \quad [3.6.a]$$

$$h_u(n)= -h_u(N-1-n), \quad n=0,1,\dots,N/2-1 \quad [3.6.b]$$

La cancelación de los efectos de *aliasing* en el banco de filtros QMF requiere que los filtros en la *Figura 3.5(a)* satisfagan la condición:

$$h_u(n)= (-1)^n h_l(n), \quad n=0,1,\dots,N-1 \quad [3.7]$$

esta es una relación de *imagen espejo* de los filtros, implicando simetría alrededor de $\pi/2$, como en la *Figura 3.5(b)*. Los coeficientes de los filtros son idénticos excepto que sus signos alternan. Por lo tanto, ambos filtros pueden ser obtenidos usando un filtro simple de N elementos, en primer medida.

Más aún, si deseamos que la salida del banco de filtros $y(n)$ sea una réplica retardada de la entrada $x(n)$ (en ausencia de errores de codificación), los filtros $h_l(n)$ y $h_u(n)$ deberán también satisfacer la condición:

$$\left|H_l(e^{j\omega})\right|^2 + \left|H_u(e^{j\omega})\right|^2 = 1 \quad [3.8]$$

donde $H_l(e^{j\omega})$ y $H_u(e^{j\omega})$ son las transformadas de Fourier de $h_l(n)$ y $h_u(n)$, respectivamente; esta es simplemente una condición para una característica *todo pasa*. Si la condición *todo pasa* es incluida en el diseño de imagen espejo [3.7], el punto de intersección de las funciones de los dos filtros en la *Figura 3.5(b)* será de -3 DB para cada función de transferencia.

El filtro requerido en [3.8] no puede obtenerse exactamente por los filtros de imagen espejo de [3.7] excepto cuando $N=2$ y cuando N tiende a infinito. Sin embargo, esto puede ser aproximado para valores modestos de N . El diseño de filtros que satisfacen [3.6] y [3.7] y aproximan la condición [3.8] puede obtenerse mediante la realización de un programa de optimización. Los coeficientes de los filtros resultantes para N valores en el rango 8 a 64 han sido tabulados.

La *Tabla 3.1* lista los coeficientes para $N=32$ y $N=16$. En un árbol QMF para $M=4$ sub-bandas, la primera sub-división puede usar filtros de orden $N=32$. En vista de la decimación de 2:1, se puede realizar un diseño que utilice en el segundo estado del árbol QMF filtros de $N=16$.

(a)

$N=32$

h(16) a h(23)	h(24) a h(31)
4.6645830E-01	1.7881950E-02
1.2846510E-01	-1.7219030E-04
-9.9800110E-02	-9.3636330E-03
-3.9244910E-02	1.4272050E-03
5.2909300E-02	4.1581240E-03
1.4468810E-02	-1.2601150E-03
-3.1155320E-02	-1.3508480E-03
-4.1094160E-03	6.5064660E-04

(b)

$N=16$	
h(8) a h(15)	
	.47211220E-00
	.11786660E-00
	-.99295500E-01
	-.26275600E-01
	.46476840E-01
	.19911500E-02
	-.20487510E-01
	-.65256660E-03

Tabla 3.1: QMF de $N=32$ y $N=16$

La *Figura 3.6* muestra la respuesta en frecuencia característica para un filtro de $N=32$ elementos. La *Figura 3.6(a)* muestra la magnitud de $H_l(e^{jw})$ y $H_u(e^{jw})$ expresada en dB como una función de w . Como en el esquema de la *Figura 5(b)*, obsérvese que las regiones roll-off de las dos respuestas interceptan en -3dB. La *Figura 3.6(b)* muestra la magnitud de la expresión $|H_l(e^{jw})|^2 + |H_u(e^{jw})|^2$ expresada en dB como una función de w . Como se puede ver el requerimiento de [3.8] es satisfecho dentro de ± 0.025 dB, lo cual es más que satisfactorio para una buena performance SBC. El error de reconstrucción en el diseño de 32 elementos de la *Tabla 3.1* es también ± 0.025 dB, pero la atenuación *stop-band* de los filtros (medido en el primer pico *stop-band*) es de 52 dB, lo cual es mucho mas grande que los 37 dB de atenuación en el ejemplo de la *Figura 3.6(b)*. En el ejemplo de 16 elementos de la *Tabla 3.1*, el error de reconstrucción es de ± 0.07 dB, y la atenuación *stop-band* de 30dB.

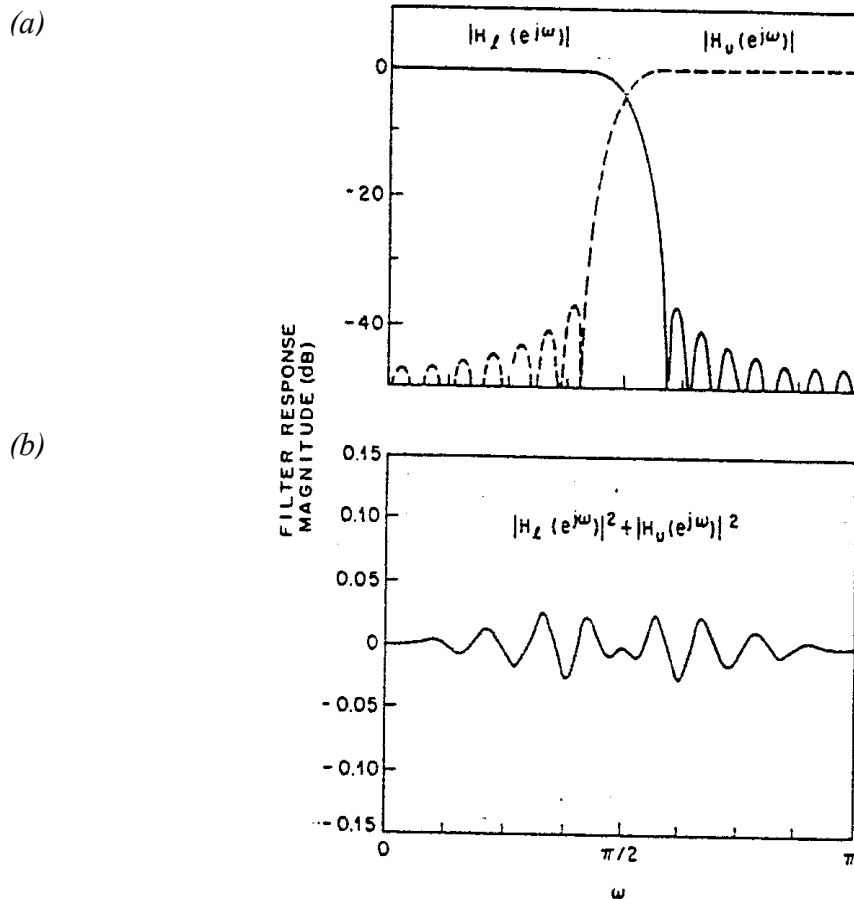


Figura 3.6: Ilustración de las respuestas de amplitud frecuencial en los bancos de filtros QMF usando filtros FIR: (a) características pasa-bajos y pasa-altos de los filtros FIR de 32 elementos; (b) características aproximadamente todo-pasa de la combinación.

El uso de filtros FIR tiene la ventaja de características de fase lineal lo cual elimina los problemas de distorsiones por retardo de grupo. Estas características también permiten que el diseño de 2-bandas de la *Figura 3.5* pueda ser convenientemente cascado sin la necesidad de compensación de fase. Sin embargo, los diseños efectivos de filtros FIR implican retardos significantes de codificación. Por ejemplo, con un diseño de 32 elementos, la codificación y decodificación se retarda debido a que el primer nivel del banco QMF lleva 4 mseg por cada uno, asumiendo una entrada muestreada a 8 KHz; y los niveles subsiguientes de la partición QMF introducen sus correspondientes retardos adicionales. De acuerdo a esto, para implementar sistemas SBC con valores más pequeños de retardo podríamos pensar en bancos QMFs basados en diseños de *respuesta de impulso infinita* (IIR). Estas propuestas incluyen procedimientos especiales para moderar las distorsiones de retardos de grupos inherentes en diseños IIR.

3.4 Codificación en Sub-Bandas de Señales de Voz

El siguiente ejemplo ilustra el diseño de sistemas SBC con asignación fija de bits para señales de voz a rangos de bits desde 16 a 32 kb/s. Los sistemas de 32 kb/s con asignación fija de bits pueden proveer una muy alta calidad subjetiva, midiendo con MOS estaría en el orden de 4.2, esta es una condición necesaria para *calidad toll* de

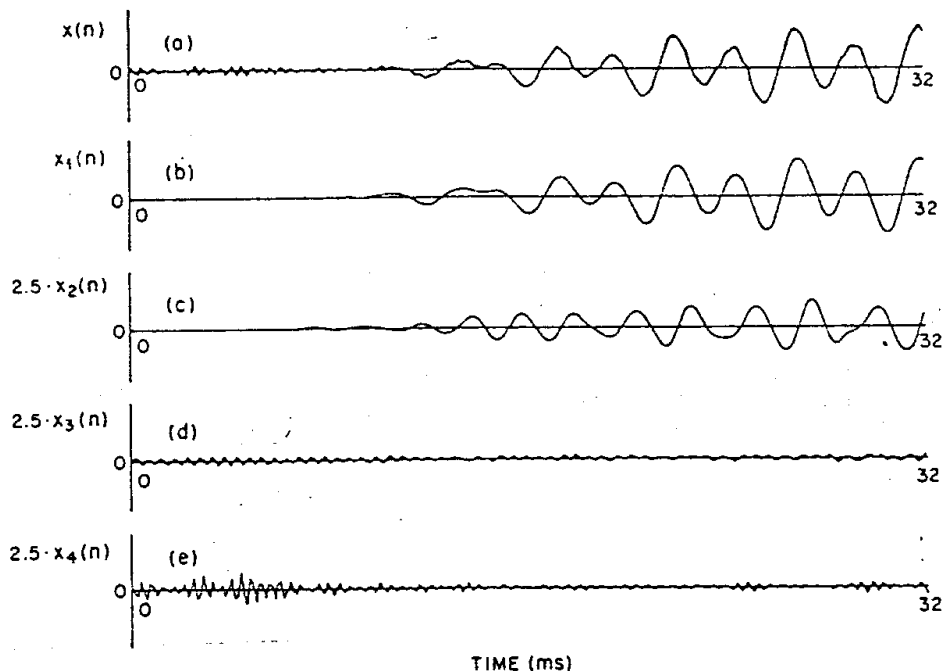
reproducción de voz telefónica; mientras un rango más bajo de bits provee diferentes grados de *calidad de comunicaciones* en la codificación.

La *Tabla 3.2* muestra un esquema de particionamiento que puede ser usado para codificación SBC en rangos de 16, 24 o 32 kb/s. Esta división es obtenida por repetidas aplicaciones del particionamiento QMF. Note que con 16 y 24 kb/s se usan sistemas con cuatro sub-bandas mientras que para 32 kb/s se usan cinco sub-bandas. El ancho de banda de la señal de voz permitido en este sistema con cinco sub-bandas incluye rangos de 3 a 4 kHz; recordemos que para voz telefónica se asume un limite de 3.2 kHz.

Banda	Factor de decimación ξ_k desde 8 kHz	Anchos de Bandas (Hz)	Rangos de Muestreo (Hz)	R_k bits/muestra para 16 kb/s	R_k bits/muestra para 24 kb/s	R_k bits/muestra para 32 kb/s
1	8	0-500	1000	4	5	5
2	8	500-1000	1000	4	5	5
3	4	1000-2000	2000	2	4	4
4	4	2000-3000	2000	2	3	4
5	4	3000-4000	2000	0	0	3

Tabla 3.2: Diseños de codificadores de sub-bandas para 16, 24, 32 kb/s.

La *Figura 7* muestra señales sub-bandas $x_k(n)$; $k=1, 2, 3, 4$ en un sistema de cuatro bandas las cuales usan las cuatro bandas más bajas de la *Tabla 3.2*. Como se ve en esta tabla, los rangos de muestreos respectivos son $f_{sk}=1000, 1000, 2000$ y 2000 Hz. Estos rangos de muestreos son obtenidos de un rango de muestreo original de $f_s=8000$ Hz usando decimadores $\xi_k:1$. Los valores respectivos de ξ_k , también listados en la tabla son 8, 8, 4 y 4.



*Figura 3.7: (a) Onda de speech de banda-completa de duración 32ms; y (b)(c)(d)(e) señales sub-bandas $x_k(n)$ para $k=1, 2, 3, 4$ del sistema SBC de la *Tabla 3.2*.*

Capítulo 4

TRANSFORMADA WAVELET

TRANSFORMADA WAVELET

4.1 Introducción

La Transformada de Fourier es una herramienta útil en el análisis de señales, provee gran versatilidad en aplicaciones que van desde reconocimientos de patrones hasta procesamiento de imágenes. Sin embargo, esta sufre de ciertas limitaciones, como veremos más adelante. La Transformada Wavelet surge tan poderosa y versátil como la transformada de Fourier, y sin algunas de sus limitaciones. A diferencia del análisis de Fourier, la transformada Wavelet no utiliza funciones senos y cosenos, sino que usa unas funciones llamadas “wavelets” (onditas).

4.2 Revisión de la Transformada de Fourier

La transformada de Fourier traslada una función en el dominio del tiempo a una función en el dominio frecuencial donde puede ser analizada por su contenido frecuencial. Esta transformación ocurre porque la transformada de Fourier expande la función original en términos de funciones básicas ortonormales senos y cosenos de duración infinita. Los coeficientes de Fourier de la función transformada representan la contribución de cada seno y coseno a cada frecuencia.

El problema de la Transformada de Fourier de una señal es que no lleva información con respecto a la traslación de la señal en el tiempo. Este problema se puede resolver utilizando Transformadas de Fourier de corto término. Al utilizar transformadas de Fourier de corto término entra en juego la elección de la longitud de ventana que se usará; si el tamaño de la ventana elegido es chico se tendrá buena resolución en tiempo y poca en frecuencias y si el tamaño es grande viceversa.

La transformada de Fourier trabaja asumiendo que la función original en el dominio del tiempo es periódica por naturaleza. Como consecuencia de esto, cuando se trabaja con Transformadas de Fourier de corto término se tiene dificultades cuando existen transiciones abruptas entre un extremo y el otro de la ventana. Este problema se puede resolver “ventaneando” el dato de entrada de modo que los valores muestreados converjan a cero en los extremos.

4.3 Transformada Wavelet Rápida

En los últimos años, han sido descubiertas nuevas familias de funciones básicas ortonormales, que transforman superando los problemas de la transformada de Fourier. Estas funciones básicas son llamadas “Wavelet”, y a diferencia de las ondas seno y coseno de la transformada de Fourier, no necesitan tener duración infinita.

La Transformada Wavelet permite trasladar una función localizada en el dominio del tiempo a una representación localizada en frecuencia y tiempo. Esto conduce a nuevos desarrollos en análisis de señales, procesamiento de imágenes, y compresión de datos.

Hay dos ecuaciones fundamentales sobre los cuales se basan los cálculos wavelets: la función de *scaling* (también llamada “ecuación de dilatación básica” o “recursión fundamental”) y la función *básica* o “wavelet primaria”, las siguientes dos ecuaciones describen ambas funciones para cuando el factor de dilatación es 2:

$$\phi(t) = \sum_{k \in \mathbb{Z}} a_k \phi(2t - k) \quad [4.1.a]$$

$$\psi(t) = \sum_{k \in \mathbb{Z}} (-1)^k a_{k+1} \phi(2t + k) \quad [4.1.b]$$

donde Z es el conjunto de los enteros y los a_k son los coeficientes de wavelets. Ambas funciones son ecuaciones de diferencias con un factor de dilatación 2. Estas funciones son los prototipos de una clase de funciones básicas ortonormales de la forma

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) \quad j, k \in \mathbb{Z} \quad [4.2.a]$$

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad j, k \in \mathbb{Z} \quad [4.2.b]$$

donde el parámetro j controla la dilatación o compresión de la función en la escala del tiempo y amplitud. El parámetro k controla el traslado de la función en el tiempo. El conjunto de funciones básicas formado por $\Psi(t)$ y $\phi(t)$ es un sistema de wavelets escalado y trasladado [Cody,1992].

Ahora bien, no cualquier conjunto de a_k puede ser usado como coeficientes de wavelets. Las condiciones que debe cumplir este conjunto son:

$$\sum_{k \in \mathbb{Z}} a_{2k} = 1 \quad [4.3.a]$$

$$\sum_{k \in \mathbb{Z}} a_{2k+1} = 1 \quad [4.3.b]$$

$$\sum_{k \in \mathbb{Z}} a_k a_k = 2 \quad [4.3.c]$$

$$\sum_{k \in \mathbb{Z}} a_k a_{k+2l} = 0 \quad \text{para } l \neq 0 \quad [4.3.d]$$

Los sistemas wavelets pueden ser de valores reales o complejos. Los sistemas wavelets pueden o no tener soporte compacto (las wavelets tienen soporte compacto si, y solo si, tienen un número finito de coeficientes distintos de cero). Este soporte compacto le permite a las wavelets localizar en tiempo y en frecuencia.

Varias técnicas han sido usadas para crear sistemas wavelets. Estas incluyen splines cúbicos, exponenciales complejas y construcciones de espacios de parámetros. Las siguientes ecuaciones permiten generar espacios de parámetros para sistemas wavelets genéricos de valores reales con 2 a 6 coeficientes, que cumplen con las condiciones consideradas en [4.3]:

$$\begin{aligned}
 a_{-2} &= \left[(1 + \cos(\alpha) + \operatorname{sen}(\alpha))(1 - \cos(\beta) - \operatorname{sen}(\beta)) + 2 \operatorname{sen}(\beta) \cos(\alpha) \right] / 4 \\
 a_{-1} &= \left[(1 - \cos(\alpha) + \operatorname{sen}(\alpha))(1 + \cos(\beta) - \operatorname{sen}(\beta)) + 2 \operatorname{sen}(\beta) \cos(\alpha) \right] / 4 \\
 a_0 &= \left[(1 + \cos(\alpha - \beta) + \operatorname{sen}(\alpha - \beta)) \right] / 2 \\
 a_1 &= \left[(1 + \cos(\alpha - \beta) - \operatorname{sen}(\alpha - \beta)) \right] / 2 \\
 a_2 &= 1 - a_{-2} - a_0 \\
 a_3 &= 1 - a_{-1} - a_1
 \end{aligned}$$

Tabla 4.1 Conjunto de funciones para generar filtros para un Sistema Wavelet con dos a seis coeficientes

donde $-\pi \leq \alpha, \beta \leq \pi$. Como se puede ver se pueden obtener infinitos filtros a partir de infinitos α y β .

Una vez que un sistema wavelet es creado este puede ser usado para expandir una función $g(t)$ en términos de las funciones bases mostradas en el [4.4.a], con los coeficientes calculados por productos internos como vemos en el [4.4.b]. Si el sistema wavelet tiene soporte compacto y un limite superior es ubicado sobre el parámetro j , entonces la ecuación de expansión es como se muestra en las siguientes ecuaciones:

$$g(t) = \sum_{i=-\infty}^{\infty} c(i)\phi_i(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \quad [4.4.a]$$

$$\begin{aligned}
 c(i) &= \langle \phi_i | g \rangle = \int g(t)\phi_i(t)dt \\
 d(j,k) &= \langle \psi_{j,k} | g \rangle = \int g(t)\psi_{j,k}(t)dt
 \end{aligned} \quad [4.4.b]$$

$$g(t) = \sum_{i \in \mathbb{Z}} c(i)\phi_i(t) + \sum_{j=0}^{J-1} \sum_{k \in \mathbb{Z}} d(j,k)\psi_{j,k}(t) \quad [4.4.c]$$

Los coeficientes de expansión $c(l)$ representan la aproximación de la señal original $g(t)$ con una resolución de un punto por cada 2^j puntos de la señal original. Los coeficientes de expansión $d(j,k)$ representan detalles de la señal original a diferentes niveles de resolución. Estos coeficientes describen completa y unívocamente la señal original y pueden utilizarse en un sentido similar a la transformada de Fourier. Las transformadas Wavelets, entonces, es el proceso de determinar los valores de $c(i)$ y de $d(j,k)$ para un $g(t)$ y un sistema Wavelet dado.

La ecuación de expansión lleva naturalmente a un algoritmo recursivo para la Transformada Wavelet haciendo ciertas suposiciones. Primero, la función $g(t)$ se toma como una secuencia de puntos discretos, $y_{j,p}$, muestreados a 2^j puntos por intervalo unitario. Estos puntos pueden verse como los productos internos de $\phi_{j,p}$ y $g(t)$. Esto es, los puntos muestreados son una aproximación de la función continua $g(t)$ (coeficientes $c(l)$). Esto permite que los términos $c(l)$ y $d(j,k)$ sean calculados por convolución directa de las muestras de $g(t)$ con los coeficientes a_k .

