



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

# TRABAJO DE GRADO

*“Compresión Digital de Audio en el Espacio  
Transformado”*

Autores:



Ramiro Fabián Sosa  
Pablo Javier Thomas

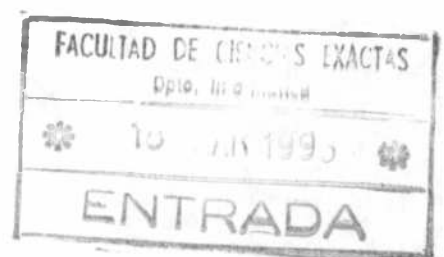
Director:

Ing. Oscar Norberto Bria

Facultad de Ciencias Exactas  
Universidad Nacional de La Plata

1996

<b>TES</b> <b>96/11</b> <b>DIF-01956</b> <b>SALA</b>	 <p>UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMÁTICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p>  <p>DIF-01956</p>
---	---



# Dedicatoria

*A mis padres.*

R.F.S.

*A mi familia.*

P.J.T.

# Indice



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

**Prólogo** ..... iv

## 1 Introducción

1.1 Reseña ..... 1  
 1.2 Objetivo General ..... 1  
 1.3 Motivación ..... 2  
 1.4 Desarrollo ..... 2

## 2 Conceptos Básicos

2.1 Introducción ..... 4  
 2.2 Medios Naturales y Ondas ..... 4  
     2.2.1 Ondas Sonoras ..... 4  
     2.2.2 Frecuencia ..... 5  
     2.2.3 Amplitud ..... 5  
     2.2.4 Suma de Formas de Onda ..... 5  
     2.2.5 Fase ..... 6  
     2.2.6 Tipo de Señales ..... 6  
 2.3 Sonido ..... 9  
     2.3.1 Frecuencia y Tono ..... 9  
     2.3.2 Amplitud e Intensidad del Sonido ..... 9  
     2.3.3 Rúbricas Sonoras ..... 10  
     2.3.4 Armónicos ..... 10  
     2.3.5 Decibel ..... 10  
     2.3.6 Señales Digitales ..... 10  
     2.3.7 Relación Señal-Ruido ..... 12  
     2.3.8 Codificadores de Señales de Audio ..... 13

### ***3 Codificadores de Forma de Onda en el Dominio Temporal***

3.1	Introducción .....	15
3.2	Pulse Code Modulation (PCM) .....	16
3.2.1	Frecuencia de Muestreo.....	17
3.2.2	Cantidad de Bits por Muestra .....	18
3.2.3	Características de Cuantización .....	18
3.3	Logarithmic Pulse Code Modulation (LogPCM) .....	19
3.4	Differential Pulse Code Modulation (DPCM) .....	20
3.5	Adaptative Differential Pulse Code Modulation (ADPCM).....	21
3.6	Delta Modulation (DM) .....	21
3.7	Linear Predictive Coding (LPC) .....	22
3.8	Métodos de Compresión Analizados en el Dominio del Tiempo.....	23
3.8.1	Truncamiento y Conexión de Dos Muestras.....	23
3.8.2	Análisis de Diferencias.....	24
3.8.3	Análisis de Diferencias con Exponente Fijo.....	24
3.8.4	Análisis de Diferencias con Exponente Variable.....	24

### ***4 Codificación de Forma de Onda en el Dominio Frecuencial***

4.1	Introducción .....	26
4.2	Block Transform Coding (BTC).....	26
4.2.1	Generalidades.....	26
4.2.2	Transformaciones Lineales.....	28
4.2.3	Técnicas de Codificación .....	30
4.2.4	Alocación Óptima de Bits.....	31
4.2.5	Problemas de Codificación Usando BTC .....	39

### ***5 Codificación en el Espectro Utilizando BTC***

5.1	Etapas de Codificación en el Espectro .....	41
-----	---	----

---

5.1.1 Cambio de Dominio Temporal al Frecuencial.....	41
5.1.2 Codificación de Líneas Espectrales .....	42
5.1.3 Decodificación de Líneas Espectrales.....	51
5.1.4 Cambio del Dominio Frecuencial al Temporal.....	51
5.2 Criterios de Fidelidad .....	51
5.3 Resultados Obtenidos.....	52
5.4 Limitaciones Prácticas en PC. Implementaciones en Tiempo Real .....	58
<b>6 Conclusión .....</b>	<b>59</b>
<b>7 Bibliografía .....</b>	<b>60</b>
<b>8 Agradecimientos .....</b>	<b>63</b>



# Prólogo

La evolución tecnológica de los sistemas digitales ha hecho posible avances revolucionarios, tanto en lo que respecta al tratamiento de la información audiovisual, como en el ámbito de las comunicaciones. Pero esta evolución de ningún modo habría sido alcanzada sin un desarrollo significativo en el campo del Procesamiento Digital de Señales (PSD), el cual debe su progreso a los estudios llevados a cabo en disciplinas tales como la Matemática Aplicada, la Electrónica y la Informática. Una aplicación inmediata del PSD es la representación de fenómenos del mundo real (de naturaleza analógica) mediante el uso de computadoras digitales. Un problema de importancia desde el punto de vista tecnológico que surge en la representación de éstos, es la gran cantidad de espacio que requiere tanto para su almacenamiento en un medio magnético u óptico, como para su transmisión a través de un canal con ancho de banda limitado.