El esquema básico de la Transformada Wavelet Rápida contiene convoluciones repetitivas del dato de entrada $y_{j,p}$ con los filtros pasa-bajos y pasa-altos (*Figura 4.1*).

Cada nivel de filtrado es seguido de una decimación por un factor de dos. Los resultados son los coeficientes de la transformada Wavelet $c(l)$ y $d(j,k)$.

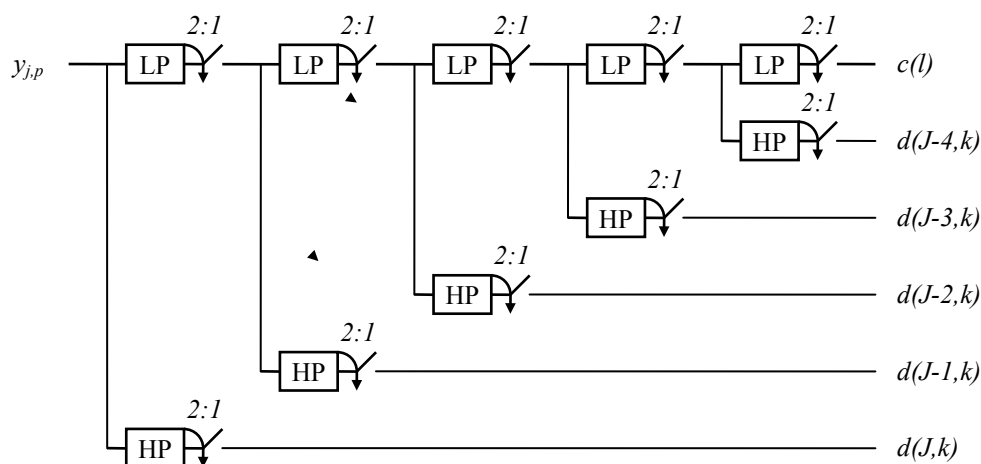


Figura 4.1 Algoritmo piramidal de la Transformada Wavelet Rápida forward

La Transformada Wavelet puede ser implementada con pares de filtros de respuesta de impulso finita (FIR) llamados “Filtros Espejo en Cuadratura” (QMF). Las condiciones que deben cumplir los coeficientes de los filtros son las mostradas en [4.3]. Los coeficientes de los filtros pasa-bajos y pasa-altos se pueden obtener mediante el conjunto de fórmulas vistas en la Tabla 4.1. Los nombres de algunos filtros y sus correspondientes α y β obtenidos mediante estas fórmulas se muestran en la siguiente tabla:

Nombre	Alfa	Beta
Harr	0	0
Daubechies 4-coef.	1,047179551	0
Daubechies 6-coef	1,359803732	-0,782107385
Hat	1,125930376	0,375465743
City	2,713208667	-1,992853802
Wall	1,952966255	-1,443386598
Car Alarm	0	-3,141592654

Tabla 4.2 Ejemplos de Filtros Wavelet y su correspondiente α y β

Es útil el empleo de filtros QMF dado que la respuesta frecuencial de los dos filtros FIR separa los componentes de altas frecuencias y bajas frecuencias de la señal original. El punto de división normalmente es el valor medio entre 0 Hertz y la mitad del rango de muestreo (la frecuencia de Nyquist).

La salida del par de filtros QMF son decimadas (o re-muestreadas) por un factor de dos. La salida del filtro pasa-bajo es puesta en la entrada de otro par de filtros QMF. Esta operación puede ser repetida recursivamente como un algoritmo piramidal. Esto lleva a una división en bandas octavas de la señal.

La salida de cada filtro pasa-bajos es $c(l)$ o componentes de aproximación de la señal original para cada nivel del árbol. La salida de cada filtro pasa-altos es $d(j,k)$, o componentes de detalle, de la señal original a resolución 2^j . El $c(l)$ del nivel previo se utiliza para generar el nuevo $c(l)$ y $d(j,k)$ para el siguiente nivel del árbol. La decimación por dos corresponde a la naturaleza multiresolución (el parámetro j) de las funciones de scaling.

La Antitransformada Wavelet realiza las mismas operaciones en sentido opuesto. Los coeficientes de expansión son combinados para reconstruir la señal original. Los mismos coeficientes a_k son usados como en la Transformada pero en sentido contrario. El proceso trabaja recorriendo las ramas del árbol hacia abajo, combinando las señales de aproximación y de detalle en señales de aproximación con altos niveles de detalle. A diferencia del proceso de decimación usado en la transformación, las señales son interpoladas; es decir, se insertan ceros entre cada muestra de aproximación y de detalle y las señales se pasan entonces, a través de los filtros pasa-bajos y pasa-altos. Los valores intermedios son reemplazados por “valores estimados”, derivados de las convoluciones. La salida de los filtros son sumadas para formar los coeficientes de aproximación del siguiente nivel más alto de resolución, en el proceso de antitransformación. Por último, el conjunto de coeficientes de aproximación obtenidos en la raíz del árbol, es una reconstrucción de las muestras de la señal original. El proceso de antitransformación se muestra en la *Figura 4.2*.

La Transformada Wavelet Rápida es más eficiente, computacionalmente hablando, que las Transformadas de Fourier Rápida y Discreta. La FFT elimina la redundancia que existían en la DFT. Mientras una DFT de longitud N requería orden N^2 , la FFT requiere $N \log_2(N)$. Una Transformada wavelet Rápida para una cadena de la misma longitud requiere orden N .

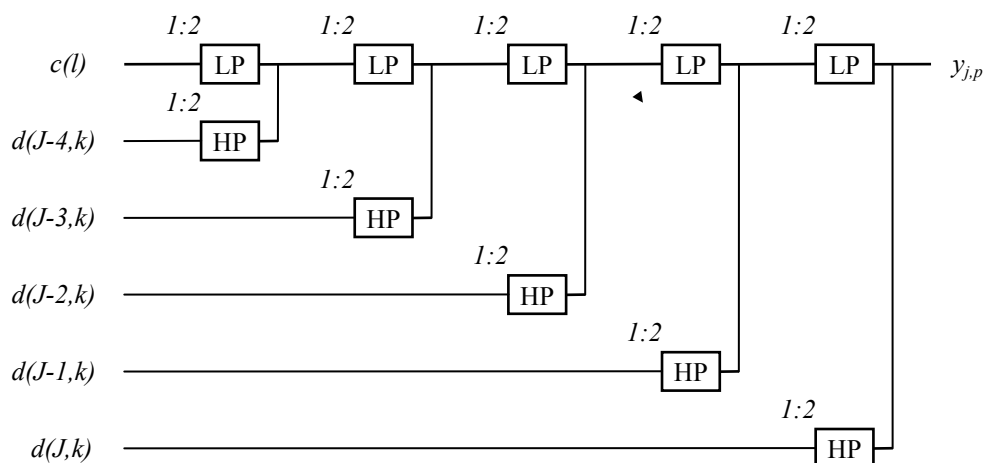


Figura 4.2 Algoritmo Piramidal de la inversa de la Transformada Wavelet Rápida

4.4 Transformada Wavelet Empaquetada

La Transformada Wavelet realiza el análisis de los datos en múltiples niveles de resolución preservando los eventos transitorios.

La Transformada Wavelet Rápida es una implementación eficiente de la Transformada Wavelet Discreta (DWT). La DWT es una transformada aplicada a una secuencia de datos muestreada regularmente. La transformada de los datos exhibe pasos discretos en tiempo sobre un eje y de niveles de resolución sobre otro.

Ya se ha mencionado la superioridad de la DWT sobre la DFT. Esta superioridad radica en la localización de frecuencia-tiempo de la DWT frente a la localización en frecuencia de la DFT. Como resultado de la aplicación de la DWT, las divisiones en frecuencias serán en bandas octavas. Cada nivel de la transformada representa un rango de frecuencias que es la mitad del nivel superior y el doble del nivel inferior; ver *Figura 4.3*.

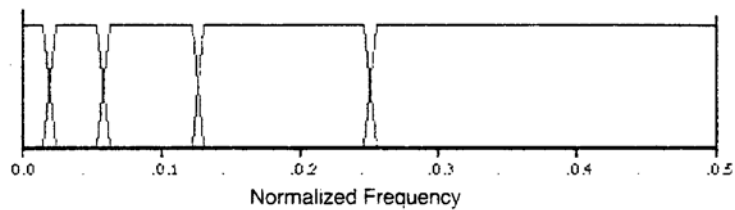


Figura 4.3 División en bandas octavas de la Transformada Wavelet Discreta

Inversamente, la escala en tiempo en cada nivel es el doble que en el superior y la mitad del inferior; ver *Figura 4.4*.

Esta característica de la Transformada Wavelet Discreta tiene problemas cuando intenta localizar en tiempo altas frecuencias. En los niveles altos de la transformada se sacrifica la discriminación en frecuencias en favor de la localización en tiempo.

Actualmente, la DWT es un subconjunto de una transformada aún más versátil, la Transformada Wavelet Empaquetada (WPT) [Cody,1994]. Esta transformada fue desarrollada por el Dr. Ronald A. Coifman de la Universidad de Yale. La WPT “generaliza” el análisis *tiempo-frecuencia* de la Transformada Wavelet. Esto arroja como resultado una familia de transformadas ortonormales de las cuales la DWT es un miembro.

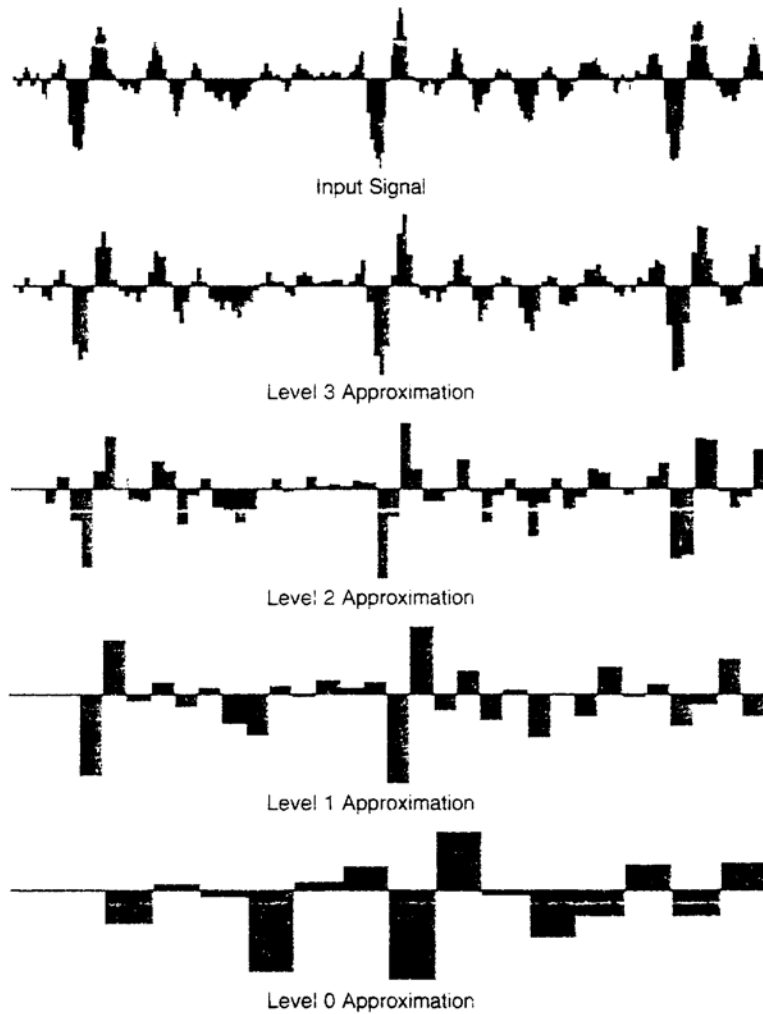


Figura 4.4 Resolución en Tiempo de la Transformada Wavelet Discreta

4.4.1 Desde la DWT a la WPT

Como vimos en el algoritmo de la Transformada Wavelet Rápida, el conjunto de muestras es pasado a través de los filtros Wavelet mediante la operación de convolución. Estos filtros son filtros pasa-bajos y pasa-altos respectivamente y tienen anchos de bandas complementarios (QMF). La salida de estos filtros son decimadas por un factor de 2. El conjunto de datos filtrados pasa-altos son los coeficientes de detalles de la transformada en cada uno de sus niveles y los filtrados pasa-bajos se denominan coeficientes de aproximación. Debido al proceso de decimación estos conjuntos de coeficientes tienen la mitad de los elementos del conjunto inicial.

Los coeficientes de aproximación pueden usarse como datos de entrada para otro par de filtros Wavelet, idénticos al primer par, generando otro conjunto de coeficientes de aproximación y de detalle en el siguiente nivel más bajo en la escala. Este proceso continúa hasta alcanzar el límite del intervalo de unidad. Por ejemplo: si se desea que la transformada tenga 6 niveles (de 0 a 5), el intervalo de unidad debe ser de 64 muestras (2^6).

El conjunto de datos puede ser de cualquier longitud siempre y cuando este sea múltiplo del intervalo de unidad.

El algoritmo de la Transformada Wavelet Rápida puede verse como un árbol binario parcial como se ve en la *Figura 4.5*.

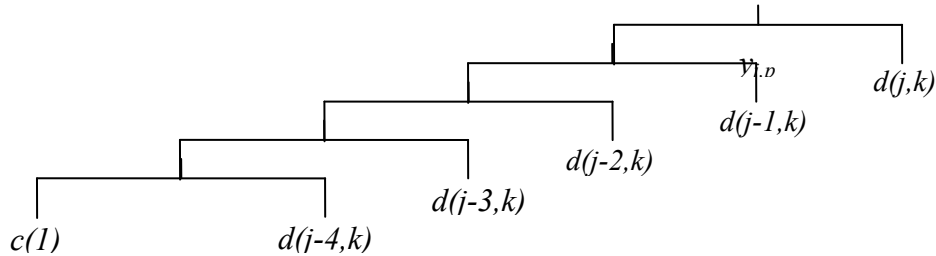


Figura 4.5 Arbol de descomposición de la Transformada Wavelet Rápida

En este gráfico puede verse como el flujo del algoritmo se mueve hacia abajo y a la izquierda, formando nuevos niveles hasta llegar a los coeficientes de aproximación. Las ramas de detalle no se usarán para más cálculos.

Observemos que la Transformada Wavelet se puede detener en cualquier nivel. La transformada “parcial” resultante es una transformada ortonormal válida. Por ejemplo, si el intervalo de unidad para un conjunto de datos fue de 32 puntos (2^5), la transformada correspondiente tendrá 5 niveles. Si la operación de transformación se detiene en el nivel 2, la transformada tendrá solamente 3 niveles, pero los coeficientes de aproximación y detalle de esta transformada se corresponden exactamente con una Transformada Wavelet cuyo intervalo de unidad sea de 8 muestras.

Por lo tanto, mientras los filtros sean filtros Wavelet y el conjunto de datos originales cumpla con los requerimientos del intervalo de unidad, las aplicaciones repetidas del esquema de filtrado resultará siempre en una transformada ortonormal.

Ahora, el conjunto de coeficientes de detalle y aproximación de cada nivel de la transformada forma un par de sub-espacios de los coeficientes del siguiente nivel. Estos subespacios se corresponden aproximadamente con las sub-bandas que muestra la *Figura 4.3*. Estos subespacios forman una cubierta disjunta del espacio de frecuencias del conjunto de datos originales. En otras palabras, los subespacios no tienen elementos en común, y la unión de las sub-bandas de frecuencias cubren el rango de frecuencias del conjunto de datos original.

Lo que Coifman demostró es que cualquier conjunto de sub-espacios, el cual es una cubierta disjunta del conjunto de datos original, es una base ortonormal. Entonces, la Transformada Wavelet básica es un integrante de esta familia de bases de transformada con intervalos de sub-bandas diferentes. Coifman denominó a esta familia como “Librería de Wavelet Empaquetadas”. Todos los integrantes de esta familia se forman por aplicaciones arbitrarias del núcleo de transformación sobre los coeficientes de detalle y aproximación de los niveles superiores.

Esta aplicación del núcleo de transformación en los coeficientes de detalle y aproximación resulta en una expansión de la estructura del algoritmo de la Transformada Wavelet Rápida. El algoritmo piramidal de la Transformada Wavelet Empaquetada se puede representar como un árbol binario completo, ver *Figura 4.6*

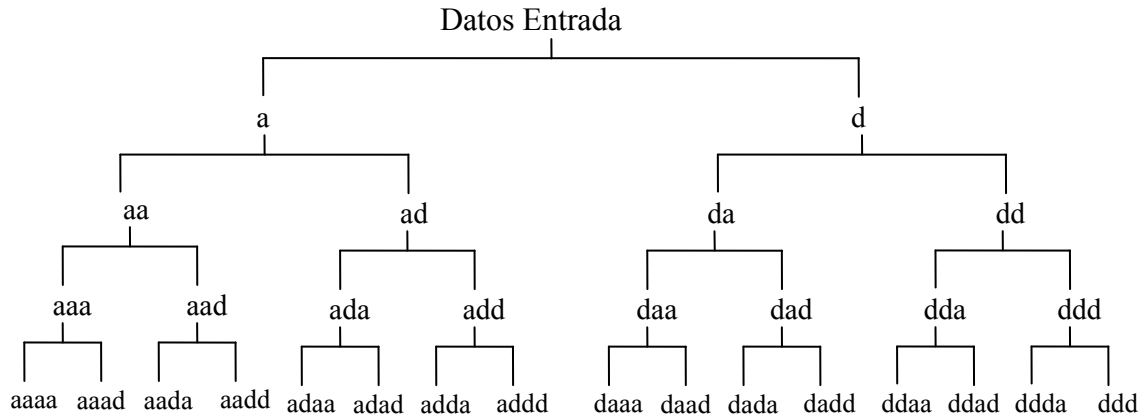
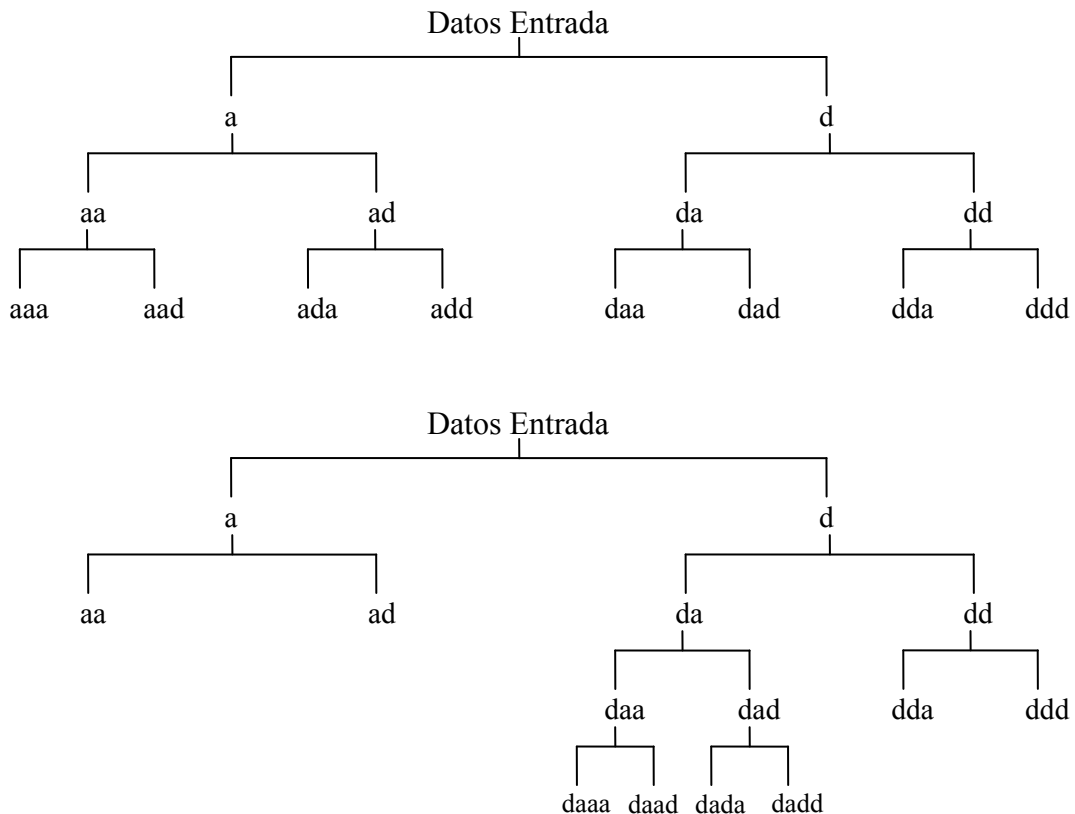


Figura 4.6 Arbol de descomposición de la Transformada Wavelet Empaquetada (árbol binario completo)

Donde los símbolos *a* y *d* en cada nodo indican el orden en que se realizaron las operaciones de filtrado correspondientes al núcleo de transformación. Cada nodo en el árbol es representativo de una Wavelet empaquetada en particular.

Para finalizar, las distintas bases ortonormales que pueden formarse (ver Figura 4.7) y el número infinito de funciones wavelets que pueden crearse, hacen de la WPT una herramienta versátil y muy flexible para el procesamiento de señales.



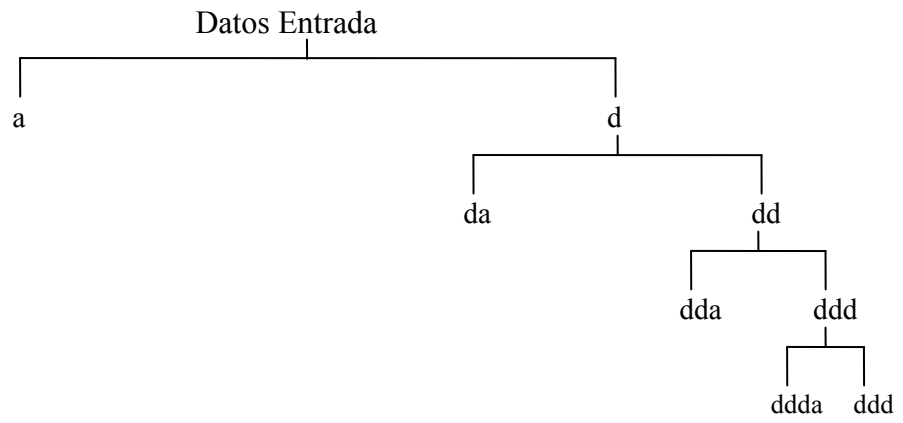


Figura 4.7 Diferentes bases de paquetes wavelets que se pueden obtener en una descomposición WPT

Capítulo 5

**ALGORITMOS DE COMPRESION DE
AUDIO**

ALGORITMOS DE COMPRESION

5.1 Introducción

Ahora que estamos familiarizados con el funcionamiento de la estructura de descomposición Wavelet, estamos en condiciones de concentrarnos en el propósito de este trabajo: la compresión de archivos de audio utilizando esta transformada.

5.2 Algoritmos de Compresión

Para lograr nuestro objetivo hemos desarrollado una serie de algoritmos que emplean la DWT o la WPT, en la descomposición de la señal de entrada. La diversidad de los distintos algoritmos de compresión fueron surgiendo producto de la búsqueda de un mayor índice de compresión, flexibilidad del algoritmo y calidad en la reconstrucción. Todos los algoritmos que se mencionan en este Capítulo han sido implementados en lenguaje C++ versión 3.1.

Todos los algoritmos trabajan con señales que fueron digitalizadas en formato WAV, muestreadas a una frecuencia de 11025 , 22050 y 44100 Hz; y cuantificadas con 16 bits por muestra. Este formato de archivo posee un header de 44 bytes en el cual indica bits por muestras, número de canales, rango de muestreo, etc.; el cual no es necesario en el procesamiento de la señal.

Por limitaciones de hardware, las señales de entrada se fragmentan en frames de igual longitud. Cada frame se procesa individualmente tanto en la etapa de análisis como de síntesis. La única restricción que tenemos es que el número de muestras de entrada debe ser un múltiplo entero de 2^j (intervalo de unidad), donde j es el número de niveles de la transformada.

A continuación enunciaremos las estructuras de los algoritmos de compresión y reconstrucción que hemos desarrollado, diferenciando entre aquellos que usan la DWT en la etapa de transformación de aquellos que usan la WPT.

5.2.1 Algoritmos de DWT

La estructura de los algoritmos de compresión desarrollados que utilizan la Transformada Wavelet Discreta, se muestra en la *Figura 5.1*.

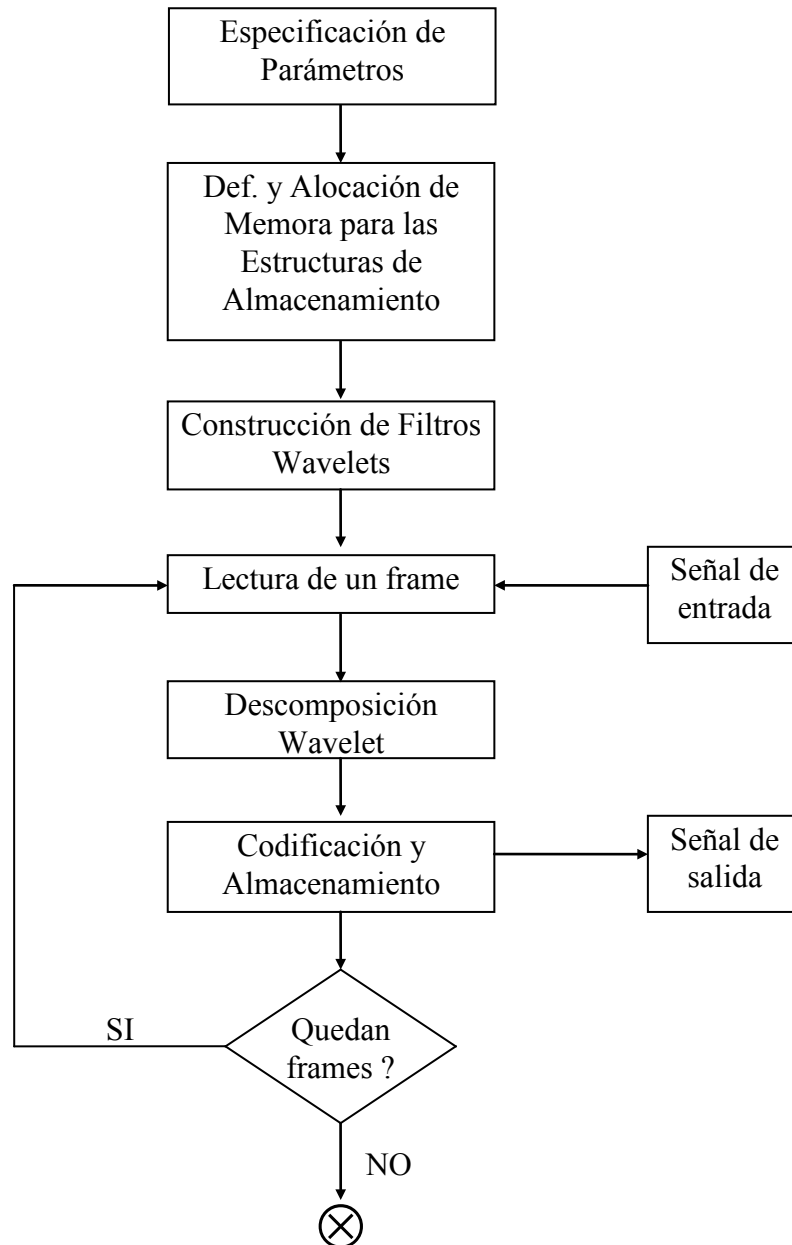


Figura 5.1: Diagrama en bloques de la estructura de los algoritmos de compresión que emplean la Transformada Wavelet Discreta

Donde:

- **Señal de entrada:** Es el archivo en formato RAW al cual se le aplicará la DWT. Un archivo en formato RAW es un archivo en formato WAV sin el header.
- **Especificación de Parámetros:** en esta etapa se inicializan las variables que regirán el proceso de descomposición y compresión de la señal de entrada. Estos parámetros son:
 - Niveles de la transformada
 - Longitud de Frame: número de muestras que conforman el frame a ser procesado.

- Tipo de Filtro: los valores de alfa y beta que se usarán para generar los filtros Wavelet en una etapa posterior.
 - Asignación de bits: cantidad de bits que se utilizaran para comprimir cada nivel de la transformada.
 - Nombre del archivo que contendrá la señal de entrada.
 - Nombre del archivo que contendrá la compresión de la señal de entrada.
- **Definición y asignación de memoria para cada una de las estructuras de almacenamiento:**
 - Dos arreglos de 6 elementos para contener los coeficientes de los filtros pasa-bajos y pasa-altos. Denominaremos a estos arreglos PASA-BAJOS y PASA-ALTOS, respectivamente.
 - Un arreglo de longitud $N+5$, donde se almacenará cada frame de la señal de entrada para su posterior descomposición. El valor N indica la longitud de frame. Las cinco últimas posiciones son seteadas a cero para permitir el overrun de la convolución. Denominaremos a este arreglo FRAME.
 - Un arreglo de arreglos creado dinámicamente, donde se almacenarán los coeficientes de la transformada. Cada par de arreglos guarda los coeficientes de aproximación y de detalle para cada nivel. La longitud de cada par de arreglos es determinada acorde a la longitud del frame y el número de niveles de la transformada. Si la longitud del arreglo de datos de entrada es $N+5$, entonces el primer par de arreglos tiene una longitud de $N/2+5$ elementos, el segundo de $N/4+5$ elementos y así siguiendo. La *Figura 5.2* ilustra esta estructura a la cual denominaremos COEFICIENTES.

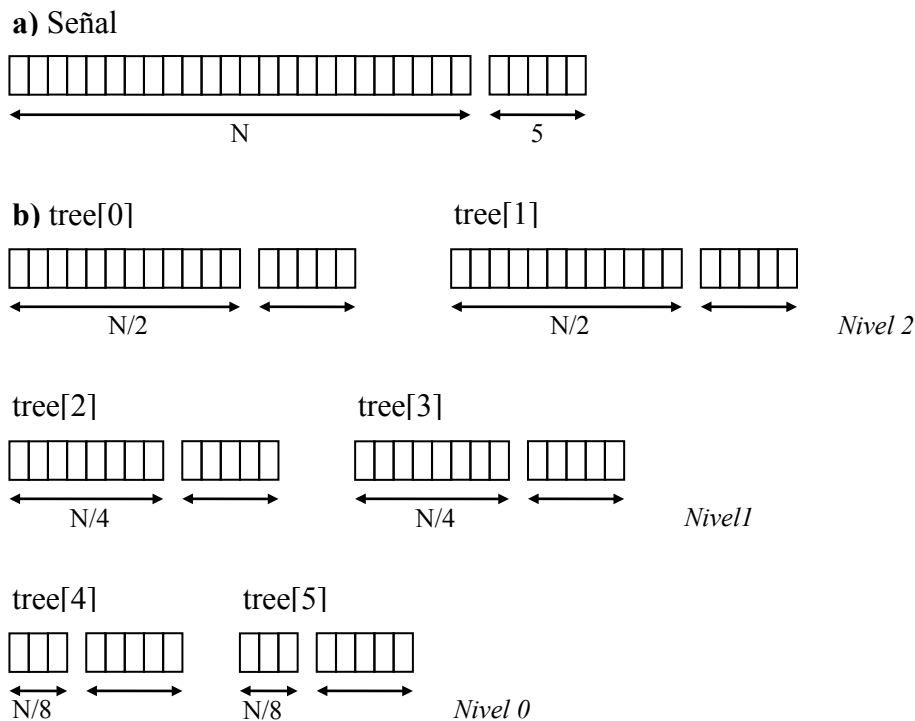


Figura 5.2 Estructuras de almacenamiento para la Transformada Wavelet Discreta

- **Construcción de filtros Wavelets:** Se construyen los filtros Wavelet a través de los ángulos alfa y beta especificados en la primer etapa. Estos filtros se construyen en base a las ecuaciones descritas en la *Tabla 4.1*.
- **Lectura de un frame desde la señal de entrada:** Se leen N muestras desde el archivo de entrada especificado y se almacenan en FRAME.
- **Descomposición Wavelet:** Se descomponen las muestras almacenadas en FRAME utilizando la estructura explicada en la sección 4.3 y guardando los resultados en COEFICIENTES. La descomposición se basa en el número de niveles y tipos de filtros definidos en la primer etapa.
- **Codificación de la descomposición y almacenamiento:** Se codifican cada uno de los niveles de la descomposición de acuerdo a una metodología seleccionada para comprimir y a las asignaciones de bits determinadas para cada nivel de la transformada. Es en esta etapa donde se originan las principales diferencias entre un algoritmo y otro. A continuación describiremos las metodologías empleadas en cada uno de ellos.

Algoritmo 1: Se fijan ya sea por parámetro o por programa, los bits asignados a cada nivel de la transformada. Este número es fijo para todos los frames que se procesen. Cada número de bits, correspondiente a un nivel determinado, será la cantidad de bits que se tomarán de cada coeficiente de ese nivel. Por Ejemplo, si para una descomposición en 6 niveles asignamos, 4 bits al nivel 5, 8 bits a los niveles 4 y 3, 12 bits a los niveles 2 y 1 y 16 bits al nivel 0, significa que obtendremos los 4 bits mas significativos de los coeficientes del nivel 5, los 8 bits mas significativos de los coeficientes de los niveles 4 y 3, etc. Los bits “extraídos” se van concatenando en bytes, que cuando formen una palabra, se almacenarán en el archivo de salida.

Esta forma arbitraria de codificar los coeficientes puede verse mejor a través del siguiente ejemplo: Supongamos que queremos codificar una banda cuyos coeficientes varían desde -32.768 a 32.767, tomando los cuatro bits más significativos de cada coeficiente. Por lo tanto, las representaciones posibles son 16 (2^4). El rango cubierto por cada codificación (4096 valores) puede verse en la *Tabla 5.1*.

Desde	Hasta	Codificación
32.767	28.672	0111
28.671	24.576	0110
24.575	20.480	0101
20.479	16.384	0100
16.383	12.288	0011
12.287	8.192	0010
8.191	4.096	0001
4.095	0	0000
-1	-4.096	1111
-4.097	-8.192	1110
-8.193	-12.288	1101
-12.289	-16.384	1100
-16.385	-20.480	1011
-20.481	-24.576	1010
-24.577	-28.675	1001
-28.673	-32.768	1000

Tabla 5.1 Ejemplo de codificación con 4 bits de una banda cuyos coeficientes varían desde -32.768 a 32.767

Por lo tanto, la técnica de codificación de este algoritmo consistente en tomar una cierta cantidad de bits mas significativos de cada coeficiente, no es más que la idea de aplicar umbrales equidistantes al rango dinámico cubierto por los coeficientes de cada banda.

Algoritmo 2: La técnica que emplea este algoritmo surge como una mejora a las falencias de aquella que utiliza el *Algoritmo 1*. Observemos en el algoritmo anterior, que las representaciones cubren un rango de valores entre -32.768 y 32.767 cualquiera sea el rango que cubra la banda a codificar. Por lo tanto, si el rango dinámico de una banda varía, digamos entre -8192 y 8191; estamos desperdiciando representaciones. Estamos codificando la banda usando las representaciones 0001, 0000, 1111 y 1110 y el resto de las representaciones están destinadas a valores que no se van a dar.

Por lo tanto, la idea de codificación del algoritmo 2 es aprovechar al máximo el número de representaciones posibles, de manera que cada una de ellas represente a algún coeficiente y todas se usen en la codificación de cada banda. Para llevar esto a cabo, es necesario distribuir los umbrales dentro del rango dinámico de la banda. Al hacerlo, la distancia entre las representaciones se acortará y se obtendrá una mejor calidad en la codificación. Para ver esto, supongamos una banda cuyo rango de valores varíe entre -8.192 y 8.191. Las representaciones posibles en cuatro bits y los valores que ellas representan se muestra en la *Tabla 5.2*.

Desde	Hasta	Codificación
8.191	7.168	0111
7.167	6.144	0110
6143	5.120	0101
5.119	4.096	0100
4.095	3.072	0011
3.071	2.048	0010
2.047	1024	0001
1.023	0	0000
-1	-1.024	1111
-1.025	-2.048	1110
-2.049	-3.072	1101
-3.073	-4.096	1100
-4.097	-5.120	1011
-5.121	-6.144	1010
-6.145	-7.168	1001
-7.169	-8.192	1000

Tabla 5.2 Ejemplo de codificación con 4 bits de una banda cuyos coeficientes varían desde -8.192 a 8.191

Observemos que la cantidad de valores representado por cada codificación disminuye de 4.096 en el algoritmo 1 a 1.024.

En la práctica, la implementación de esta técnica se logra:

1. Obteniendo el máximo absoluto de la banda a codificar (*MAX*).
2. Obtener el número máximo codificable (*MAX_COD*) con una cierta cantidad de bits (*CANT_BITS*):

$$\text{MAX_COD} = 2^{(\text{CANT_BITS} - 1)} - 1.$$

3. Obtener un coeficiente de amplitud (*COEF_AMPLITUD*) que representa la distancia entre cada una de las representaciones:

$$\text{COEF_AMPLITUD} = \text{MAX} / \text{MAX_COD}$$

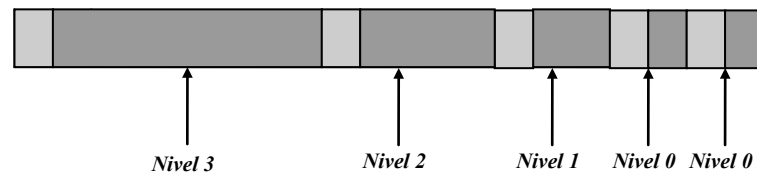
4. Para cada coeficiente (*COEF*) de la banda se codifica con *CANT_BITS* el valor resultante de

$$\text{COEF} / \text{COEF_AMPLITUD}$$

- **Señal de salida**

Algoritmo 1: La señal de salida de este algoritmo será un archivo binario donde para cada frame se guardan las codificaciones de los coeficientes de detalles de cada nivel de la transformada y las codificaciones de los coeficientes de aproximación del nivel 0.

Algoritmo 2: Difiere del *Algoritmo 1* en que el coeficiente de amplitud (COEF_AMPLITUD) encabeza la codificación de cada banda. Para una descomposición Wavelet en tres niveles, la codificación de un frame se ve como:



Donde:

- Coeficiente de Amplitud
- Codificación de una Banda

Figura 5.3 Almacenamiento de un frame para el Algoritmo 2

En la etapa de reconstrucción, la estructura de los algoritmos es similar a la de la etapa de descomposición. Un diagrama generalizado se presenta en la *Figura 5.4*, donde:

- **Señal Comprimida:** Es el archivo que contiene la señal comprimida (extensión .PAK).
- **Especificación de Parámetros:** en esta etapa se inicializan las variables que regirán el proceso de descomposición y compresión de la señal de entrada. Estos parámetros son:
 - Niveles de la transformada
 - Longitud de Frame: número de muestras que conforman el frame a ser procesado.
 - Tipo de Filtro: los valores de alfa y beta que se usarán para generar los filtros Wavelet.
 - Asignación de bits: Es la cantidad de bits que se utilizaron para comprimir cada nivel de la transformada. Este parámetro es necesario para determinar la cantidad de datos comprimidos que se deben leer, de manera que representen la longitud del frame.
 - Nombre del archivo que contendrá la señal comprimida.
 - Nombre del archivo que contendrá la señal reconstruida.
- **Definición y alocaión de memoria para cada una de las estructuras de almacenamiento:** Las estructuras que se manejan en esta etapa son las mismas que las utilizadas en la etapa de descomposición.
- **Construcción de filtros Wavelets:** Se construyen los filtros Wavelet a través de los ángulos alfa y beta especificados en la primer etapa. Estos filtros se construyen en base a las ecuaciones descriptas en la *Tabla 4.1* y deben ser idénticos a los usados en la descomposición.