En el presente trabajo se considera el fenómeno físico del sonido, el análisis de su representación digital, el estudio de un método conveniente de codificación (*Block Transform Coding*) basado en una teoría matemática de complejidad media, y finalmente la concreción de una implementación de dicho método que permita desarrollar una técnica de compresión, basándose en las características que le son propias.

Para la implementación de los algoritmos desarrollados se recurrió a la infraestructura brindada por el *Laboratorio de Investigación y Desarrollo en Informática (L.I.D.I.)*, que nos proporcionó un equipo PC que cuenta con una tarjeta de sonido de 16 bits. La codificación de los mismos se efectuó en lenguaje C, para lo cual se utilizó un compilador *Borland C++ versión 3.00* en ambiente DOS.

Para la realización de las diferentes pruebas se obtuvieron archivos de señales de sonido de colecciones de voces y efectos sonoros previamente digitalizados y de registros de discos compactos de audio. Las ediciones de las señales de audio con las que trabajamos se efectuaron con un software para edición de ondas denominado *Goldwave version 2.0*.

Por último sólo quisiéramos agregar a modo de reflexión sobre la tarea realizada lo siguiente, el emprendimiento de una investigación en el área de PSD ha resultado ser una experiencia enriquecedora puesto que nos permitió poner en práctica conceptos previos y nuevos del campo de la Matemática para obtener resultados e implementaciones concretas sobre computadoras digitales.

# Capítulo 1 : INTRODUCCION



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

## 1.1 Reseña

La manipulación de audio por medio de computadoras, es un desarrollo relativamente nuevo. Tal manipulación ha sido posible desde el nacimiento de las computadoras digitales, pero sólo durante los últimos cinco años, la disminución de los costos del hardware ha permitido que esta tecnología llegue a los escritorios de los usuarios. Ahora es de esperar que la posibilidad de escuchar sonido digital se transforme en una parte integral de la inminente "Revolución Multimedia". [NEL94]

Como las tecnologías digitales evolucionan, y las economías de integración de gran escala tienden a ser alcanzadas, se renueva el interés por métodos eficientes de codificación digital de audio. [FLA79]

Desde el punto de vista de los usuarios, un tema obligado en el uso de Multimedia es la Compresión de Datos. En el caso particular de audio digitalizado, los requerimientos de almacenamiento son muy elevados. Para lograr calidad de audio CD se necesita muestrear 44100 veces por segundo, con una resolución de 16 bits por cada muestra. En stereo esto lleva a 1,4 Mbits por segundo, con lo cual se ve la imperiosa necesidad de comprimir. [THO94]

## 1.2 Objetivo General

Estudiar la compresión digital de audio en el espacio transformado, poniendo énfasis en el problema de codificación y decodificación de señales de audio, así como el almacenamiento y recuperación de las mismas.



Esta clase de problemas que podríamos clasificar dentro del análisis de señales de audio, permite investigar algoritmos eficientes de almacenamiento de señales, utilizando técnicas de compresión específicas.

## 1.3 Motivación

La codificación de audio de sucesivos bloques en el espacio transformado, es una de las más eficientes que se conoce, en lo referente a buena calidad y alta compresión.

El estudio de la compresión digital de audio en el espacio transformado es un excelente ejemplo de:

1. La utilidad práctica de las propiedades matemáticas de las transformaciones lineales.
2. Cómo una teoría medianamente compleja, es útil para predecir resultados de asignación del óptimo número de bits para la cuantización de números reales en un caso concreto.
3. Cómo esa teoría debe y puede adaptarse a la situación práctica de precisión finita de las máquinas digitales.
4. Un algoritmo de procesamiento digital de señales de complejidad computacional media.

## 1.4 Desarrollo

En este trabajo primeramente, se realiza una revisión de conceptos generales. Se presentan además, distintos métodos de compresión en el dominio temporal.

Luego, se tratan los fundamentos teóricos de la codificación de señales de audio en el espacio transformado.

Posteriormente desarrollamos un método de compresión en lenguaje C y varios ejemplos de codificación de señales de audio reales, para diversos factores de compresión.

Se expresan las limitaciones prácticas que aparecen cuando se intenta implementar los resultados teóricos en una máquina digital.

Finalmente, se realizan sugerencias en cuanto a qué hacer para mejorar la calidad de reconstrucción del esquema, y cómo lograr una implementación en tiempo real.

# Capítulo 2 : CONCEPTOS BASICOS

## 2.1 Introducción

Uno de los aspectos más notables del trabajo con combinaciones de medios es su “*similitud*”. Todas las herramientas electrónicas (equipos de audio, video, y cómputo) operan según los mismos principios básicos, y la forma como se comportan refleja el comportamiento de los medios naturales.

## 2.2 Medios Naturales y Ondas

### 2.2.1 Ondas Sonoras

Las *ondas sonoras* se propagan en la materia, como por ejemplo el aire, la madera y el metal. Mediante un diapasón podemos mostrar la forma como se crean y transmiten las ondas sonoras. Al golpearlo su brazo se mueve en dirección del impacto; las moléculas de aire de esa zona se comprimen y presentan alta energía potencial.