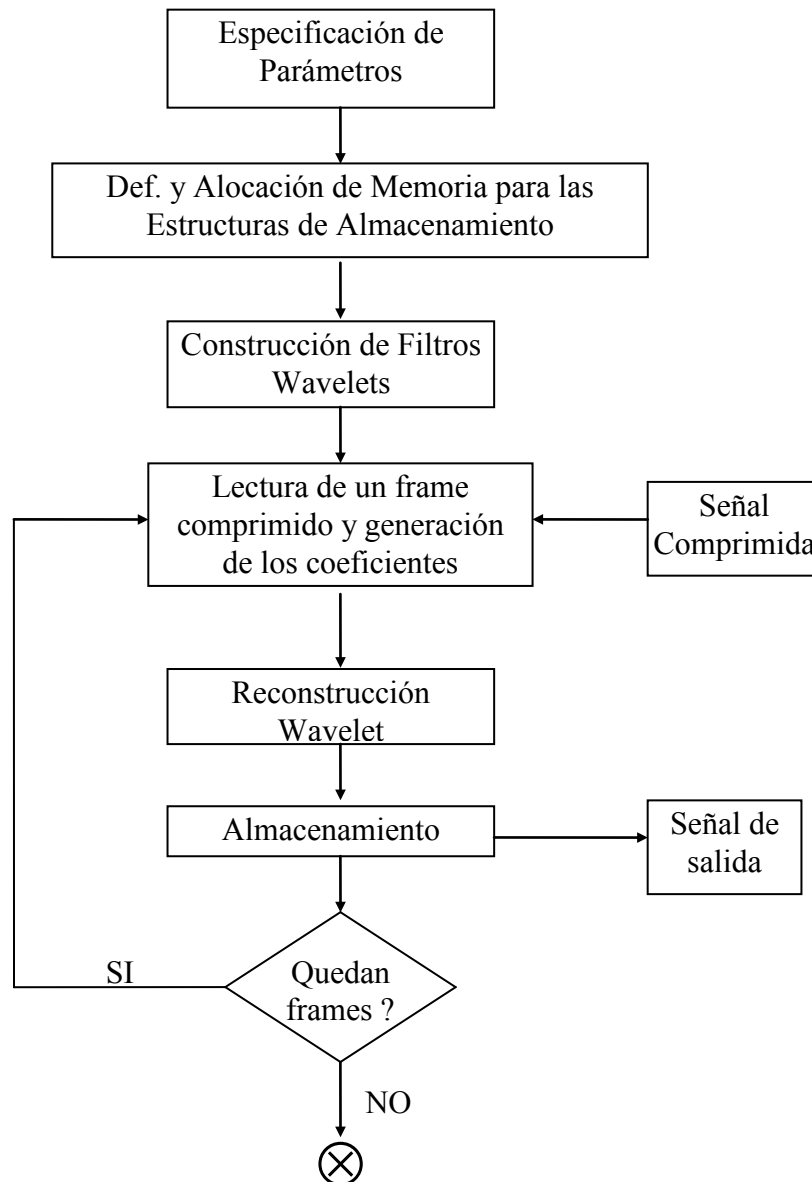


Figura 5.4 Diagrama en bloques de la estructura de los algoritmos de descompresión que usan la Transformada Wavelet Discreta

- **Lectura de un frame comprimido y generación de los coeficientes:** Aquí se leen los coeficientes comprimidos de un frame determinado, se los descomprime y se guardan en COEFICIENTES. La cantidad de bytes que representan a un frame depende de los bits asignados en la compresión de cada banda, del número de niveles de la transformada y del algoritmo de codificación propiamente dicho. Según las técnicas de codificación empleadas en los algoritmos 1 y 2, tendremos las siguientes posibilidades en la descompresión de los coeficientes

Algoritmo 1: A medida que vamos leyendo los bytes de la señal comprimida y sabiendo de antemano los bits asignados en cada banda, extraemos los bits que correspondan a cada coeficiente. Como las compresiones se guardan consecutivamente es fácil encontrar casos en que los bits de un coeficiente se distribuyan en dos o más bytes consecutivos. Con los bits extraídos (n) se forma un

coeficiente en 16 bits donde los n bits representan la parte más significativa del mismo. Los bits que restan hasta completar la palabra se setean de la siguiente manera:

1. Si el bit más significativo de n es 0, sabemos que el coeficiente es positivo. Por lo tanto, la palabra se completa con ceros.
2. Si el bit más significativo de n es 1, sabemos que el coeficiente es negativo. Por lo tanto, la palabra se completa con unos.

Si consideramos una codificación en cuatro bits entonces las representaciones posibles y los coeficientes resultantes de su reconstrucción se muestra en la *Tabla 5.3*.

Codificación	Coeficiente Reconstruido	Coeficiente Decimal
0111	0111000000000000	28.672
0110	0110000000000000	24.576
0101	0101000000000000	20.480
0100	0100000000000000	16.384
0011	0011000000000000	12.288
0010	0010000000000000	8.192
0001	0001000000000000	4.096
0000	0000000000000000	0
1111	1111111111111111	-1
1110	1110111111111111	-4.097
1101	1101111111111111	-8.193
1100	1100111111111111	-12.289
1011	1011111111111111	-16.385
1010	1010111111111111	-20.481
1001	1001111111111111	-24.577
1000	1000111111111111	-28.673

Tabla 5.3 Decodificación de coeficientes en 4 bits para un rango -32.768 a 32.767

Algoritmo 2: En este algoritmo la técnica de reconstrucción de los coeficientes sigue los siguientes pasos:

1. Se lee el coeficiente de amplitud de la primer banda (*COEF_AMPLITUD*).
2. Se leen los bits correspondientes a cada coeficiente de la banda y se guarda en una palabra. Los bits de un coeficiente ocupan la parte menos significativa.
3. Se multiplica cada coeficiente por *COEF_AMPLITUD*. De este modo, obtenemos la reconstrucción del coeficiente.
4. Idem pasos anteriores para cada una de las bandas codificadas.

Supongamos que una banda fue codificada con cuatro bits y que el coeficiente de amplitud utilizado en la compresión fue $COEF_AMPLITUD=1024$, entonces los valores resultantes en la reconstrucción se muestran en la *Tabla 5.4*

Codificación	Coficiente Reconstruido
7	7168
6	6144
5	5120
4	4096
3	3072
2	2048
1	1024
0	0
-1	-1024
-2	-2048
-3	-3072
-4	-4096
-5	-5120
-6	-6144
-7	-7168
-8	-8192

Tabla 5.4 Decodificación de coeficientes en 4 bits para el algoritmo 2

Este proceso de leer palabras y formar coeficientes, en ambos algoritmos, se repite hasta que la estructura COEFICIENTES se complete. Una vez que reconstruimos todos los coeficientes del frame, aplicamos la inversa de la DWT.

- **Reconstrucción Wavelet:** Reconstruimos el frame correspondiente a partir de los valores almacenados en COEFICIENTES, aplicando la inversa de la DWT (Ver Sección 4.3). Los muestras del frame reconstruido se guardan en FRAME.
- **Almacenamiento:** Se almacenan consecutivamente los datos contenidos en FRAME en el archivo que representará a la señal de salida
- **Señal de salida:** Es un archivo en formato RAW que contiene la señal reconstruida.

5.2.2 Algoritmos de WPT

La siguiente etapa en nuestro trabajo fue desarrollar un algoritmo que utilice la Transformada Wavelet Empaquetada en la compresión de audio. Como vimos en la sección 4.4, la WPT permite múltiples caminos de reconstrucción, de los cuales la DWT es uno de ellos. La diversidad de estos caminos (bases) y la flexibilidad de su estructura (niveles, filtros, etc.), hacen de la WPT una herramienta muy potente y atractiva para la compresión de audio. Para la elección de una base ortonormal válida se debe tener en cuenta que si los vectores bases de la familia son puestos en arreglos de N columnas por $n=\log_2(N)$ filas (Ver Figura 5.5), entonces cualquier subconjunto admisible será una base ortonormal. Un subconjunto admisible debe cumplir con:

- Cada columna contiene exactamente un elemento,
- Los elementos en una fila simple aparecen en bloques contiguos de 2^k elementos, donde $0 \leq k \leq n$ es un entero.
- Los bloques en las filas comienzan en un múltiplo entero de su longitud desde el extremo izquierdo del arreglo.

A1				D1			
A1.1		A1.2		D1.1		D1.2	

Figura 5.5 Vectores bases dispuestos en arreglos de N columnas por $n=\log_2(N)$ filas

De todas las bases válidas, la elección de la mejor de ellas para la reconstrucción de un frame, es un punto determinante en el desarrollo de los algoritmos de compresión que utilicen esta transformada.

En la práctica, hemos desarrollado un algoritmo que emplea el análisis multiresolucional de la Transformada Wavelet Empaquetada en la compresión de señales de audio. La estructura de este algoritmo en la etapa de análisis (Ver *Figura 5.6*) es muy similar a la empleada por los dos anteriores; se especifican algunos parámetros que regirán todo el proceso, se descompone la señal en frames, se codifican y almacenan los coeficientes del camino de reconstrucción que hayamos elegido.

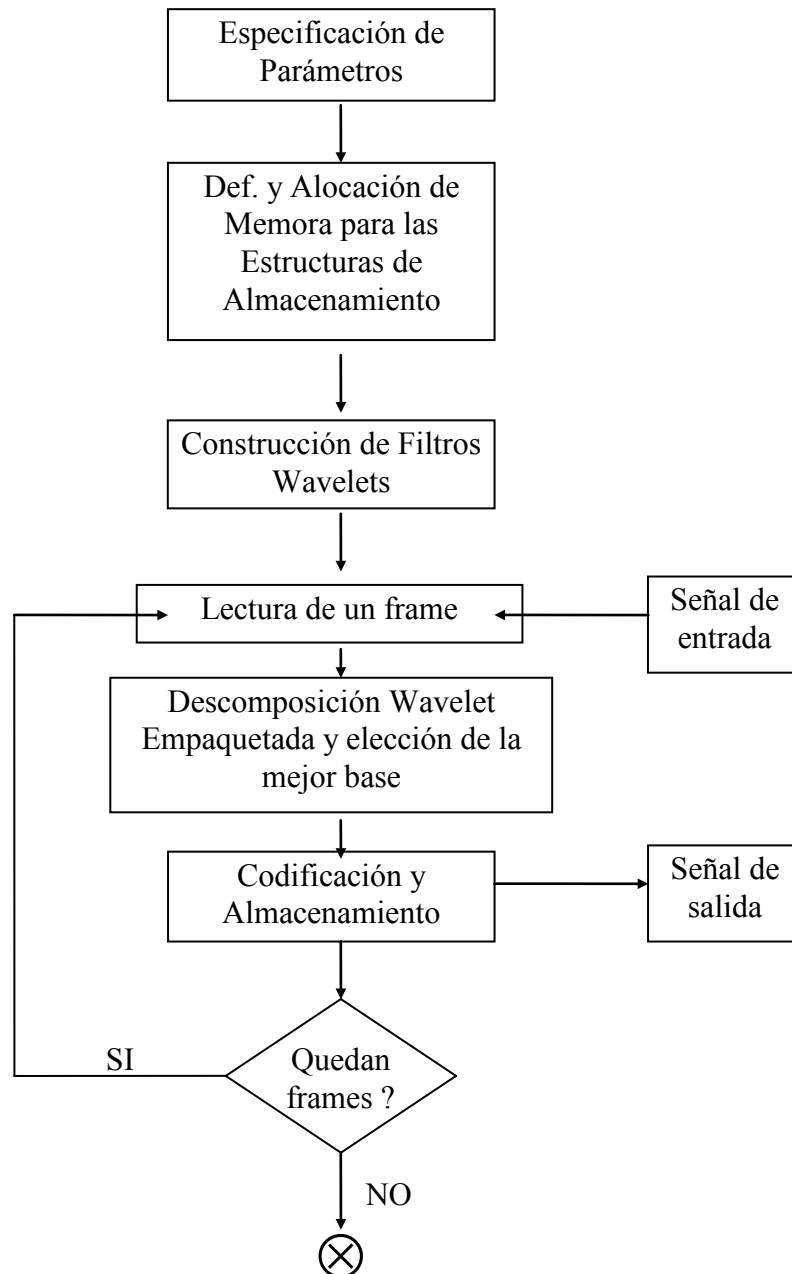


Figura 5.6 Diagrama en bloques de la estructura del algoritmo que usa la Transformada Wavelet Empaquetada

Donde:

- **Señal de entrada:** Es el archivo en formato RAW al cual se le aplicará la WPT.
- **Especificación de Parámetros:** Los parámetros que se inicializan son similares a los utilizados en los algoritmos que implementan la DWT. Estos parámetros son:
 - Niveles de la transformada
 - Longitud de Frame: número de muestras que conforman el frame a ser procesado.
 - Tipo de Filtro: los valores de alfa y beta que se usarán para generar los filtros Wavelet.

- Nivel de Compresión deseado.
 - Nombre del archivo que contendrá la señal de entrada.
 - Nombre del archivo que contendrá la compresión de la señal de entrada.
- **Definición y asignación de memoria para cada una de las estructuras de almacenamiento:** A las estructuras de los filtros pasa bajos (PASA-BAJOS), pasa altos (PASA-ALTOS) y la que almacena las muestras de un frame (FRAME), explicadas anteriormente, se agregan:
 - Un tensor creado dinámicamente, donde se almacenarán los coeficientes de la transformada. La *Figura 5.7* ilustra esta estructura a la cual denominaremos COEFICIENTES.

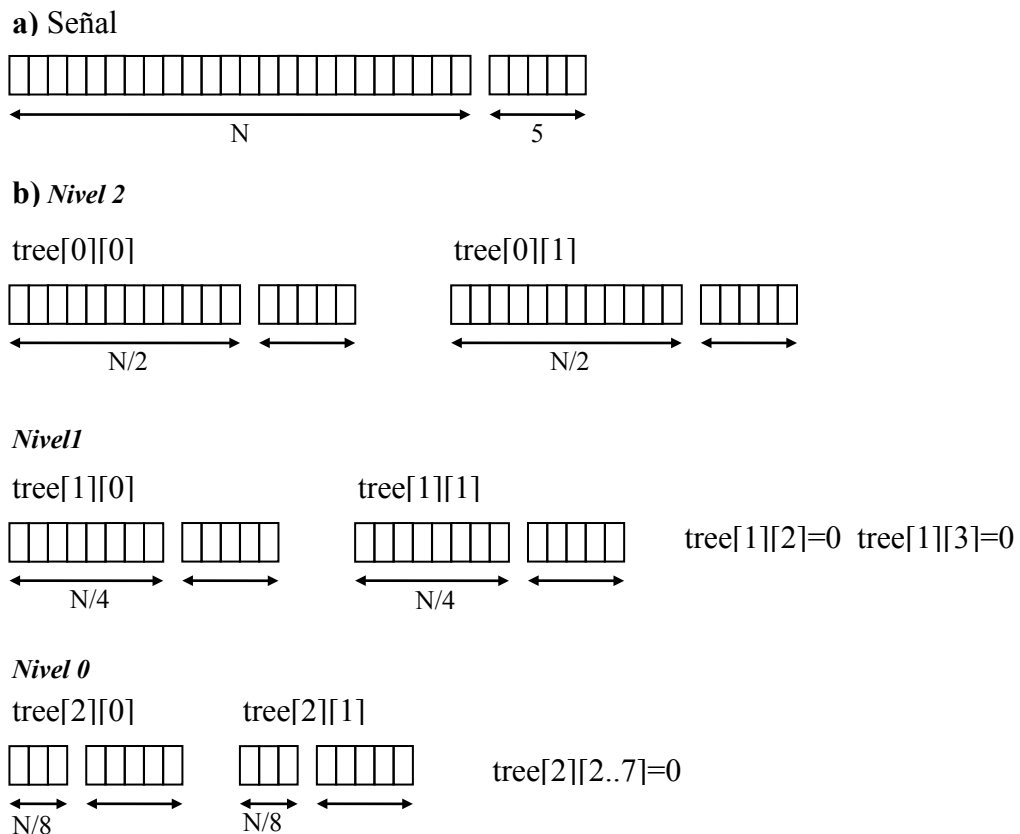


Figura 5.7 Estructuras de Almacenamiento para la Transformada Wavelet Empaquetada

- Una matriz creada dinámicamente, donde se almacenarán los bits asignados a cada banda. El propósito de esta estructura se explicará en la etapa de Codificación. Denominaremos a esta estructura BITS_ASIG.
 - Una matriz creada dinámicamente, donde se almacenarán los coeficientes de amplitud de las bandas codificadas. Denominaremos a esta estructura COEF_AMPLITUD.
- **Construcción de filtros Wavelets:** Se construyen los filtros Wavelet a través de los ángulos alfa y beta especificados en la primer etapa. Estos filtros se construyen en base a las ecuaciones descritas en la *Tabla 4.1*.

- **Lectura de un frame desde la señal de entrada:** Se leen N muestras desde el archivo de entrada especificado y se almacenan en FRAME.
- **Descomposición Wavelet Empaquetada y elección de la mejor base:** Se descomponen las muestras almacenadas en FRAME utilizando la estructura explicada en 4.4, se almacenan los coeficientes resultantes en COEFICIENTES y los bits asignados a los coeficientes de cada banda en BITS_ASIG. Para la descomposición del frame se utilizó el criterio de mínima entropía. Esto es, el núcleo de transformación se aplica mientras la cantidad de bits necesaria para codificar los hijos, sea menor que la necesaria para codificar el padre o se alcance el número máximo de niveles de la transformada. La cantidad de bits necesarias para codificar una banda se obtiene de acuerdo a la cantidad óptima para codificar la amplitud máxima presente en la misma ($\log_2(\text{máxima amplitud})$) y la cantidad de coeficientes de la banda. Por ejemplo, la cantidad de bits necesarios para codificar una banda cuyo máxima amplitud es 8.192 y la cantidad de coeficientes es 64 es $\log_2(8192)*64=832$ bits.

El proceso de transformación para cada frame consiste en:

1. Aplicar el núcleo de transformación al frame de entrada. Esto nos da como resultados los elementos A1 y D1 de la *Figura 5.5*.
 2. Calcular las amplitudes máximas MAX_{A1} y MAX_{D1} de las bandas A1 y D1 respectivamente
 3. Asignar una cantidad de bits de codificación, tanto a los coeficientes de aproximación A1 como a los de detalle D1. La cantidad de bits se asigna de acuerdo al $LOG_2(MAX_{A1})$ para los coeficientes de A1 y al $LOG_2(MAX_{D1})$ para los coeficientes de D1. La cantidades obtenidas se almacena en BITS_ASIG.
 4. Aplicar el núcleo de transformación a A1 para obtener A1.1 y A1.2.
 5. Aplicar el paso 2 y 3 a los elementos obtenidos en el paso 4.
 6. Si $Bits_Asignados(A1.1)*Long(A1.1) + Bits_Asignados(A1.2)*Long(A1.2)$ es mayor que $Bits_Asignados(A1)*Long(A1)$ entonces, volver al paso 4 y aplicarlo a A1.1 y A1.2.
En caso contrario no seguir descomponiendo.
- El proceso descrito por los pasos 4, 5 y 6 se ejecutará siempre y cuando no se sobrepase el nivel máximo de descomposición previsto.
7. Idem 4, 5, y 6 para D1.

Observemos que el camino de descomposición que se obtiene y los bits asignados a la codificación de cada banda, dependen exclusivamente de las características del frame de entrada.

- **Codificación de la descomposición y almacenamiento:** Una vez obtenido el camino de descomposición del frame y los bits asignados a cada banda, se verifica que la cantidad total de bits que insumiría la codificación del frame cumpla con el nivel de compresión deseado. Para esto, se debe determinar la cantidad máxima de bits que debería tener un frame comprimido, esta es:

$$CANT_MAX = 16 * \text{longitud del frame} / \text{porcentaje de compresión}$$

Si la cantidad de bits en la compresión del frame es mayor que CANT_MAX, se debe “podar” las cantidades de bits asignadas a las bandas. El proceso de “podar” comienza a partir de las bandas de más alta frecuencia hasta las de más baja, procesando de a una banda a la vez. Se decrementa en un bits la cantidad asignada a la primer banda, se recalcula la cantidad total de bits y si no cumple con el nivel de compresión deseado, se repite el proceso para las siguientes bandas. Si se procesan todas las bandas y no se alcanza el objetivo, se reanuda el proceso con la primer banda.

Cuando la descomposición del frame en cuestión cumpla con el índice de compresión especificado, se procede a comprimir los coeficientes. La codificación se lleva a cabo según las cantidades de bits almacenadas en CANT_BITS y siguiendo la técnica de codificación del *Algoritmo 2*. En este caso se codifican además, los coeficientes de amplitud de cada banda de la siguiente manera:

1. Se eleva el coeficiente de amplitud a la -1, de esta manera se obtiene un valor entre 0 y 1.
2. Se busca en la *Tabla 5.5* el valor superior inmediato al cociente de la división anterior.
3. Se toma el índice de la tabla correspondiente al valor encontrado y se lo codifica con 5 bits.

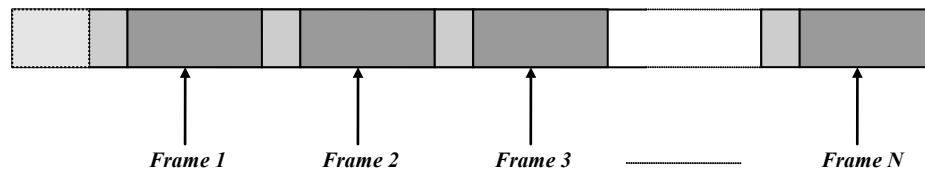
0)	1.000000000000000	23)	0.00492156660115
1)	0.79370052598410	24)	0.00390625000000
2)	0.62996052494744	25)	0.00310039267963
3)	0.500000000000000	26)	0.00246078330058
4)	0.39685026299205	27)	0.00195312500000
5)	0.31498026247372	28)	0.00155019633981
6)	0.250000000000000	29)	0.00123039165029
7)	0.19842513123686	30)	0.00097655625000
8)	0.15749013123686	31)	0.00077509816991
9)	0.125000000000000	32)	0.00061519582514
10)	0.09921256574801	33)	0.00048828125000
11)	0.07874506561843	34)	0.00038754908495
12)	0.062500000000000	35)	0.00030759791257
13)	0.04960628287401	36)	0.00024414062500
14)	0.03937253280921	37)	0.00019377454248
15)	0.031250000000000	38)	0.00015379895629
16)	0.02480314143700	39)	0.00012207031250
17)	0.01968626640461	40)	0.00009688727124
18)	0.015625000000000	41)	0.00007689947814
19)	0.01240157071850	42)	0.00006103515625
20)	0.00984313320230	43)	0.00004844363562
21)	0.007812500000000	44)	0.00003844973907
22)	0.00620078535925	45)	0.00003051757813

Tabla 5.5 Valores de coeficientes de amplitud y su correspondiente codificación según el índice de la Tabla

Por lo tanto el almacenamiento de la codificación de un frame, implica almacenar por cada banda:

1. Cantidad de bandas que se codifican.
2. La identificación de la banda y la cantidad de bits asignados en la codificación.
3. La codificación del índice de la *Tabla 5.5*.
4. La codificación de coeficientes.

• **Señal de salida:** La señal de salida es un archivo binario, con extensión .PAK. La *Figura 5.8* muestra su estructura



Donde:

- Header del Archivo*
- Header inter-frame*
- Coeficientes comprimidos del frame*

Figura 5.8 Estructura de un archivo comprimido según el algoritmo que usa la Transformada Wavelet Empaquetada

El header del archivo contiene el número de frames comprimidos. El header inter-frame contiene los datos especificados en los pasos 1 y 2 del punto anterior.

En cuanto al proceso de reconstrucción podemos decir que es muy similar al utilizado en la etapa de análisis y se vale principalmente de los datos ubicados en el header del archivo (Número de frames a procesar), en los headers *inter-frame* (Bits asignados, Coeficiente de amplitud, etc.) y de las mismas estructuras de datos utilizadas en la etapa de compresión. Los pasos a seguir en el proceso de reconstrucción son:

1. Leer el header del archivo.
2. Por cada frame comprimido hacer:
 - 2.1. Leer header inter-frame .
 - 2.2. Armar el árbol BITS_ASIG y COEF_AMPLITUD.
 - 2.3. Leer coeficientes comprimidos del frame.
 - 2.4. Armar la estructura COEFICIENTES.
 - 2.5. Aplicar la WPT inversa a COEFICIENTES. El resultado se almacenará en FRAME.
 - 2.6. Almacenar FRAME en un archivo binario de salida.

Capítulo 6

**INTERFASE WINDOWS PARA
COMPRESION DE AUDIO**

INTERFASE WINDOWS PARA COMPRESION DE AUDIO

6.1 Introducción

En el Capítulo 5 hemos descripto los algoritmos de compresión que desarrollamos a lo largo de nuestro trabajo. De todos ellos, decidimos utilizar el que emplea la WPT, como un resultado práctico de nuestro estudio y sobre el cual realizaremos algunas pruebas de calidad que mostraremos en el Capítulo 7. Si bien la primera implementación fue para un entorno DOS, nos pareció más atractivo migrarla a un entorno Windows, para tener una representación más visual de su procesamiento. De todas maneras, y no conformes con una “caja negra” que comprimiera y descomprimiera archivos de audio, decidimos implementar una herramienta Windows que nos permita editar, comprimir, y descomprimir archivos de audio y sobre todo explote una de las principales características de la transformada, como es visualizar los distintos caminos de reconstrucción en cada frame. También se agregó a esta herramienta, la posibilidad de medir la calidad de reconstrucción del algoritmo a través de la medida SNR (Ver Anexo A).

Por lo tanto, el resultado obtenido es una herramienta que en un principio surgió como una alternativa a una implementación DOS y terminó siendo la interfase de nuestro algoritmo, permitiendo evaluarlo a través del SNR y visualizar cada camino de reconstrucción y los bits asignados en cada banda de descomposición.

A lo largo de este Capítulo desarrollaremos cada una de las características de esta interfase la cual fue implementada en C++ para Windows Versión 3.1.

6.2 Barra de Menús

La interfase es en realidad una aplicación MDI que tiene una barra de título, una barra de herramientas y un área de trabajo. La barra de menús es la que muestra la *Figura 6.1*.

El menú *Archivo* agrupa las operaciones que manejan los archivos de audio (Abrir, Guardar, etc.), el menú *Operaciones* agrupa las operaciones que se pueden realizar con un archivo de audio (Comprimir, Descomprimir, etc.) y el menú *Ventana* realiza las operaciones básicas de una aplicación MDI (Cascada, Mosaico, etc.). La opción *Play* permite escuchar el archivo de audio de la ventana activa. y la opción *Acerca de* despliega una leyenda con datos del trabajo.

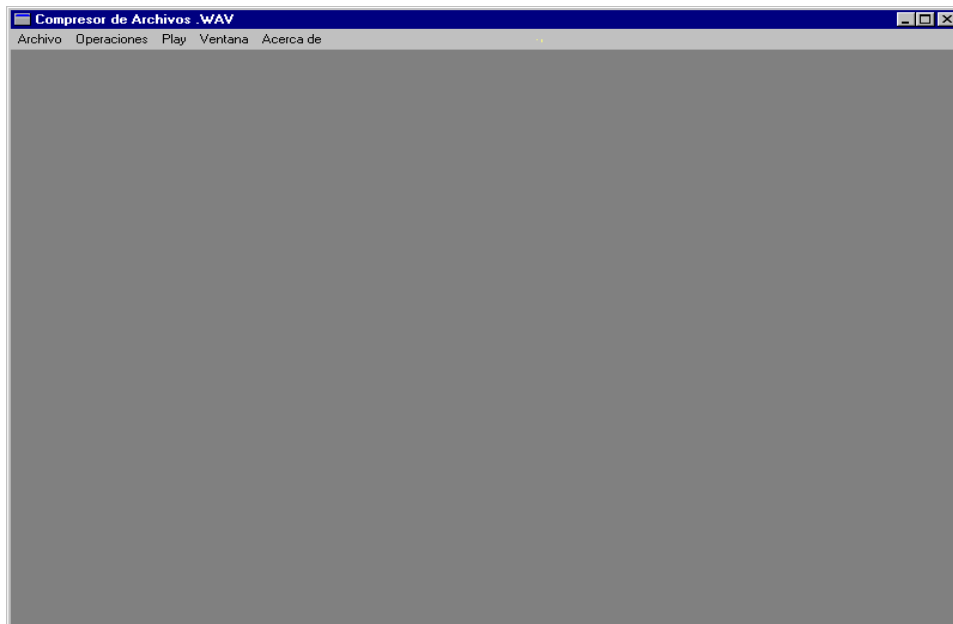


Figura 6.1 Barra de menús de la Interfase

6.3 Menú Archivo

El menú *Archivo* presenta cinco alternativas de selección como muestra la Figura 6.2.



Figura 6.2 Operaciones del menú Archivo

La opción *Abrir* presenta un cuadro de diálogo (Ver Figura 6.3) que permite al usuario abrir un archivo de audio en formato WAV.

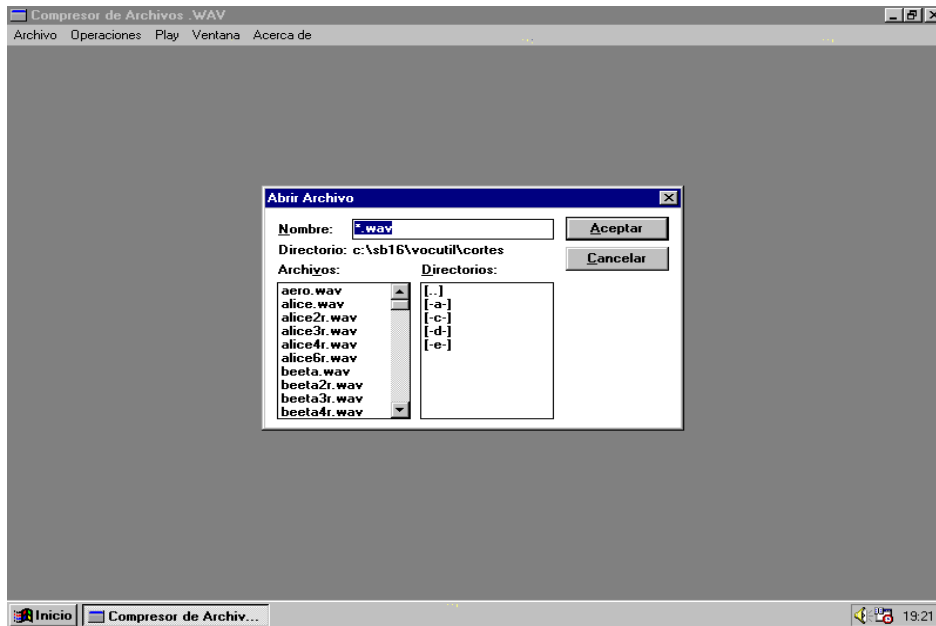


Figura 6.3 Cuadro de Diálogo “Abrir”

Al elegir el archivo y clicar en *Aceptar* se abre una ventana que muestra un gráfico escaleado de las muestras del archivo (Ver *Figura 6.4*). El título de la ventana es el path del archivo elegido en la figura anterior.

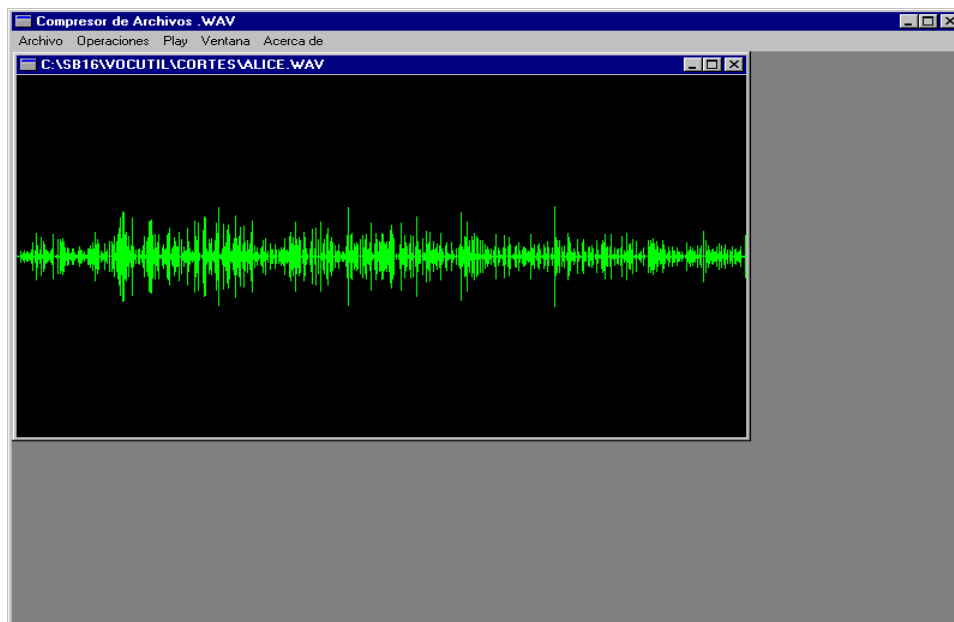


Figura 6.4 Muestra la ventana generada luego de abrir el archivo *Álice.wav*.

Como veremos más adelante, al comprimir un archivo de audio o al recomponer uno ya comprimido, se genera una ventana que representa la señal reconstruida. Esta ventana nos permitirá comparar la señal original y su reconstrucción en forma interactiva, analizar la descomposición, etc. Ahora bien, las opciones de menú *Guardar como .PAK* y *Guardar como .WAV*, están disponibles solo cuando la ventana activa representa la reconstrucción de una señal. La función de cada una de estas operaciones

es almacenar bajo el nombre que el usuario elige, a través de un cuadro de diálogo similar al de la *Figura 6.3*, la compresión de la señal original (*Guardar como .PAK. Extensión .PAK*) o la reconstrucción en formato WAV (*Guardar como .WAV*). En ambas opciones el título de la ventana activa cambia por el nombre elegido por el usuario.

La opción de menú *Cerrar* cierra la ventana activa si la señal que representa esta en formato WAV o ha sido guardada en disco. Si la señal representada no ha sido guardada en disco, la aplicación interroga al usuario si quiere almacenar la representación en formato WAV o con extensión .PAK. En ambos casos se presenta un mensaje como el que muestra la *Figura 6.5*.

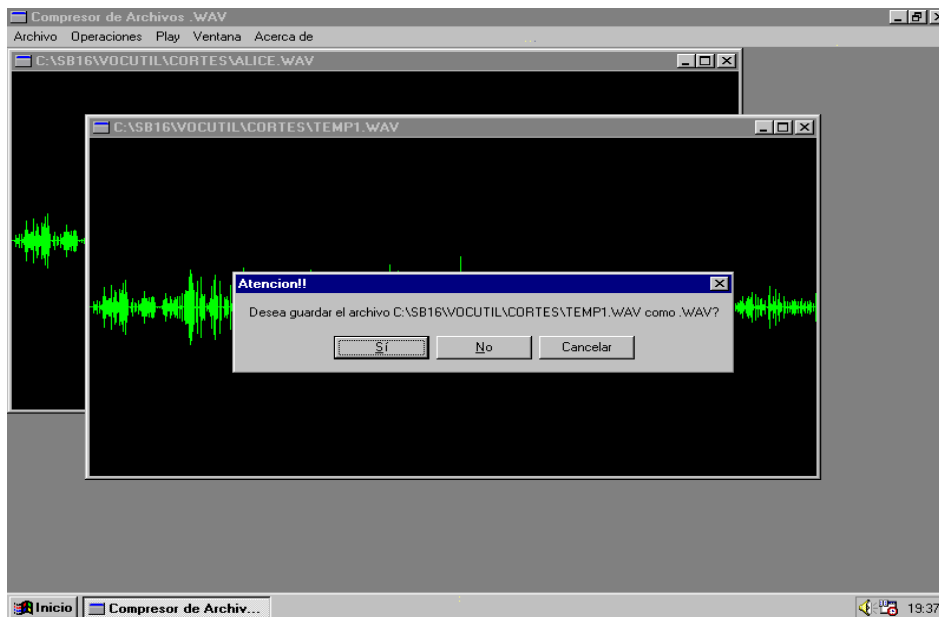


Figura 6.5 Muestra el mensaje que aparece al querer cerrar un archivo que representa una reconstrucción de la señal que aparece en segundo plano.

Si el usuario responde que *SI* entonces, la aplicación ejecuta las acciones explicadas para *Guardar como...* . Si responde que *NO* sólo se cierra la ventana y si elige *Cancelar* la operación se cancela y la ventana activa no se cierra.

Si elegimos la opción *Salir* la aplicación cierra todas las ventanas abiertas en el área de trabajo y termina.

6.4 Menú Operaciones

Este menú presenta, como se ve en la *Figura 6.6*, cuatro operaciones: *Comprimir*, *Descomprimir*, *SNR* y *Arbol de Bits*.

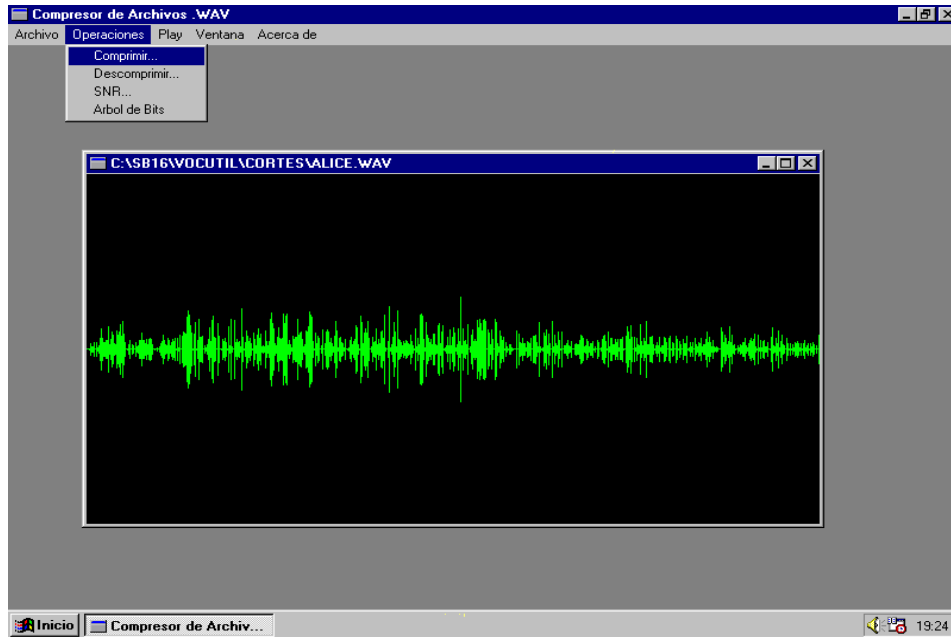


Figura 6.6 Opciones del menú Operaciones

La opción *Comprimir* despliega el cuadro de diálogo de la Figura 6.7, donde da a elegir el rango de compresión que se desea para la señal, en formato WAV, que muestra la ventana activa.

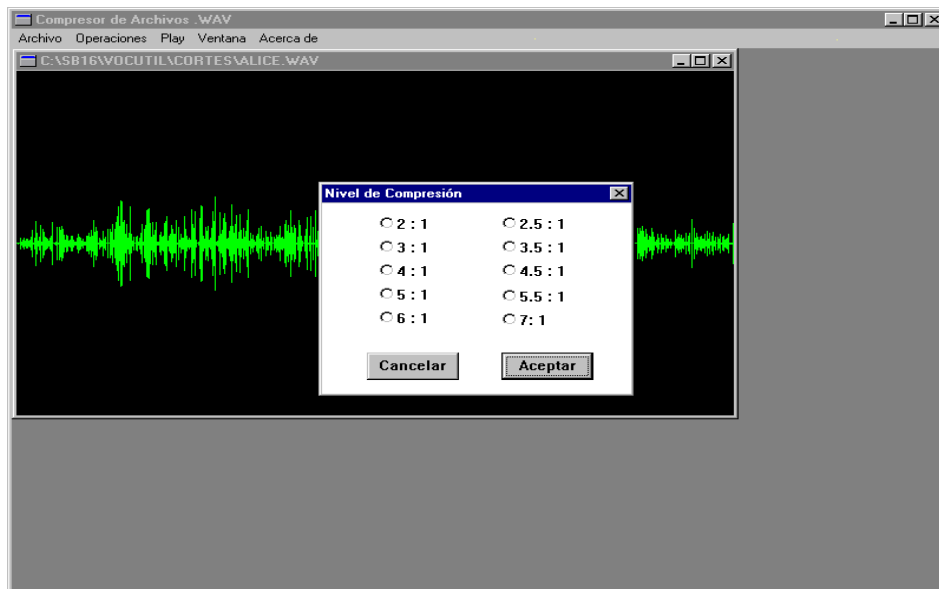


Figura 6.7 Cuadro de Diálogo que permite elegir el índice de compresión para la señal representada en la ventana activa.

Una vez elegido el rango de compresión, la aplicación ejecutará la compresión y reconstrucción de la señal por medio del algoritmo explicado en 5.2.2 y mostrará la señal resultante en una nueva ventana.

Cuando queremos *Descomprimir* un archivo se despliega el cuadro de diálogo que muestra la *Figura 6.8* en el que se solicita al usuario el nombre de un archivo comprimido, con extensión .PAK.

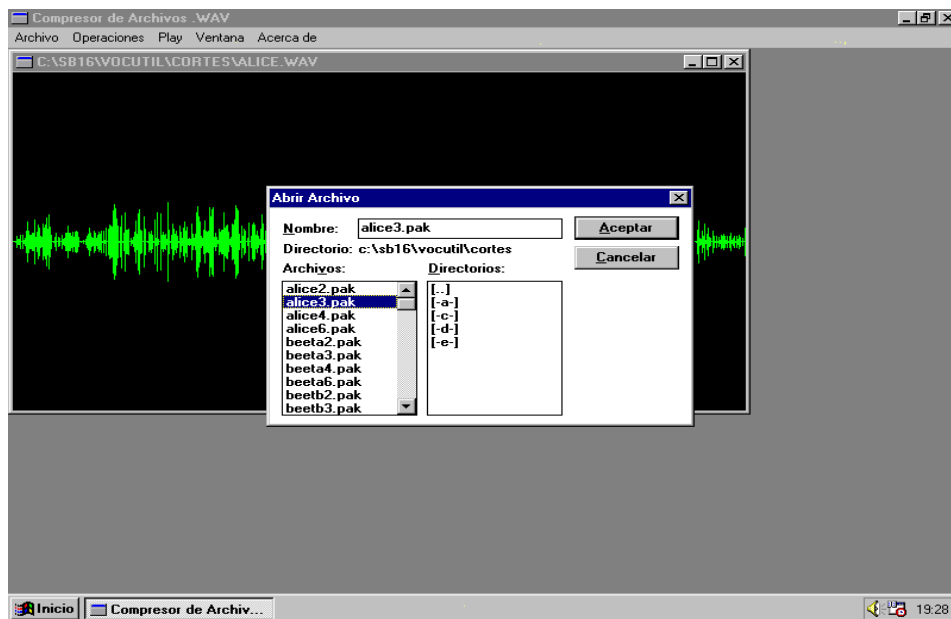


Figura 6.8 Cuadro de Diálogo para ingresar el nombre de un archivo .PAK a descomprimir

Una vez elegido el nombre del archivo se realiza la reconstrucción explicada en 5.2.2 y se despliega la señal resultante en una nueva ventana. La *Figura 6.9* ilustra este hecho.