Como la naturaleza tiende a un estado de equilibrio, las moléculas se ven obligadas a descomprimirse hacia afuera. La energía dinámica gastada durante la descompresión hace que los grupos vecinos de moléculas se compriman hasta un estado de alta energía potencial. Este proceso se repite en una especie de efecto dominó, empujando aire comprimido en todas las direcciones a partir de la fuente.

Nuevamente, al tender al estado de equilibrio, el brazo del diapason no solo vuelve a su posición original, sino que la pasa. Esta acción mueve aire al volumen que antes ocupaba el brazo, produciendo un área de compresión deficiente o aire enrarecido. Se genera una reacción en cadena similar, moviendo las moléculas que antes se habían empujado.

Si sujetáramos una pluma al diapasón y desplazáramos una hoja de papel bajo ella a velocidad constante, el resultado sería una onda simétrica y repetitiva.

Desde luego, nuestros oídos son los principales receptores del sonido. El tímpano forma una membrana conectada a un volumen de líquido, y es sensible a cambios de presión dentro de ciertas longitudes de ondas. Los cambios de presión producen una acción hidráulica que se convierten en las señales eléctricas que el cerebro interpreta en última instancia como sonido.

### 2.2.2 Frecuencia

La forma más común de describir una onda es especificar su *frecuencia*, más que la longitud de onda de ciclos individuales. La frecuencia es una medida del número de ondas que pasan por un punto fijo en un período especificado. La forma común de expresar la frecuencia es en *ciclos por segundo* (Hertz). La frecuencia determina los tonos que percibimos en el sonido.

### 2.2.3 Amplitud

La altura y profundidad de las cimas y abismos de la onda se denomina intensidad de la onda o *amplitud*. Diferentes partes de la onda pueden tener amplitud positiva o negativa.

### 2.2.4 Sumas de Formas de Onda

La representación de la señal en el tiempo se denomina *forma de onda*. Las olas creadas al dejar caer una piedra en un estanque representa la más sencilla de todas las formas de onda: la *onda sinusoidal*. La curva suave y simétrica de la onda sinusoidal corresponde al movimiento básico de la mayor parte de las vibraciones naturales.

Las ondas complejas resultan de la superposición de dos o más vibraciones. Jean Batisté Fourier estudió la interacción de ondas sinusoidales y postuló que todas las formas de onda, por más complejas que sean, se pueden describir en función de sus ondas sinusoidales componentes. En las ondas periódicas, las ondas sinusoidales componentes son múltiplos enteros, o *armónicos*, de la frecuencia fundamental.

### 2.2.5 Fase

El término *fase* se refiere a la porción del ciclo de una onda periódica que ha transcurrido a partir de un punto predeterminado. Las relaciones de fases se describen en grado, siendo  $360^\circ$  el ciclo completo de una forma de onda.

Dado que las ondas son aditivas, la forma de onda que resulte de las ondas combinadas dependerá de su relación de fase, además de su amplitud y frecuencia.

### 2.2.6 Tipo de Señales

Las señales pueden dividirse en 4 categorías [AHM83]:

1. *Analógicas*
2. *Continuas en el tiempo*
3. *Discretas en el tiempo*
4. *Digitales*

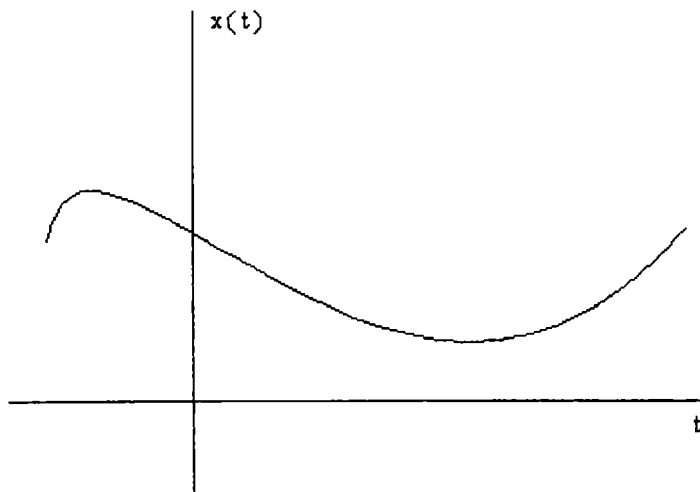
Estas señales están definidas con respecto a una variable independiente: *el tiempo*.

1. Una señal *Analógica* (Fig. 2.1) es aquella definida sobre un continuo de valores de tiempo, y cuya amplitud puede ser un rango continuo de valores. El término "analógico" fue quizás introducido en conexión con cálculos analógicos, donde voltaje y corriente son usados para representar cantidades físicas (velocidad, desplazamiento).

2. Las señales que están definidas como un continuo en el tiempo y cuyas amplitudes varían tanto en un rango continuo de valores o en una cantidad finita de valores posibles, se denominan señales *Continuas en el tiempo* (Fig. 2.2).