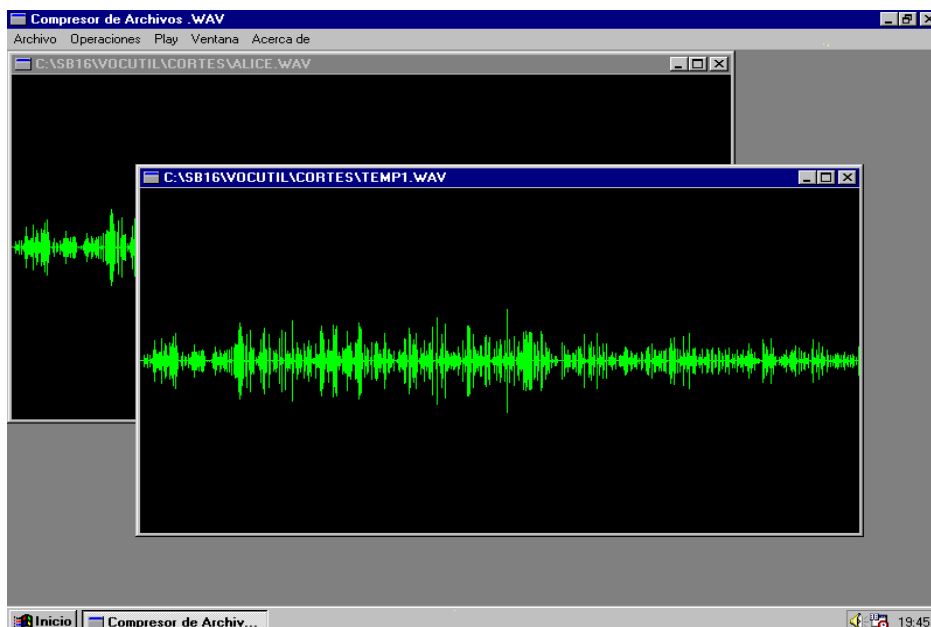


Figura 6.9 Ventana resultante de la reconstrucción de una señal

La opción *SNR* permite medir la calidad de reconstrucción de nuestro algoritmo comparando la señal original con una reconstrucción. Para medir calidad debemos tener

activa una ventana que represente una señal en formato WAV y seleccionar SNR del menú Operaciones. Una vez seleccionada esta opción se abre un cuadro de diálogo similar al de la *Figura 6.3* en el que el usuario debe elegir el otro archivo en formato WAV que intervendrá en la comparación. Una vez realizado el proceso de comparación se muestra el resultado como en la *Figura 6.10*.

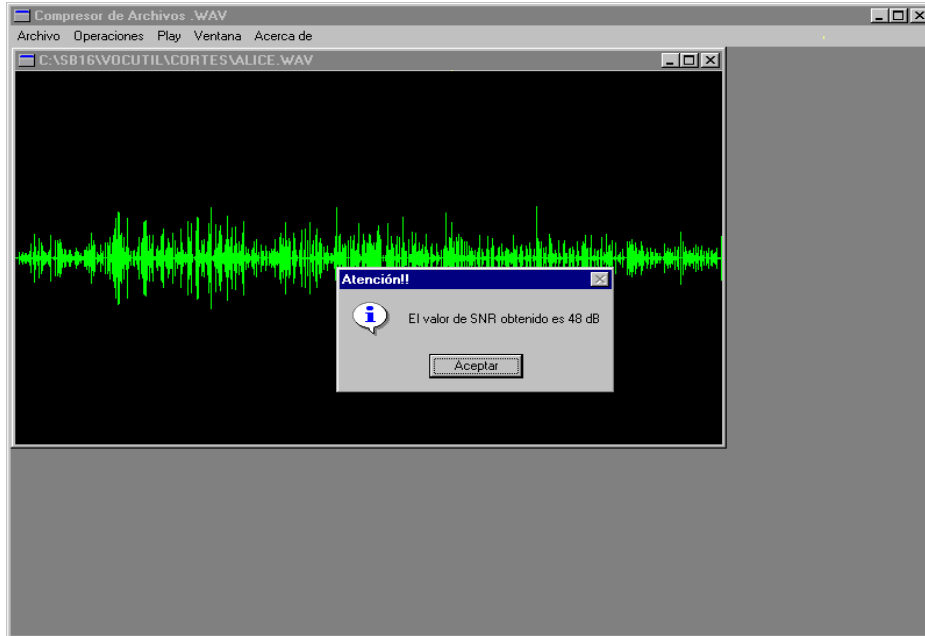


Figura 6.10 Resultado de la comparación SNR entre dos archivos WAV

Cuando la ventana activa muestra la reconstrucción de una señal esta disponible la opción *Arbol de Bits*. Esta opción permite mostrar en una nueva ventana, el camino de reconstrucción (Wavelet Básica) de cada frame. La *Figura 6.11* muestra una ventana de este tipo.

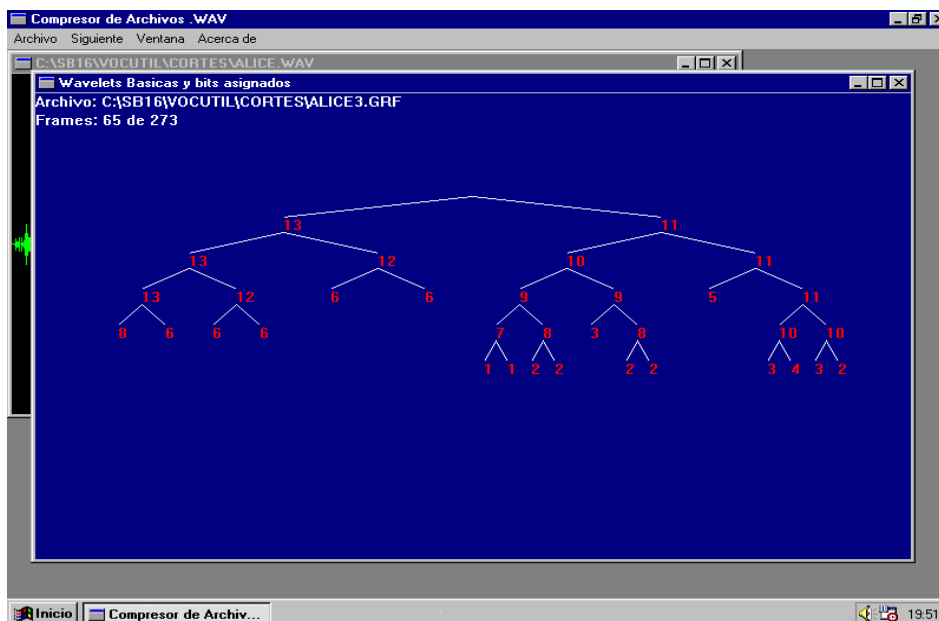


Figura 6.11 Ventana que muestra las Wavelets Básicas de un frame descompuesto

Observemos que cuando la ventana activa es la que muestra un árbol de bits cambia la barra de menús. El menú operaciones se reemplaza por una opción *Siguiente* que permite iterar las muestras de la ventana.

6.5 Menú Ventana

El menú *Ventana* permite organizar y manejar las ventanas abiertas en el área de trabajo de la aplicación. La *Figura 6.12* muestra las cuatro opciones de este menú.

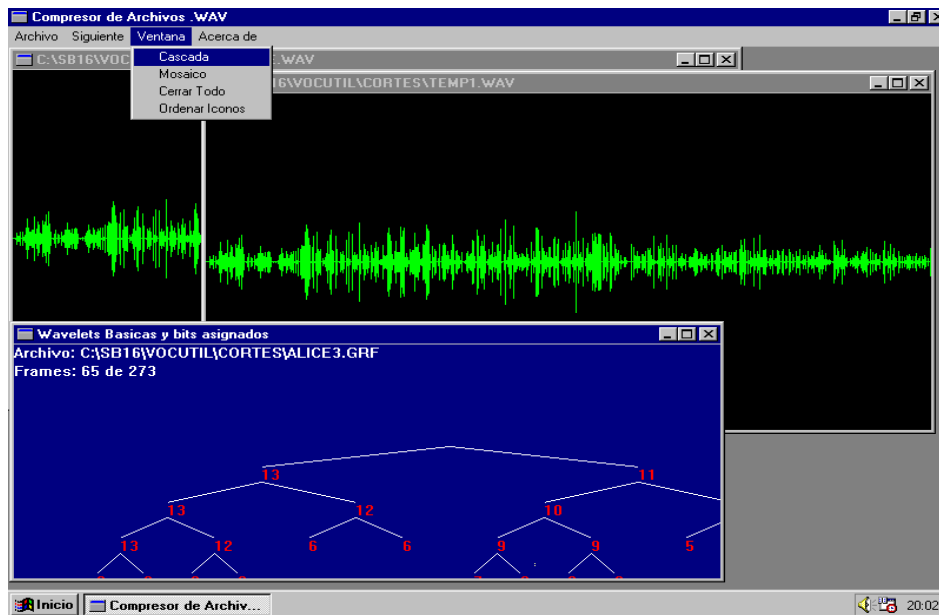


Figura 6.12 Opciones del menú Ventana

Las opciones *Cascada*, *Organizar iconos* y *Mosaico* son las clásicas opciones que podemos encontrar en cualquier aplicación Windows. La opción *Cerrar Todo* cierra todas las ventanas activas. La *Figura 6.13* muestra la disposición en cascada de las ventanas abiertas en la *Figura 6.12*.

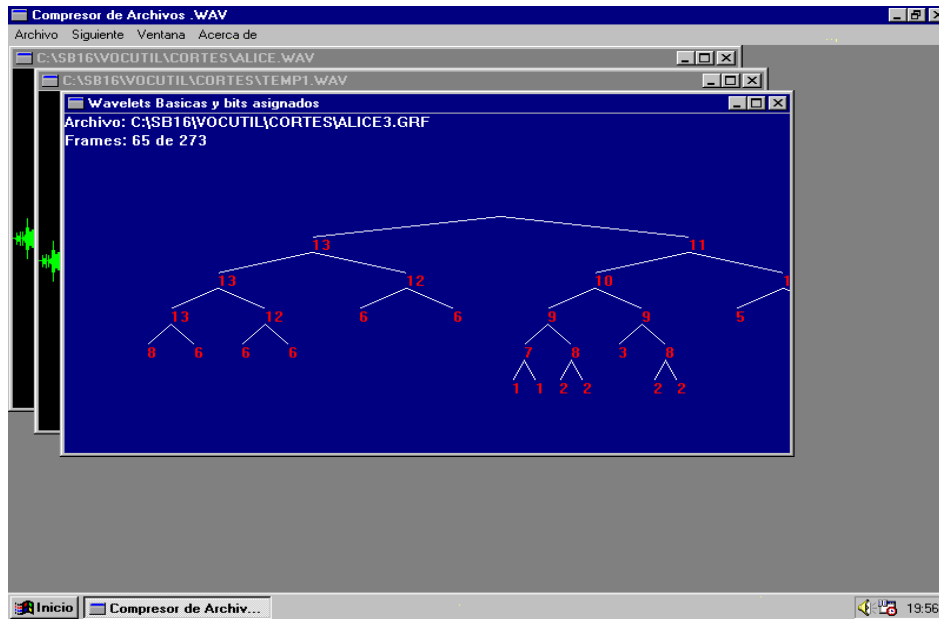


Figura 6.13 Organización en Cascada.

Capítulo 7

**EVALUACION, RESULTADOS Y
CONCLUSIONES**

EVALUACION, RESULTADOS Y CONCLUSIONES

7.1 Introducción

Ha llegado el momento de probar que tanto comprime nuestro algoritmo. Para ello, hemos seleccionado algunos extractos musicales de diversos géneros y muestras de señales de voz.

Indudablemente, estas pruebas, por su calidad y cantidad, no intentan ser una medida exacta de la performance de nuestro algoritmo, sino una mera estimación informal de su funcionamiento.

Luego de enumerar las muestras, sus características y evaluar la compresión con distintos índices, enunciaremos algunas conclusiones sobre la WPT y su uso práctico.

7.2 Evaluación y Resultados

Cuando explicamos los distintos algoritmos que habíamos implementado a lo largo de este trabajo, dijimos que trabajaríamos con señales en formato WAV, muestreadas a 11025 Hz, 22050 Hz o 44100 Hz y muestreadas a 16 bits por muestra. Bajo estas restricciones, tomamos algunas muestras utilizando una placa de sonido Sound Blaster de 16 bits, y diferenciando entre tres “estilos”: Clásico, Rock y Voz.

Los distintos archivos de audio y sus características físicas puede verse en la *Tabla 7.1*

Archivo	Longitud (Bytes)	Frecuencia de Muestreo	Nro. de Frames	Estilo
Beeta.wav	694.300	44.100	339	Clásico
Beetb.wav	454.024	44.100	222	Clásico
Beetc.wav	406.438	44.100	199	Clásico
Brahms.wav	494.630	44.100	242	Clásico
Brahma.wav	532.332	44.100	260	Clásico
Mozart.wav	727.720	44.100	356	Clásico
Alice.wav	560.799	44.100	274	Rock
Metal.wav	426.746	44.100	209	Rock
Metala.wav	983.454	44.100	481	Rock
El38.wav	1.556.972	44.100	761	Rock
Extreme.wav	494.798	44.100	242	Rock
Extre.wav	223.548	44.100	110	Rock
Kiss.wav	609.718	44.100	298	Rock
Lenny.wav	656.694	44.100	321	Rock
Memph.wav	440.180	44.100	215	Rock
Ray.wav	490.448	44.100	240	Rock
Redo.wav	408.460	44.100	200	Rock
Cielo.wav	857.684	44.100	419	Rock
Divi.wav	245.774	44.100	120	Voz
Ruso.wav	189.172	11.025	93	Voz
Wnoche.wav	970.156	11.025	474	Voz
Wcross.wav	476.442	22.050	233	Voz
Whora.wav	198.476	11.025	97	Voz
Wnani.wav	264.644	22.050	130	Voz
Wmundo.wav	243.750	11.025	119	Voz
Vero.wav	308.744	22.050	151	Voz
Bocha.wav	110.346	22.050	54	Voz
Bochi.wav	264.696	22.050	130	Voz
Bifur.wav	466.816	44.100	228	Voz

Tabla 7.1 Muestras de sonidos

Luego de obtener estas muestras procedimos, mediante la aplicación descrita en el Capítulo 6, a comprimir cada una de estas muestras. Si bien el rango de índices de compresión propuesto por esta herramienta es muy diverso, utilizamos sólo cuatro de estos para comprimir cada uno de los archivos. Las señales de salida, su longitud, y el SNR obtenido para cada estilo de muestra se ven en las *Tablas 7.2, 7.3 y 7.4.*

Archivo	Indice Requerido	Archivo Comprimido	Longitud	Indice Obtenido	SNR	Archivo Reconstruido
Alice.wav	2	Alice2.pak	273.746	2,05	48	Alice2r.wav
Alice.wav	3	Alice3.pak	185.354	3,03	37	Alice3r.wav
Alice.wav	4	Alice4.pak	138.482	4,05	31	Alice4r.wav
Alice.wav	6	Alice6.pak	91.582	6,12	22	Alice6r.wav
Metal.wav	2	Metal2.pak	210.878	2,02	43	Metal2r.wav
Metal.wav	3	Metal3.pak	140.778	3,03	28	Metal3r.wav
Metal.wav	4	Metal4.pak	104.964	4,07	20	Metal4r.wav
Metal.wav	6	Metal6.pak	68.970	6,19	12	Metal6r.wav
Metala.wav	2	Metala2.pak	488.230	2,01	45	Metala2r.wav
Metala.wav	3	Metala3.pak	324.520	3,03	30	Metala3r.wav
Metala.wav	4	Metala4.pak	242.258	4,06	22	Metala4r.wav
Metala.wav	6	Metala6.pak	159.380	6,17	13	Metala6r.wav
EI38.wav	2	EI382.pak	753.002	2,07	47	EI382r.wav
EI38.wav	3	EI383.pak	511.590	3,04	34	EI383r.wav
EI38.wav	4	EI384.pak	383.860	4,06	27	EI384r.wav
EI38.wav	6	EI386.pak	253.548	6,14	19	EI386r.wav
Extreme.wav	2	Extreme2.pak	245.656	2,01	45	Extreme2r.wav
Extreme.wav	3	Extreme3.pak	163.530	3,03	31	Extreme3r.wav
Extreme.wav	4	Extreme4.pak	121.558	4,07	23	Extreme4r.wav
Extreme.wav	6	Extreme6.pak	80.192	6,17	15	Extreme6r.wav
Extre.wav	2	Extre2.pak	111.058	2,01	46	Extre2r.wav
Extre.wav	3	Extre3.pak	73.688	3,03	31	Extre3r.wav
Extre.wav	4	Extre4.pak	55.094	4,06	24	Extre4r.wav
Extre.wav	6	Extre6.pak	36.436	6,14	16	Extre6r.wav
Kiss.wav	2	Kiss2.pak	302.884	2,01	46	Kiss2r.wav
Kiss.wav	3	Kiss3.pak	201.164	3,03	32	Kiss3r.wav
Kiss.wav	4	Kiss4.pak	150.438	4,05	23	Kiss4r.wav
Kiss.wav	6	Kiss6.pak	99.276	6,14	16	Kiss6r.wav
Lenny.wav	2	Lenny2.pak	325.480	2,02	48	Lenny2r.wav
Lenny.wav	3	Lenny3.pak	216.916	3,03	36	Lenny3r.wav
Lenny.wav	4	Lenny4.pak	161.942	4,06	29	Lenny4r.wav
Lenny.wav	6	Lenny6.pak	107.184	6,13	20	Lenny6r.wav
Memph.wav	2	Memph2.pak	218.622	2,01	47	Memph2r.wav
Memph.wav	3	Memph3.pak	145.334	3,03	34	Memph3r.wav
Memph.wav	4	Memph4.pak	108.430	4,06	26	Memph4r.wav
Memph.wav	6	Memph6.pak	71.606	6,15	18	Memph6r.wav
Ray.wav	2	Ray2.pak	241.640	2,03	48	Ray2r.wav
Ray.wav	3	Ray3.pak	161.956	3,03	34	Ray3r.wav
Ray.wav	4	Ray4.pak	121.106	4,05	27	Ray4r.wav
Ray.wav	6	Ray6.pak	80.060	6,13	19	Ray6r.wav
Redo.wav	2	Redo2.pak	200.290	2,04	46	Redo2r.wav
Redo.wav	3	Redo3.pak	134.760	3,03	33	Redo3r.wav
Redo.wav	4	Redo4.pak	100.826	4,05	25	Redo4r.wav
Redo.wav	6	Redo6.pak	66.478	6,14	17	Redo6r.wav
Cielo.wav	2	Cielo2.pak	399.316	2,15	47	Cielo2r.wav
Cielo.wav	3	Cielo3.pak	282.658	3,03	39	Cielo3r.wav
Cielo.wav	4	Cielo4.pak	212.102	4,04	34	Cielo4r.wav
Cielo.wav	6	Cielo6.pak	140.336	6,11	26	Cielo6r.wav

Tabla 7.2 Datos obtenidos en la compresión de muestras de Rock

Archivo	Indice Requerido	Archivo Comprimido	Longitud	Indice Obtenido	SNR	Archivo Reconstruido
Divi.wav	2	Divi2.pak	119.994	2,05	46	Divi2r.wav
Divi.wav	3	Divi3.pak	80.996	3,03	32	Divi3r.wav
Divi.wav	4	Divi4.pak	60.694	4,05	25	Divi4r.wav
Divi.wav	6	Divi6.pak	39.964	6,15	16	Divi6r.wav
Ruso.wav	2	Ruso2.pak	93.854	2,02	39	Ruso2r.wav
Ruso.wav	3	Ruso3.pak	62.380	3,03	24	Ruso3r.wav
Ruso.wav	4	Ruso4.pak	46.178	4,10	16	Ruso4r.wav
Ruso.wav	6	Ruso6.pak	30.536	6,20	8	Ruso6r.wav
Wnoche.wav	2	Wnoche2.pak	412.564	2,35	43	Wnoche2r.wav
Wnoche.wav	3	Wnoche3.pak	316.088	3,07	34	Wnoche3r.wav
Wnoche.wav	4	Wnoche4.pak	239.094	4,06	28	Wnoche4r.wav
Wnoche.wav	6	Wnoche6.pak	158.048	6,14	20	Wnoche6r.wav
Wcross.wav	2	Wcross2.pak	211.134	2,26	45	Wcross2r.wav
Wcross.wav	3	Wcross3.pak	155.056	3,07	36	Wcross3r.wav
Wcross.wav	4	Wcross4.pak	117.392	4,06	31	Wcross4r.wav
Wcross.wav	6	Wcross6.pak	77.924	6,11	22	Wcross6r.wav
Whora.wav	2	Whora2.pak	79.712	2,49	42	Whora2r.wav
Whora.wav	3	Whora3.pak	64.264	3,09	35	Whora3r.wav
Whora.wav	4	Whora4.pak	48.924	4,06	30	Whora4r.wav
Whora.wav	6	Whora6.pak	32.498	6,11	23	Whora6r.wav
Wnanis.wav	2	Wnanis2.pak	110.988	2,38	44	Wnanis2r.wav
Wnanis.wav	3	Wnanis3.pak	85.550	3,09	37	Wnanis3r.wav
Wnanis.wav	4	Wnanis4.pak	65.164	4,06	32	Wnanis4r.wav
Wnanis.wav	6	Wnanis6.pak	43.262	6,12	24	Wnanis6r.wav
Wmundo.wav	2	Wmundo2.pak	116.338	2,10	45	Wmundo2r.wav
Wmundo.wav	3	Wmundo3.pak	80.272	3,04	33	Wmundo3r.wav
Wmundo.wav	4	Wmundo4.pak	60.212	4,05	26	Wmundo4r.wav
Wmundo.wav	6	Wmundo6.pak	39.736	6,13	17	Wmundo6r.wav
Vero.wav	2	Vero2.pak	131.392	2,35	45	Vero2r.wav
Vero.wav	3	Vero3.pak	100.382	3,08	38	Vero3r.wav
Vero.wav	4	Vero4.pak	76.180	4,05	33	Vero4r.wav
Vero.wav	6	Vero6.pak	50.384	6,13	26	Vero6r.wav
Bocha.wav	2	Bocha2.pak	50.260	2,20	46	Bocha2r.wav
Bocha.wav	3	Bocha3.pak	35.954	3,07	37	Bocha3r.wav
Bocha.wav	4	Bocha4.pak	27.166	4,06	32	Bocha4r.wav
Bocha.wav	6	Bocha6.pak	17.938	6,15	24	Bocha6r.wav
Bochi.wav	2	Bochi2.pak	111.178	2,38	44	Bochi2r.wav
Bochi.wav	3	Bochi3.pak	85.738	3,09	37	Bochi3r.wav
Bochi.wav	4	Bochi4.pak	65.336	4,05	32	Bochi4r.wav
Bochi.wav	6	Bochi6.pak	43.324	6,11	25	Bochi6r.wav
Bifur.wav	2	Bifur2.pak	231.960	2,01	42	Bifur2r.wav
Bifur.wav	3	Bifur3.pak	153.914	3,03	27	Bifur3r.wav
Bifur.wav	4	Bifur4.pak	114.858	4,06	18	Bifur4r.wav
Bifur.wav	6	Bifur6.pak	75.716	6,17	10	Bifur6r.wav

Tabla 7.3 Datos obtenidos en la compresión de muestras de Voz.