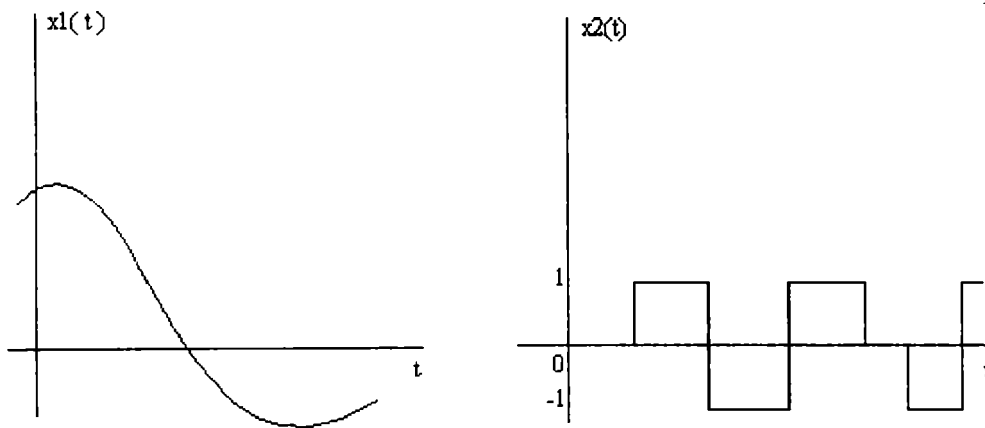
Así las señales analógicas pueden verse como un caso especial de señales continuas en el tiempo. En la práctica, sin embargo, la tendencia es usar los términos "analógico" y "continuo en el tiempo" indistintamente.

**Señal Analógica**



**Fig. 2.1**

**Señales Continuas en el Tiempo**



**Fig. 2.2**

3. Las señales *Discretas en el tiempo* (Fig. 2.3) están definidas sólo para un conjunto particular de valores de tiempo, lo cual significa que tales señales pueden representarse como secuencias de números. Una secuencia de números  $x$  en la cual el  $n$ -ésimo elemento se denota por  $x(n)$ , se expresa como

$$x = \{ x(n) \}, \quad |n| < \infty$$

Aunque lo anterior implica que  $x(n)$  es en realidad el  $n$ -ésimo elemento de la secuencia  $x$ , es conveniente denotar la secuencia misma por  $x(n)$ . Vemos que  $x(n)$  está definida solamente para valores enteros, o sea, no está definida para valores no enteros de  $n$ .

### Señal Discreta en el Tiempo

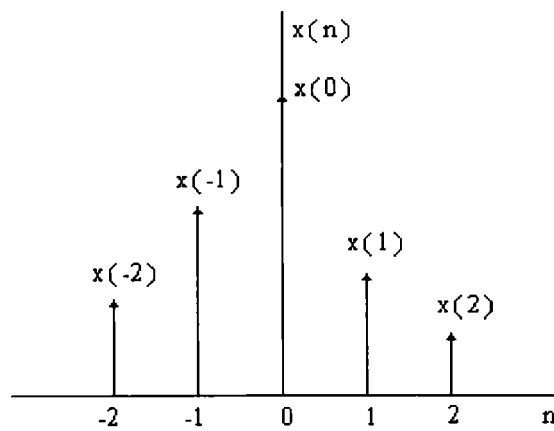


Fig. 2.3

4. En las señales *Digitales* (Fig. 2.4) tanto el tiempo como la amplitud son discretos.

### Señal Digital

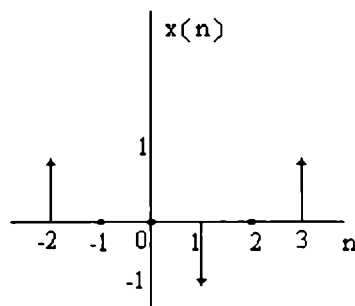


Fig. 2.4

A partir de esta clasificación de señales, se identifican diferentes tipos de *Sistemas*. Por lo tanto, *Sistemas Analógicos* son aquellos para los cuales tanto la entrada como la salida son señales analógicas. Aquellos cuya entrada y salida son señales discretas en el tiempo se denominan *Sistemas Discretos en el Tiempo*. Por último los que tienen como entrada y salida señales digitales, son llamados *Sistemas Digitales*.

## 2.3 Sonido

El sonido puede describirse como oscilaciones de presión del aire que estimulan el tímpano y por extensión, los nervios auditivos y el cerebro. Para crear esa estimulación, las oscilaciones deben estar dentro de cierto intervalo de frecuencias y amplitudes.

### 2.3.1 Frecuencia y Tono

La frecuencia y el tono están relacionados, pero difieren en varios aspectos importantes. En primer lugar, la frecuencia es una medición científica que corresponde a una de las características físicas de una forma de onda, en tanto que el tono es la cualidad subjetiva que nuestros cerebros perciben con base en la frecuencia.

### 2.3.2 Amplitud e Intensidad del Sonido

Derivamos la percepción relativa de la intensidad del sonido a partir de su amplitud absoluta, de la misma manera que el tono se deriva de la frecuencia absoluta de las ondas.

No es lo mismo volumen que intensidad del sonido. El volumen se refiere al nivel de potencia sonora en el cual el oído percibe diversos grados de intensidad dependiendo del intervalo de frecuencias.



### 2.3.3 Rúbricas Sonoras

La onda sinusoidal representa el movimiento más simple posible que podría resultar en un sonido. Si nuestro mundo auditivo fuese exclusivamente ondas sinusoidales simples, no podríamos distinguir entre fuentes de sonido distintas. Cada instrumento, persona u objeto en el mundo posee una rúbrica sónica o forma de onda propia cuando emite un sonido. La percepción de la característica de una forma de onda de audio, se denomina *timbre*.