Archivo	Indice Requerido	Archivo Comprimido	Longitud	Indice Obtenido	SNR	Archivo Reconstruido
Beeta.wav	2	Beeta2.pak	335.880	2,07	50	Beeta2r.wav
Beeta.wav	3	Beeta3.pak	228.582	3,04	43	Beeta3r.wav
Beeta.wav	4	Beeta4.pak	171.344	4,05	37	Beeta4r.wav
Beeta.wav	6	Beeta6.pak	113.276	6,13	28	Beeta6r.wav
Beetb.wav	2	Beetb2.pak	184.318	2,46	45	Beetb2r.wav
Beetb.wav	3	Beetb3.pak	148.120	3,07	39	Beetb3r.wav
Beetb.wav	4	Beetb4.pak	111.668	4,07	34	Beetb4r.wav
Beetb.wav	6	Beetb6.pak	74.230	6,12	26	Beetb6r.wav
Beetc.wav	2	Beetc2.pak	178.540	2,28	48	Beetc2r.wav
Beetc.wav	3	Beetc3.pak	133.626	3,04	44	Beetc3r.wav
Beetc.wav	4	Beetc4.pak	100.530	4,04	39	Beetc4r.wav
Beetc.wav	6	Beetc6.pak	66.524	6,11	31	Beetc6r.wav
Brahms.wav	2	Brahms2.pak	237.768	2,08	48	Brahms2r.wav
Brahms.wav	3	Brahms3.pak	160.404	3,08	38	Brahms3r.wav
Brahms.wav	4	Brahms4.pak	120.948	4,09	33	Brahms4r.wav
Brahms.wav	6	Brahms6.pak	80.696	6,13	23	Brahms6r.wav
Brahma.wav	2	Brahma2.pak	264.268	2,01	49	Brahma2r.wav
Brahma.wav	3	Brahma3.pak	175.884	3,03	38	Brahma3r.wav
Brahma.wav	4	Brahma4.pak	131.534	4,05	32	Brahma4r.wav
Brahma.wav	6	Brahma6.pak	86.794	6,13	22	Brahma6r.wav
Mozart.wav	2	Mozart2.pak	297.006	2,45	45	Mozart2r.wav
Mozart.wav	3	Mozart3.pak	237.176	3,07	40	Mozart3r.wav
Mozart.wav	4	Mozart4.pak	179.760	4,05	35	Mozart4r.wav
Mozart.wav	6	Mozart6.pak	118.980	6,12	25	Mozart6r.wav

Tabla 7.4 Datos obtenidos en la compresión de muestras de música clásica.

Después de promediar la totalidad de los SNRs obtenidos en los tres estilos para cada índice de compresión, elaboramos el siguiente gráfico, *Figura 7.1*, que muestra como se degrada la calidad de reconstrucción a medida que aumentamos el índice de compresión. Los valores para un factor de compresión 5 es un estimado utilizando los obtenidos para los índices 4 y 6.

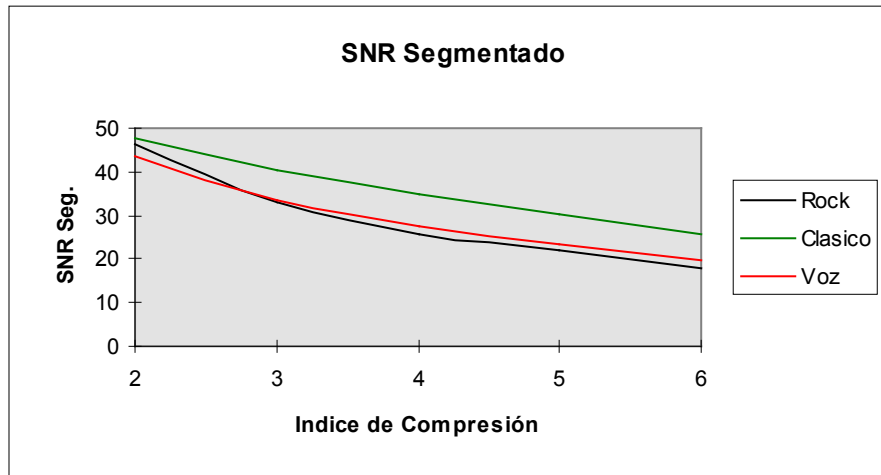


Figura 7.1 Gráfico comparativo de calidad para los diferentes estilos de muestras según el índice de compresión utilizado.

7.3 Conclusiones

Como vemos en el Anexo A, la medida de calidad SNR es una medida objetiva y como tal es un buen predictor de las medidas subjetivas; es decir, de aquellas que miden la calidad percibida por uno o más oyentes. Este hecho puede observarse en la *Figura 7.1*, donde puede verse que a medida que aumentamos el índice de compresión, el valor de SNR obtenido decae, lo cual se corresponde con la calidad de las señales medidas auditivamente. Por otro lado, las muestras de música clásica, si bien obtienen los valores de SNR más elevados en promedio, no son de las que se obtienen las mejores reconstrucciones desde el punto de vista auditivo. Sin duda, si nos ubicamos por un momento en el lugar de un escuchante, las reconstrucciones de música clásica suenan menos “inteligibles” que las de Rock y Voz.

La Transformada Wavelet Empaquetada surge como una herramienta versátil y flexible en su utilización en el análisis de señales. La utilización de una gran variedad de filtros Wavelet en el esquema de descomposición y reconstrucción, la descomposición arbitraria en niveles y la división en subbandas con la consiguiente resolución en tiempo y frecuencia, hacen de esta transformada un elemento apropiado para el procesamiento de señales. Por otro lado, la elección de la Wavelet Básica en cada frame analizado, la cual podría determinarse de acuerdo a la aplicación que este desarrollando, es a nuestro juicio, la característica más importante de esta Transformada.

Dado que la estructura de descomposición Wavelet tiene una estructura paralelizable, se podría idear un implementación en DSP (*Procesador de Señales Digitales*) para que funcione en tiempo real.

Una muestra del creciente interés de los investigadores por esta transformada, es la implementación en VLSI realizada en el CETAD de la Facultad de Ingeniería de la Universidad Nacional de La Plata.

De un tiempo a esta parte, es posible encontrar páginas en Internet dedicadas a divulgar aplicaciones de esta transformada. Un ejemplo es la que posee MathSoft (*mathsoft.com*), donde algunos de los muchos trabajos que se citan son :

- Distintos trabajos introductorios y generales sobre la teoría de Wavelet.

- Su aplicación en el análisis de señales y procesamiento de imágenes.
- Su utilización en la compresión de huellas digitales desarrollado por el FBI.
- El empleo de la Transformada Wavelet en el procesamiento de Speech.
- Transformada Wavelet en el desarrollo matemático.
- Wavelet y Economía.
- Wavelet y Fractales.
- Wavelet y Comunicaciones.

Anexo A

**EVALUACION DE CALIDAD DE LA
VOZ**

EVALUACION DE CALIDAD DE LA VOZ

A.1 Introducción

A.1.1 La Necesidad de Evaluar Calidad

Existen dos áreas del procesamiento de señales de voz en donde la evaluación de la calidad de estas señales es de interés primario: la codificación y el realce. Históricamente, la mayoría de los métodos de testeo de calidad se han formulado para evaluar el desempeño de los algoritmos de codificación de voz. La necesidad de tales métodos llegaron a ser evidentes durante el desarrollo de nuevos sistemas de comunicación analógica para datos de voz en los 1950s. El advenimiento de los algoritmos de codificación digital de señales de voz en los 1960s, condujeron a los sistemas de comunicación de los 1980s y motivaron la formulación de nuevas y más sofisticadas herramientas para la evaluación de calidad, en los sistemas de codificación de bandas vocalizadas. Tales herramientas eran esenciales para el diseño óptimo de sistemas de codificación y un planeamiento de redes de comunicaciones efectivas. Recientemente, muchas de esas pruebas o medidas han sido aplicadas exitosamente para mejorar la cuantificación en algoritmos de realce de voz.

Para establecer medidas justas de comparación de algoritmos de codificación o de realce, se han formulado una gran variedad de técnicas de evaluación de calidad. Generalmente, las pruebas se agrupan en dos clases: medidas de calidad subjetiva y medidas de calidad objetiva. Las medidas subjetivas están basadas en las comparaciones de los datos de la voz procesada y el original por un escuchante o un grupo de escuchantes, quien subjetivamente ranquean la calidad de las señales a lo largo de una escala predeterminada. Las medidas de calidad objetivas están basadas en una comparación matemática de las señales de voz procesada y original. La mayoría de las medidas de calidad objetiva cuantifican la calidad con una distancia numérica o un modelo de como el sistema auditivo interpreta la calidad. Debido a que la distorsión introducida en los sistemas de codificación de voz, el ruido de fondo, y las variaciones de los algoritmos de realce, han emergido un conjunto de medidas de calidad y tests para diferentes aplicaciones.

Generalmente, hay tres áreas en las que es deseable medir la distorsión debido al ruido. Primero, podemos querer una medida de calidad para la cantidad de distorsión introducida por un algoritmo de compresión de voz. Este tipo de ruido puede ser aditivo, pero en muchos casos es no lineal y dependiente de la señal. Tales evaluaciones se pueden usar para determinar los seteos de los parámetros o la alocaión de bits para los parámetros, para mejorar la calidad, o para fomentar la reducción del tamaño de los datos. Segundo, podemos desear medir el nivel de distorsión introducido por un ambiente ruidoso (durante la recolección de la voz) o durante la transmisión sobre un canal ruidoso. En la mayoría de los casos, este ruido es aditivo, a lo largo de toda la banda, y hasta estadísticamente independiente de la señal de voz. Una excepción podría ser una explosión de errores de bits sobre un canal, que puede provenir, por ejemplo, de un enlace satelital de comunicación a causa de cierta actividad solar. Finalmente, podemos desear medir la performance de un algoritmo de realce para ver si la calidad de

la voz procesada se ha mejorado. La consistencia de la señal de entrada, así como también los grupos de escuchantes son esenciales si la performance de los algoritmos son comparados apropiadamente. Dado que el ruido o la distorsión introducida en cada área mencionada es bastante diferente, la aplicabilidad de cada test de calidad variará.

A.1.2 Calidad Versus Inteligibilidad

A fin de evaluar algoritmos de procesamiento de voz, sería útil identificar las similitudes y diferencias entre la calidad percibida y la inteligibilidad medida subjetivamente. La calidad de la voz se refiere a "como" un hablador transmite una expresión y puede incluir ciertos atributos "naturales". En contraste, la inteligibilidad se refiere a lo que el hablador ha pretendido decir o la información contenida detrás de las palabras. En la actualidad no se comprende claramente la interrelación entre calidad y inteligibilidad percibida. Generalmente, las señales de voz ininteligibles no se juzgarían como de calidad alta; sin embargo, la recíproca no es verdadera. Por ejemplo, muchos pronunciamientos mecánicos, sintéticos pueden ser altamente inteligibles. Por lo tanto, la inteligibilidad puede ser considerada como una de muchas "dimensiones" de la noción abstracta de calidad.

La dificultad en separar las nociones de calidad y inteligibilidad se debe, en parte, a la dificultad en aislar y caracterizar esa correlación acústica de calidad o inteligibilidad de la voz. Sin embargo, se han efectuado investigaciones intensivas en tests subjetivos y objetivos para comprobar la calidad e inteligibilidad. Estos tests se han usado extensivamente para evaluar sistemas de codificación/ transmisión de señales de voz.

A.2 Medidas de Calidad Subjetivas

Las medidas subjetivas están basadas en la opinión de un escuchante o un grupo de escuchantes de la calidad de una expresión. Un medio de clasificar las medidas de calidad subjetiva es agruparlas como utilitarias o analíticas [Hecker y Williams, 1966]. Las medidas utilitarias emplean procedimientos de testeos que son eficientes y confiables, y que producen una medida de calidad de voz sobre una escala unidimensional. La ventaja principal es que el resultado es un único número, el cual puede usarse para comparar los sistemas de procesamiento de señales de voz.

En contraste, los métodos analíticos buscan identificar los componentes psicológicos subyacentes que determinan la calidad percibida. Estos métodos se orientan más hacia caracterizar la percepción de la voz que a medir la calidad percibida y casi siempre usan más de una de dimensión para reportar los resultados. Como hemos notado, la inteligibilidad puede ser vista como un aspecto de calidad, dado que una señal de voz de alta calidad implica generalmente buena inteligibilidad. Sin embargo, la recíproca no es necesariamente cierta. Los test de calidad (a excepción de los que miden inteligibilidad) son empleados comunmente para evaluar sistemas con un contenido alto de inteligibilidad, dado que una inteligibilidad baja es generalmente un buen indicador de mala calidad.

Los test y sus siglas se resumen en la *Tabla A.1*. Las primeras medidas subjetivas hicieron incapié en la inteligibilidad de la voz, un aspecto importante de la calidad total. Algunos test se formularon usando listas de palabras que riman, tal como *Modified*

Rhyme Test (MRT) y *Diagnostic Rhyme Test* (DRT). Aquí los escuchantes son presentados con palabras que riman, que difieren solamente en sus principales fonemas consonantes. La calidad del sistema de procesamiento de voz se basa en el número y posiblemente la facilidad de los escuchantes de responder correctamente. Otra clase de medida subjetiva que intenta evaluar de una forma más simple la inteligibilidad de la voz es el test de calidad *Isometric Absolute Judgment* (IAJ). En estos test, los escuchantes juzgan la calidad total de una lista de palabras distorcionadas. En este caso, "isométric" significa que los escuchantes juzgan la calidad sin ninguna señal de referencia. La ventaja aquí es que los escuchantes son libres de asignar cualquier medida de percepción a su escala, variando desde excelente a insatisfactorio.

Test	Type of Test
Modified rhyme test (MRT)	Subjective intelligibility
Diagnostic rhyme test (DRT)	Subjective intelligibility
Isometric absolute judgment (IAJ)	Subjective quality
Mean opinion score (MOS)	Subjective quality
Paired acceptability rating (PAR)	Subjective quality
Parametric absolute judgment (PAJ)	Subjective quality
Quality acceptance rating test (QUART)	Overall subjective quality
Diagnostic acceptability measure (DAM)	Overall subjective quality
Articulation index (AI)	Objective intelligibility
Signal-to-noise ratio (SNR)	Objective quality
Segmental SNR (SNRseg)	Frame-based SNR
Frequency weighted segmental SNR (SNRfw-ses)	Frame-based SNR with spectral weighting
Itakura log-likelihood measure	Objective quality
Log-area ratio measure (LAR)	Objective quality
Other LP-based measures	Objective quality
Weighted-spectral slope measure (or Klattmeasure) (WSSM)	Objective quality

Tabla A.1: Medidas de Calidad

La desventaja de estas pruebas es que pueden haber variaciones o influencias entre los escuchantes. Por lo tanto, un escuchante puede ranquear siempre la voz distorcionada en algún nivel promedio inferior que otro escuchante. Para determinar esta predisposición, algunos investigadores presentan un conjunto estandar de expresiones distorcionadas al escuchante. Se asume que esta predisposición es constante para esta sección y se usa como una compensación para evaluar la voz distorcionada en cuestión. Algunos de estos test incluyen el test "bondadoso", *Mean Opinion Score* (MOS), y el método *Paired Acceptability Rating* (PAR). Otras clases de medidas de calidad subjetivas que se orientan al problema de las diferencias entre las preferencias individuales es el test de calidad *Parametric Absolute Judgment* (PAJ). Aquí, los escuchantes juzgan la voz distorcionada en varias escalas de percepción en vez de juzgar la calidad total. Por ejemplo, estas escalas podrían describir el grado de percepción de un efecto en la voz. Estos test son efectivos, dado que los escuchantes frecuentemente concuerdan en el grado de degradación pero varían en su rechazo a una degradación

específica. Un valor de calidad total puede encontrarse combinando los valores paramétricos. Dos de estos tests incluyen el *Quality Acceptance Rating Test* (QUART), y el muy exitoso *Diagnostic Acceptability Measure* (DAM).

A.3 Medidas de Calidad Objetiva

La continua investigación en codificación y realce de voz ha necesitado de técnicas de evaluación de calidad mejoradas. Dado que el objetivo de codificar o realzar es producir señales de voz que sean percibidas por el sistema auditivo como naturales y libres de degradación, es comprensible que las medidas de calidad subjetiva sean los medios preferibles de evaluación de calidad. Sin embargo, dado que los algoritmos de compresión y realce de voz son cada vez más complejos, es necesario distinguir entre las diferencias más sutiles en la calidad de la voz procesada. Además, las medidas subjetivas sirven generalmente como un medio para obtener una medida grosera de la performance. De acuerdo a las medidas subjetivas las diferencias de calidad deben ser lo suficientemente grandes para ser distinguibles por el grupo escuchante. Si existen solamente diferencias de calidad marginales, puede ser difícil fijar los parámetros del algoritmo para la calidad resultante. El testeo subjetivo requiere recursos de personal y tiempo significativos. Las medidas de calidad de voz objetivas son confiables, fáciles de implementar, y se ha visto que son buenos predictores de calidad subjetiva.

La performance del criterio para una medida de calidad de voz objetiva es su correlación con la estimación de la calidad subjetiva. Para obtener el coeficiente de correlación, las medidas de calidad subjetiva y objetiva deben aplicarse a una base de datos de señales de voz procesadas. Un análisis de correlación se aplica para determinar la capacidad de la medida de calidad objetiva para predecir la calidad como ha sido juzgada por los escuchantes en la evaluación subjetiva. Tales procedimientos de evaluación se realizaron en laboratorios de investigación sobre bases de datos extensivas.

La evaluación de calidad por medio de una medida objetiva provee una medida precisa, cuantitativa, y repetitiva de comparación. Todas las medidas objetivas hacen una comparación directa entre una forma de onda original (o de referencia) y la versión procesada (resintetizada o realzada). Dado que la comparación es directa, es necesario que las formas de onda de la señal de voz procesada y original se sincronicen. Un método común que implementa una medida objetiva es particionar la señal en frames de 10-30 msec de duración, entonces calcular una medida de distancia/distorsión para cada frame. La mayoría de las medidas “pesan” las diferencias en las características espectrales entre los datos procesados y originales. Una medida final se forma combinando las medidas de distorsión en cada frame. Se usan una o más condiciones matemáticas en la formulación de las medidas objetivas.