### 2.3.4 Armónicos

La teoría de Fourier de que cualquier forma de onda es una mezcla de diversas ondas sinusoidales también se aplica en audio. La onda sinusoidal cuya frecuencia corresponde a la nota real que se toca se llama *frecuencia fundamental*. Casi todos los sonidos naturales tienen además una serie de armónicos u ondas sinusoidales de más alta frecuencia que se fusionan con la fundamental, para crear una forma de onda más compleja y un timbre más rico. En los sonidos afinados, todos estos armónicos son múltiplos enteros de la fundamental. La frecuencia del segundo armónico es el doble, la del tercero el triple, y así sucesivamente. Cada sonido tiene su propia combinación compleja de resonancias armónicas, y cada armónico tiene su propia amplitud.

El contenido armónico afecta a la respuesta de frecuencia requerida para reproducir fielmente los sonidos.

### 2.3.5 Decibel

Los niveles perceptibles del sonido abarcan intervalos tan amplios que se utiliza una escala logarítmica para medirlos. Un *decibel* es la décima parte de un *bel*, el cual equivale a una potencia con razón de 10:1. La unidad de medida de potencia (intensidad) sonora relativa es el *bel*.

### 2.3.6 Señales Digitales

El sonido analógico se transforma en representaciones numéricas mediante convertidores analógico - digital. No todo el sonido digital se genera igual. El número de bits de definición determina el intervalo dinámico, pues cada bit contribuye 6 decibeles en dicho intervalo. Por lo tanto, el sonido de 8 bits produce un intervalo

dinámico de 48 dB. El sonido de 16 bits produce un intervalo dinámico de 96 dB que tienen los discos compactos, similar al oído humano.

El proceso de cuantización empleado para digitalizar el audio, produce a veces efectos extraños conocidos como *ruido de cuantización*.

La tasa de muestreo determina la respuesta de frecuencia de la grabación de acuerdo con el teorema de Nyquist. El aliasing que aparece cuando se intenta grabar frecuencias que exceden la mitad de la tasa de muestreo, se manifiesta como ruido anómalo en la grabación. Las pendientes de corte desde los filtros pasa bajo empleados para eliminar frecuencias altas que pudieran provocar aliasing, casi siempre hacen que la respuesta de frecuencia efectiva sea menos que la mitad de la tasa de muestreo; debido a ello la tasa de muestreo de 44.1 KHz empleada con los CD de audio produce una respuesta de frecuencia de 0 a 20 KHz.

Un aumento en la tasa de muestreo o en la definición implica una mayor demanda de rendimiento y almacenamiento. Por ejemplo, los datos de 16 bits contienen el doble de información que los de 8 bits; una tasa de muestreo de 44.1 KHz requiere el doble que una de 22.05 KHz; el sonido estereofónico requiere el doble que el monoaural. Esto hace que las necesidades de rendimiento y almacenamiento, vayan desde aproximadamente 1.25 MB por minuto en el caso de audio monoaural de 8 bits a 22.01 KHz, hasta 10 MB por minuto para el audio estereofónico de 16 Bits a 44.1 KHz.

Cuando se desea trabajar sobre una señal analógica, es necesario llevarla a un formato digital, para lo cual debe tener ancho de banda limitado o debe filtrarse para reducir el mismo. Es posible, realizar la discretización en relación al tiempo sin pérdida de información. Sin embargo no es posible realizar la digitalización completa sin pérdida, por lo cual es objeto de estudio llevar a cabo esta transformación manteniendo un error mínimo.

### 2.3.7 Relación Señal-Ruido (SNR)

La SNR es el método de medición más usado para los sistemas de codificación de forma de onda y analógicos, y ha contribuido en la evaluación de distorsiones de ruido en diversos algoritmos. Existen diferentes variaciones incluyendo la SNR clásica, SNR segmental, entre las más usadas. Es importante notar que las mediciones basadas en SNR son sólo apropiadas para los sistemas de codificación que buscan reproducir una forma de onda de entrada original. Sea  $y(n)$  una señal de audio que contiene ruido en el instante  $n$ ,  $s(n)$  es la señal sin ruido equivalente, y  $s^*(n)$  es la señal procesada. El error del procesamiento de la señal es

$$\varepsilon(n) = s(n) - s^*(n)$$

La energía del error es:

$$E_\varepsilon = \sum_{n=-\infty}^{\infty} \varepsilon^2(n) = \sum_{n=-\infty}^{\infty} [s(n) - s^*(n)]^2$$

La energía contenida en la señal de audio:

$$E_s = \sum_{n=-\infty}^{\infty} s^2(n)$$

La medida resultante SNR ( expresada en decibeles) se obtiene como:

$$SNR = 10 \log_{10} \frac{E_s}{E_\varepsilon} = 10 \log_{10} \frac{\sum_n s^2(n)}{\sum_n [s(n) - s^*(n)]^2}$$