A.3.1 Relación Señal - Ruido (SNR)

El SNR es la medida más usada para sistemas de codificación de forma de onda y también ha sido usada en el pasado para la evaluación de algoritmos de realce y distorsiones de ruido en toda la banda. Existen variantes del método, incluyendo SNR clásico, SNR segmentado, y SNR segmentado de peso frecuencial, por nombrar algunos pocos. Es importante notar que las medidas basadas en SNR son únicamente apropiadas

para sistemas de codificación o realce que buscan reproducir la forma de onda de entrada original. Sea $y(n)$ una señal de voz ruidosa de tiempo n , $s(n)$ su equivalente libre de ruido y $\hat{s}(n)$ la correspondiente señal procesada o realizada. Se asume que estas señales son de energía. El error en el tiempo puede escribirse como

$$e(n) = s(n) - \bar{s}(n) \quad [A.1]$$

La energía del error es entonces

$$E_e = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} [s(n) - \bar{s}(n)]^2 \quad [A.2]$$

La energía contenida en la señal de voz es

$$E_s = \sum_{n=-\infty}^{\infty} s^2(n) \quad [A.3]$$

La medida SNR resultante(en dB) se obtiene como

$$SNR = 10 \log_{10} \left(\frac{E_s}{E_e} \right) = 10 \log_{10} \left(\frac{\sum_n s^2(n)}{\sum_n [s(n) - \bar{s}(n)]^2} \right) \quad [A.4]$$

Como en el caso de muchas medidas de calidad que se usaron para evaluar sistemas de codificación o realce, la señal de voz original $s(n)$ se necesita para determinar las mejoras. Por lo tanto, tales medidas son usadas primariamente en simulación donde la degradación y las señales libres de ruido están disponibles.

El principal beneficio de la medida de calidad SNR es su simplicidad matemática. La medida representa un error promedio en tiempo y frecuencia para una señal procesada. El SNR clásico es un estimador pobre de calidad de voz para un rango amplio de distorsiones de voz. Esto es porque el SNR no está relacionado a ningún atributo subjetivo de calidad de la voz, y porque el SNR pesa igualmente todos los errores en el dominio del tiempo en la señal. La energía de la señal es en general variante en el tiempo. Si asumimos que las distorsiones por el ruido son en toda la banda con poca fluctuación de energía, entonces la medida SNR seguro debería variar de frame en frame. Una medida SNR engañosa puede obtenerse si una expresión contiene una concentración alta de segmentos vocalizados, dado que el ruido tiene un efecto de percepción mayor en los segmentos de baja energía (p. ej., fricativas no vocalizadas). Una medida de calidad mejorada puede obtenerse si el SNR se mide sobre frames cortos y los resultados promediados. La medida basada en frames se llama *Segmental SNR* (SNR_{seg}), y su fórmula es

$$SNR_{seg} = \frac{1}{M} \sum_{j=0}^{M-1} 10 \log_{10} \left[\sum_{n=m_j-N+1}^{m_j} \frac{s^2(n)}{[s(n) - \bar{s}(n)]^2} \right] \quad [A.5]$$

donde m_0, m_1, \dots, m_{M-1} son los puntos finales de los M frames, cada uno de los cuales es de longitud N . Para cada frame (de 15-25 mseg), una medida SNR se computa y la medida final se obtiene por promediación de estas medidas sobre todos los segmentos de la forma de onda. La segmentación del SNR permite a las medidas objetivas asignar igual peso a las porciones suaves y fuertes de la señal.

En algunos casos, los problemas surgen con la medida SNR_{seg} si se incluyen frames de silencio, dado que el resultado del SNR_{seg} es muy negativo. Este problema es aliviado identificando períodos de silencio y excluyéndolos de los cálculos del SNR_{seg} . Otro enfoque está en colocar un umbral pequeño y reemplazar todos los frames con SNR_{seg} bajo, con este umbral (por ej., un umbral de 0 dB es razonable). Esto previene que la medida sea mala por unos pocos frames de silencio. En el otro extremo, los frames con medidas de SNR_{seg} mayores que 35 dB no son percibidos por los escuchantes como significativamente diferentes. Por lo tanto, un umbral superior (normalmente 35 db) es usado para reajustar cualquier medida de frame SNR_{seg} extraordinariamente alta. Por medio de los dos umbrales se previene que la medida de SNR_{seg} final sea parcial en dirección positiva o negativa dados unos pocos frames que no contribuyen significativamente a la calidad de la señal total.

La medida SNR final que presentamos es la *frequency weighted segmental SNR* (SNR_{fw-seg}). Se han formulado distintas variantes [Tribolet, 1978; Barnwell, 1980] en busca de obtener un SNR fragmentado dentro de un conjunto de bandas de frecuencia proporcionalmente espaciadas. Tal medida permite una serie de pesos de percepción para ser aplicada en cada banda y por medio de estos producir una medida SNR mejor relacionada a la noción de calidad percibida por los escuchantes. La medida se forma como se indica a continuación:

$$SNR_{fw-seg} = \frac{1}{M} \sum_{j=0}^{M-1} 10 \log_{10} \left[\frac{\sum_{k=1}^K w_{j,k} 10 \log_{10} \left[\frac{E_{s,k}(m_j)}{E_{e,k}(m_j)} \right]}{\sum_{k=1}^K w_{j,k}} \right] \quad [A.6]$$

donde M representa el número de frames de la señal (indexados por m_1, \dots, m_{M-1}), K el número de bandas de frecuencia, $E_{s,k}(m_j)$ la energía de la señal a corto plazo en la k -ésima banda de frecuencia para el frame de la señal libre de ruido indexada por m_j , y $E_{e,k}$ la cantidad similar para la secuencia de ruido $e(n)$. Los estudios han demostrado que la medida SNR_{fw-seg} es un mejor predictor de calidad de voz que el SNR clásico o que el SNR_{seg} .

Anexo B

FILTROS ESPEJO EN CUADRATURA

FILTROS ESPEJO EN CUADRATURA

El bloque básico para construir aplicaciones de filtros espejo en cuadratura (QMF) es un banco QMF de dos canales como se muestra en la *Figura B.1*. Esta estructura emplea dos decimadores en la sección de análisis y dos interpoladores en la de síntesis. Los filtros pasa-bajos y pasa-altos en la sección de análisis tienen respuestas al impulso $h_0(n)$ y $h_1(n)$, respectivamente. De la misma manera, los filtros pasa-bajos y pasa-altos contenidos en la sección de síntesis tienen respuestas al impulso $g_0(n)$ y $g_1(n)$, respectivamente.

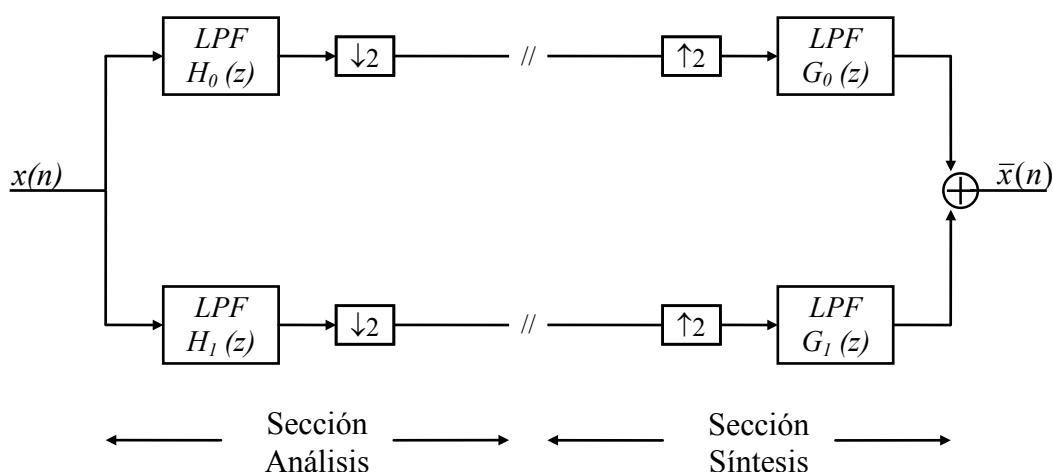


Figura B.1: Banco QMF de dos canales.

Para un decimador de factor M la relación entre la transformada de Fourier de las señales de entrada y salida, $X(w)$ e $Y(w)$ respectivamente, es:

$$Y(w) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{w-2\pi k}{M}\right) \quad [\text{B.1}]$$

y para un interpolador de factor M es:

$$Y(w) = X(Mw) \quad [\text{B.2}]$$

Supongamos que $X(w)$ es la DTFT (Transformada de Fourier Discreta en Tiempo) de $x(n)$. Entonces la salida de los filtros pasa-bajos y pasa-altos será respectivamente:

$$Y_0(w) = X(w)H_0(w)$$

$$Y_1(w) = X(w)H_1(w)$$

Luego del proceso de decimación las salidas correspondientes son:

$$X_{a,0}(w) = \frac{1}{2} \sum_{k=0}^I Y_0\left(\frac{w-2\pi k}{M}\right) \quad (\text{Por [B.1]})$$

$$= \frac{1}{2} \left[Y_0\left(\frac{w}{2}\right) + Y_0\left(\frac{w-2\pi}{2}\right) \right]$$

$$= \frac{1}{2} \left[X\left(\frac{w}{2}\right)H_0\left(\frac{w}{2}\right) + X\left(\frac{w-2\pi}{2}\right)H_0\left(\frac{w-2\pi}{2}\right) \right] \quad [\text{B.3}]$$

$$X_{a,1}(w) = \frac{1}{2} \sum_{k=0}^I Y_1\left(\frac{w-2\pi k}{M}\right) \quad (\text{Por [B.1]})$$

$$= \frac{1}{2} \left[Y_1\left(\frac{w}{2}\right) + Y_1\left(\frac{w-2\pi}{2}\right) \right]$$

$$= \frac{1}{2} \left[X\left(\frac{w}{2}\right)H_1\left(\frac{w}{2}\right) + X\left(\frac{w-2\pi}{2}\right)H_1\left(\frac{w-2\pi}{2}\right) \right] \quad [\text{B.4}]$$

Si $X_{s,0}(w)$ y $X_{s,1}(w)$ representan las dos entradas a la sección de síntesis, entonces la salida de los respectivos interpoladores es:

$$Y_0(w) = X_{s,0}(2w) \quad \text{Por [B.2]}$$

$$Y_1(w) = X_{s,1}(2w) \quad \text{Por [B.2]}$$

Luego, la salida de los respectivos filtros pasa-bajos y pasa-altos en la sección de síntesis será:

$$X_0(w) = Y_0(w)G_0(w) = X_{s,0}(2w)G_0(w)$$

$$X_1(w) = Y_1(w)G_1(w) = X_{s,1}(2w)G_1(w)$$

Por lo tanto

$$\bar{X}(w) = X_0(w) + X_1(w)$$

$$\bar{X}(w) = X_{s,0}(2w)G_0(w) + X_{s,1}(2w)G_1(w) \quad [\text{B.5}]$$

Si conectamos el filtro de análisis o su correspondiente filtro de síntesis, de modo que $X_{a,0}(w)=X_{s,0}(w)$ y $X_{a,1}(w)=X_{s,1}(w)$, entonces sustituyendo [B.3] y [B.4] en [B.5]

$$\bar{X}(w) = X_{a,0}(2w)G_0(w) + X_{a,1}(2w)G_1(w)$$

$$\begin{aligned}
 &= \frac{1}{2} [X(w)H_0(w) + X(w-\pi)H_0(w-\pi)]G_0(w) \\
 &+ \frac{1}{2} [X(w)H_1(w) + X(w-\pi)H_1(w-\pi)]G_1(w) \\
 &= \frac{1}{2} [X(w)H_0(w)G_0(w) + X(w-\pi)H_0(w-\pi)G_0(w)] \\
 &+ \frac{1}{2} [X(w)H_1(w)G_1(w) + X(w-\pi)H_1(w-\pi)G_1(w)] \\
 &= \frac{1}{2} X(w) [H_0(w)G_0(w) + H_1(w)G_1(w)] \\
 &+ \frac{1}{2} X(w-\pi) [H_0(w-\pi)G_0(w) + H_1(w-\pi)G_1(w)] \quad [\text{B.6}]
 \end{aligned}$$

El primer término en [B.6] es la señal de salida deseada del banco de filtros QMF. El segundo término representa los efectos del *aliasing* los cuales deberán eliminarse. Por lo tanto requerimos que:

$$H_0(w-\pi)G_0(w) + H_1(w-\pi)G_1(w) = 0 \quad [\text{B.7}]$$

Esta condición se satisface seleccionando $G_0(w)$ y $G_1(w-\pi)$ como:

$$\begin{aligned}
 G_0(w) &= H_1(w-\pi) \\
 G_1(w) &= -H_0(w-\pi) \quad [\text{B.8}]
 \end{aligned}$$

De este modo el segundo término en [B.6] desaparece.

Asumamos ahora que $H_0(w)$ es un filtro pasa bajo y $H_1(w)$ es un filtro pasa alto de imagen espejo. Entonces podemos expresar $H_0(w)$ y $H_1(w)$ como:

$$\begin{aligned}
 H_0(w) &= H(w) \\
 H_1(w) &= H(w-\pi) \quad [\text{B.9}]
 \end{aligned}$$

donde $H(w)$ es la respuesta frecuencial de un filtro pasa bajo. En el dominio del tiempo, las relaciones correspondientes son:

$$\begin{aligned}
 h_0(n) &= h(n) \\
 h_1(n) &= (-1)^n h(n) \quad [\text{B.10}]
 \end{aligned}$$

La *Figura B.2* muestra la simetría de imagen espejo de los filtros $H_0(w)$ y $H_1(w)$ alrededor de las frecuencias $w = \pi/2$

Para ser consistente con la restricción [B.8], seleccionamos el filtro pasa bajo G_0 y el filtro pasa alto G_1 como:

$$G_0(w) = 2H(w)$$

$$G_1(w) = -2H(w - \pi) \quad [B.11]$$

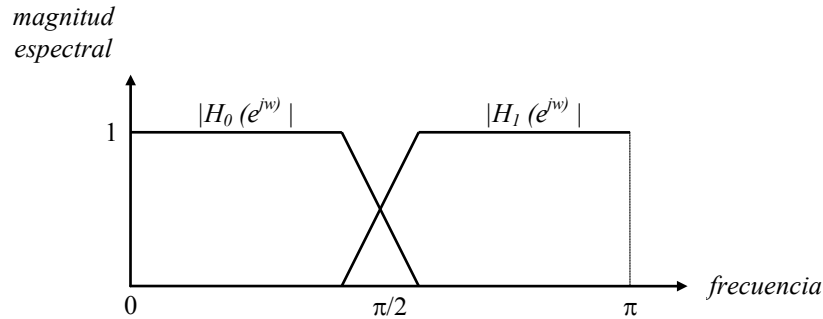


Figura B.2: respuesta en frecuencia ideal característica de un filtro QMF

En el dominio temporal estas relaciones serán

$$g_0(n) = 2h(n)$$

$$g_1(n) = -2(-1)^n h(n) \quad [B.12]$$

El factor 2 en $g_0(n)$ y $g_1(n)$ corresponde al factor de interpolación que es usado para normalizar la respuesta frecuencial total del QMF. Con esta elección de las características de los filtros, las componentes debido al *aliasing* desaparece. De este modo el *aliasing* resultante de la decimación es cancelado totalmente en el proceso de interpolación. Así los filtros QMF de dos canales se comportan como un sistema lineal invariante en el tiempo.

Si sustituimos $H_0(w), H_1(w), G_0(w)$ y $G_1(w)$ en el primer término de [B.6] obtenemos:

$$\bar{X}(w) = [H^2(w) - H^2(w - \pi)]X(w) \quad [B.13]$$

Idealmente el banco QMF de dos canales deberá tener ganancia unitaria, es decir:

$$|H^2(w) - H^2(w - \pi)| = 1 \quad [B.14]$$

También es deseable que éstos tengan fase lineal. Consideremos el uso de un filtro de fase lineal $H(w)$, el cual puede ser expresado de la forma:

$$H(w) = H_r(w)e^{-jw(N-1)/2} \quad [B.15]$$

donde $H_r(w)$ es la DTFT de la versión no retardada de $h(n)$ y N es la duración de la respuesta de impulso del filtro. Entonces:

$$H^2(w) = H_r^2(w)e^{-jw(N-1)} \quad [\text{B.16}]$$

$$H^2(w) = |H(w)|^2 e^{-jw(N-1)}$$

y

$$H^2(w - \pi) = H_r^2(w - \pi)e^{-j(w-\pi)(N-1)} \quad [\text{B.17}]$$

$$H^2(w - \pi) = (-1)^{N-1} |H(w - \pi)|^2 e^{-jw(N-1)}$$

De este modo la función de transferencia total de los QMF de dos canales que emplean filtros de fase lineal FIR es:

$$\frac{\bar{X}(w)}{X(w)} = \left[|H(w)| - (-1)^{N-1} |H(w - \pi)|^2 \right] e^{-jw(N-1)}$$

Obsérvese que este filtro tiene un retardo de $(N-1)$ muestras y una magnitud característica:

$$A(w) = |H(w)|^2 - (-1)^{N-1} |H(w - \pi)|^2$$

- Si N es impar, entonces $A(\pi/2) = 0$ dado que $|H(\pi/2)| = |H(3\pi/2)|$, esta es una propiedad indeseada para el diseño de un QMF.
- Si N es par:

$$A(w) = |H(w)|^2 + |H(w - \pi)|^2$$

Esto evita el problema de un cero en $\pi/2 = w$. Para N par el QMF de dos canales ideal debe satisfacer la condición:

$$A(w) = |H(w)|^2 + |H(w - \pi)|^2 = 1 \quad [\text{B.18}]$$

Desafortunadamente la única función de respuesta frecuencial de un filtro que satisface [B.18] es una en la cual el espectro de magnitud trivial es:

$$H(w) = \cos^2(aw)$$

Consecuentemente, cualquier filtro FIR de la fase lineal no trivial $H(w)$ introducirá alguna distorsión de amplitud. Esta puede ser minimizada optimizando los coeficientes del filtro. Un método efectivo es diseñar los coeficientes del filtro $H(w)$ tal que $A(w)$ sea lo más plana posible minimizando simultáneamente la energía de stop-band de $H(w)$. Esta aproximación lleva a la minimización del error cuadrático medio.

$$\xi = w \int_{w_s}^{\pi} |H(w)|^2 dw + (1-w) \int_0^{\pi/2} [A(w) - I]^2 dw \quad [\text{B.19}]$$

donde w es un factor de peso en el rango de $0 < w < 1$ y w_s es la frecuencia stop-band. En la optimización el filtro FIR debe ser simétrico (de fase lineal).

Como alternativa al uso de filtros de fase lineal FIR, se pueden diseñar filtros IIR que satisfagan la restricción dada por la ecuación [B.14].

Dado que los QMF podrían introducir alguna distorsión de fase, la señal de salida del QMF puede ser pasada a través de un ecualizador de fase todo-pasa diseñado para minimizar la distorsión de fase.

Bibliografía

Bibliografía:

- L. Rabiner and R. Schafer, *Digital Processing of Speech Signal*, Prentice Hall, 1978.
- S. N. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice Hall 1984.
- A. Oppenheim and R. Schafer, *Discrete Time Signal Processing*, Prentice-Hall, 1989.
- J. R. Deller Jr. et. al., *Discrete-Time Processing of Speech Signals*, Macmillan, 1993.
- J. Flanagan et. al., *Speech Coding*, IEEE Transaction on Communications, Abril 1979.
- P. P. Vaidyanathan and P. Q. Hoang, *Lattice Structures for Optimal Design and Robust Implementation of Two-Channel Perfect-Reconstruction QMF Banks*, IEEE Transaction on Acoustic, Speech and Signal Processing, January 1988.
- M. V. Wickerhauser, *Acoustic Signal Compression with wave packets*, Yale University, Agosto 1989.
- F. Safar, F. Blasetti, J. P. Villa, D. Cocimano y S. Katz, *Representación Adaptativa de Imágenes en Bases Ortogonales Optimas y Aplicaciones a Compresión*, Universidad Nacional de La Plata, 1991.
- O. Rioul and P. Duhamel, *Fast Algorithms for Discrete and Continuous Wavelets Transforms*, IEEE Transaction on Information Theory, Marzo 1992.
- M. A. Cody, *The Fast Wavelet Transform*, Dr. Dobb's Journal, Abril 1992.
- M. A. Cody, *The Wavelet Packet Transform*, Dr Dobb's Journal, Abril 1994.
- A. Gersho, *Advances in Speech and Audio Compression*, Proceeding of the IEEE, Junio 1994.



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

DONACION..... TES
\$..... 96/3
Fecha..... 19-8-05
Inv. E..... Inv. B..... 1927
+ DRS 23-24