La medida representa un error promedio sobre el tiempo y la frecuencia para una señal procesada. La SNR clásica es por lo tanto una primera estimación de la distorsión de una señal de audio codificada. La SNR pesa todos los errores de la forma de onda en el dominio del tiempo de igual modo. La energía de una señal de audio

generalmente varía en el tiempo. Si asumimos que la distorsión producida por el ruido es amplia con una pequeña fluctuación de energía, entonces la medida de SNR debería variar en cada bloque o frame de la señal. Una medida de mayor calidad puede obtenerse si la SNR se mide en bloques o frames y los resultados se promedian. Esta medición basada en frames se denomina SNR segmental ( $SNR_{seg}$ ):

$$SNR_{seg} = \frac{1}{M} \sum_{j=0}^{M-1} 10 \log_{10} \left[ \sum_{n=m_j}^{m_{j+1}} \frac{s^2(n)}{[s(n) - s^*(n)]^2} \right]$$

donde  $m_0, m_1, \dots, m_{M-1}$  son los tiempos en que finaliza cada uno de los  $M$  frames, los cuales tienen longitud  $N$ . La segmentación de la SNR permite obtener una medida objetiva de la calidad de una señal de audio codificada.

En algunos casos pueden surgir problemas con  $SNR_{seg}$  si los frames de silencio son incluidos en su cálculo, dado que pueden aparecer valores negativos muy grandes. Este problema se resuelve identificando los periodos de silencio y excluyéndolos del cálculo. Otro enfoque es fijar un umbral inferior, y reemplazar todos los frames con medidas de  $SNR_{seg}$  inferiores por dicho umbral (un umbral de 0 dB es razonable). Por otro lado los frames con  $SNR_{seg}$  mayor que 35 dB no resultan ser demasiado diferentes para los oyentes. Por lo tanto un umbral superior (normalmente de 35 dB) se usa para evitar medidas altas de  $SNR_{seg}$  inusuales. De este modo los dos umbrales establecidos acotan la  $SNR_{seg}$  para que resulte objetiva y razonable.

### 2.3.8 Codificadores de Señales de Audio

El proceso de conversión de una señal analógica a formato digital se conoce como *codificación de audio*. Un objetivo importante en dicha codificación es la compresión de la señal, esto es, emplear la menor cantidad de bits posible para lograr eficiencia del ancho de banda en la transmisión, para una variedad de canales de comunicación; o bien para el almacenamiento en algún medio magnético u óptico.

Varias técnicas de codificación han sido propuestas, analizadas y desarrolladas durante las últimas cinco décadas. Las técnicas que trataremos posteriormente pueden clasificarse en dos grandes categorías: *codificadores de forma de onda* y *codificadores de voz o vocoders (voice coders)*.

El objetivo de los codificadores de forma de onda es obtener una representación adecuada de la forma de la señal de audio, aprovechando sus características temporales y/o espectrales (análisis frecuencial).

Para el caso particular de los codificadores de voz humana o vocoders, el propósito es lograr una representación de la señal del habla mediante la identificación de un conjunto de parámetros, la estimación de los mismos a partir de porciones de dicha señal, y la codificación eficiente de tales parámetros en formato digital a fin de realizar una posible transmisión o almacenamiento.

En la siguiente tabla se presenta una clasificación de las técnicas de codificación para señales de audio (algunas de las cuales serán tratadas en los próximos capítulos) :

<b>Codificadores de Señales de Audio</b>	Codificadores de Forma de Onda	Dominio del Tiempo	PCM, LogPCM DPCM ADPCM DM
		Dominio Espectral	SBC BTC
	Codificadores de Voz ( Vocoders )	LPC, CELP RELPC Multipulse LPC	

# Capítulo 3 : CODIFICADORES DE FORMA DE ONDA EN EL DOMINIO TEMPORAL

## 3.1 Introducción

Veremos la transición desde una señal de audio analógica, la cual es continua con respecto a los valores que puede tomar y con respecto al tiempo, a una señal digital la cual es discreta con respecto a dichos parámetros.

Los pasos a seguir pueden esquematizarse de la siguiente forma:

**Señal Analógica → Prefiltrado → Muestreo → Codificación → Señal Digital**

El prefiltrado se realiza para asegurar que la señal tendrá un ancho de banda limitado; esto consiste en dejar pasar algunas frecuencias, sólo las menores a un valor de frecuencia dado.

El muestreo consiste simplemente en tomar muestras de la señal original a intervalos de tiempo regulares.

Para reconstruir una señal analógica la teoría de muestreo básica establece que la tasa de muestreo se basa en el ancho de banda de la señal.

El segundo atributo de importancia de una señal muestreada es la resolución, o cantidad de bits, que se usa para digitalizar la forma de onda de entrada. La mayoría de los sistemas de sonido para PC ofrece resolución de 8 y 16 bits, mientras los sistemas de música o de calidad CD ( Hi-Fi ) requieren 16 bits de resolución. Estos dos atributos ( tasa de muestreo y resolución ) determinan la tasa de datos completa y la cantidad total de memoria requerida para almacenar una señal digitalizada.

La codificación se realiza mediante la asignación de un número (representado con una cantidad de bits) a cada uno de los valores muestreados.

Toda señal que se desee muestrear puede ser pensada como una función del tiempo  $f(t)$ , es posible entonces modelar el comportamiento de la señal y analizarla en forma matemática [NIE89].

Describiremos diferentes técnicas de codificación aplicadas a señales de audio. En particular consideraremos *PCM (Pulse Code Modulation)*, *DPCM (Diferencial Pulse Code Modulation)*, *DM (Delta Modulation)*, y algunas versiones adaptativas de estos métodos.

### 3.2 Pulse Code Modulation (PCM)

Un método básico para obtener una versión digital de una señal es *PCM*, como se muestra en la (Fig. 3.1) para una función  $f(t)$ . El primer paso es la cuantización de la variable independiente  $t$  a un número de *valores de muestreo* discreto. Se establece que :

$$f(t) \rightarrow f_i = f(t_0 + i \Delta t) = f(i), i = 0, \dots, M - 1, \text{ donde } M \text{ es el número de muestras ,}$$

#### Digitalización de una señal en PCM

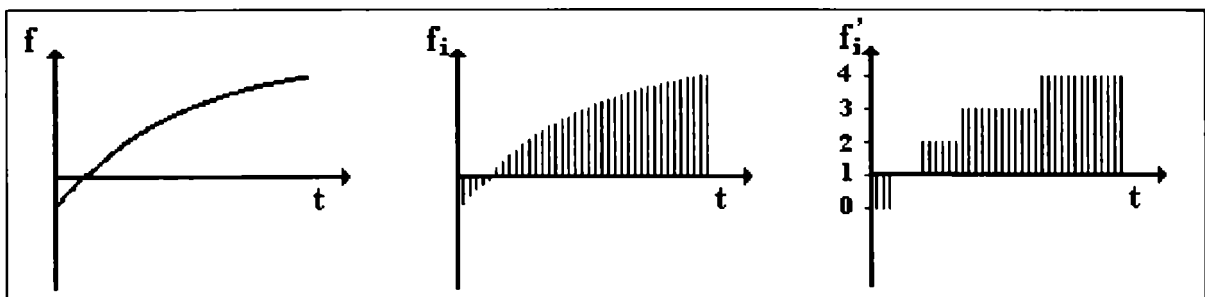


Fig. 3.1

Como las amplitudes tomadas en puntos de muestreo discretos toman valores continuos  $f_i$ , el segundo paso es cuantizarlos a valores de amplitud discretos  $f'_i$  representados por  $B$  bits, para procesamiento digital. Esto puede ser visto como que un rango continuo  $f_{\min} \leq f_i \leq f_{\max}$  es mapeado a enteros entre  $0, 1, \dots, 2^B - 1$  los cuales

pueden ser almacenados utilizando B bits.; siendo  $L=2^B$  bits ( la cantidad de intervalos).

Para aplicar *PCM* deben realizarse tres elecciones:

- 1- La frecuencia de Muestreo ( Sample Rate ) o el número M de puntos de muestreo.
- 2 - El número B de bits para representar los puntos de muestreo.
- 3 - La Característica de Cuantización la cual mapea un intervalo de amplitudes continuas a amplitudes discretas.

### 3.2.1 Frecuencia de Muestreo.

El Sample Rate es determinado por el Teorema de Muestreo (Nyquist), el cual sostiene que :

Si  $f(t)$  es una función con Transformada de Fourier

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \exp(-i2\pi\omega t) dt, \quad \text{tal que } F(\omega) = 0 \text{ si } |\omega| \geq 2f_g$$

entonces  $f(t)$  es determinado completamente por los valores muestreados  $f_i$ , con

$$f_i = f(t_0 + i \Delta t), \quad i=0, \pm 1, \pm 2, \dots$$

si el periodo de muestreo es  $\Delta t \leq 1/(2 f_g)$ .

La función continua  $f(t)$  puede ser reconstruida por la fórmula de interpolación :

$$f(t) = \sum_{i=-\infty}^{+\infty} f_i \text{sen}(2\pi f_g (t - i\Delta t)) / (2\pi f_g (t - i\Delta t))$$

Este teorema establece que una función con ancho de banda limitado, puede ser representada por sus valores discretos de muestreo.

Para señales de audio las diferentes frecuencias de muestreo son las la siguientes [REI94]:



Calidad	Frecuencia de muestreo (Hz.)
Toll	8000
AM	11025
FM	22050
Hi - Fi	44100

### 3.2.2 Cantidad de Bits por Muestra.

El bit rate se define como la cantidad de bits por unidad de tiempo. Si se muestrea con periodo de muestreo  $\Delta t$  fijo, entonces el bit rate estará dado por  $B/\Delta t$  (bits/unidad de tiempo), o un total de  $M * B$  bits. Debido a que no es posible aumentar el  $\Delta t$ , usualmente se intenta reducir  $B$ .

$B$  determinará la precisión con la que  $f_i'$  aproximará a  $f_i$ .

Se introducirá el concepto de ruido de cuantización con el fin de estudiar la precisión de una representación de *PCM*. Dicho ruido se define como

$$n_i = f_i - f_i'$$

Una medida de la precisión de una representación *PCM* es:

$$r' = E\{f_i^2\} / E\{n_i^2\}, \text{ donde } E\{x\} \text{ denota la esperanza de } x,$$

$r'$  representa la relación señal-ruido de cuantización.

Daremos una relación entre  $B$  y la variable  $r = 10 \log_{10}(r')$ .

Si  $E\{n\} = E\{f\} = 0$ , la cuantización es razonable (ej.  $B > 6$ ), entonces la relación:

$$r = 6B - 7.2 \text{ se cumple.}$$

Por ejemplo si se trabaja con  $B = 11$  bits se obtiene una relación señal-ruido de 60 dB aproximadamente.

### 3.2.3 Características de Cuantización.

La característica de cuantización es la forma de mapear un rango de valores continuos, el cual estará dado por el dominio de la señal a muestrear, a un rango de valores discreto.

La idea es realizar una partición del dominio ( rango continuo ) en  $L = 2^B$  intervalos, cada uno de los cuales será representado por un valor discreto.

A fin de medir la calidad de la característica de cuantización, definiremos un criterio, el cual estará dado por el error cuadrático medio:

$$\varepsilon = \sum_{v=1}^L \int_{a_v}^{a_{v+1}} (f - b_v)^2 p(f) df$$

donde  $L$  es la cantidad de intervalos,

$( a_v, a_{v+1} )$  es uno de los  $L$  intervalos,

$f$  es la señal,

$b_v$  es el valor discreto resultante de la cuantización para cada subrango,

$p( f )$  es la probabilidad de que la señal tome algún valor del subrango del intervalo

No es necesario que la cuantización sea realizada uniformemente, es decir que no todos los subrangos del intervalo serán necesariamente del mismo tamaño.

### 3.3 Logarithmic Pulse Code Modulation (LogPCM)

Las señales de speech tienen la característica que las pequeñas amplitudes ocurren con más frecuencia que las grandes. Más aún, el oído humano exhibe una sensibilidad logarítmica. Sin embargo un cuantizador uniforme proporciona el mismo salto entre niveles sucesivos dentro del rango dinámico completo de la señal. Una mejor aproximación es usar un cuantizador no uniforme, que provea niveles más cercanos entre sí para las amplitudes más bajas, y niveles más distantes para las amplitudes más altas. Para un cuantizador no uniforme, el error de cuantización resultante es menor *PCM*. Un cuantizador no uniforme se obtiene usualmente mediante el pasaje de la señal a través de un filtro, que comprime la amplitud de la misma, seguido por un cuantizador uniforme.

*LogPCM* es una técnica basada en una forma de compresión que es logarítmica en vez de lineal. Generalmente se aplica alguna de las siguientes funciones logarítmicas para la representación:

(a) Ley  $\mu$  ( $\mu$ -Law) (Sistemas de telecomunicaciones de EE. UU. y Canadá)

$$|y| = \frac{\log(1 + \mu|s|)}{\log(1 + \mu)}$$

(b) Ley A (A-Law) (Sistemas de telecomunicaciones de Europa)

$$|y| = \frac{1 + \log A|s|}{1 + \log A}$$

Donde  $|s|$  es la magnitud de la entrada,  $|y|$  la magnitud de salida y,  $\mu$  y  $A$  son parámetros que se seleccionan para dar la característica de compresión deseada.

La naturaleza del algoritmo provee una razón de compresión cercana a 2:1, permitiendo que se disponga de 14 bits de rango dinámico contenido en una muestra que requiera 8 bits para su almacenamiento físico. Existen varios dispositivos actualmente disponibles que implementan esta técnica, en particular aquellos relacionados con el almacenamiento señales de voz para los cuales LogPCM es adecuada [REI94].

### 3.4 Differential Pulse Code Modulation (DPCM)

En *PCM*, cada muestra de una forma de onda se codifica en forma independiente de las restantes. Sin embargo, las mayoría de las señales fuente muestreadas a frecuencias altas, exhiben una correlación significativa entre muestras sucesivas. En otras palabras, la variación de amplitud promedio entre muestras sucesivas es relativamente pequeña. Consecuentemente, un esquema de codificación que explote la redundancia en las muestras, resultará en una tasa de bits más bajas para la salida.

Una solución relativamente simple es codificar las diferencias entre muestras sucesivas en lugar de las muestras mismas. La técnica resultante se denomina *DPCM*

(*Diferencial Pulse Code Modulation*). Dado que se espera que las diferencias entre las muestras sean menores que los valores de amplitud reales, es también de esperar que se requieran menos bits para representarlas. En este caso cuantizamos y transmitimos la secuencia de diferencias:

$e(n) = s(n) - s(n-1)$ , donde  $s(n)$  y  $s(n-1)$  son los valores de entrada y  $e(n)$  la diferencia.

### 3.5 Adaptive Differential Pulse Code Modulation (ADPCM)

Otra variación de *PCM*, llamada "*Adaptive Differential PCM*" (*ADPCM*), proporciona una tasa de compresión 4:1. La misma usa la historia de la señal de audio de entrada como un método para predecir los cambios en la misma. En lugar de representar el valor de entrada de la señal, *ADPCM* codifica la diferencia entre el valor de la muestra de entrada y el valor de predicción para la misma. Si la predicción (la cual se realiza típicamente con un proceso de filtrado mediante un loop de retroalimentación) es buena y la diferencia pequeña, dicha diferencia puede codificarse fácilmente con menos bits que los requeridos para representar directamente la muestra de entrada. La calidad de una entrada de audio comprimida con esta técnica está en gran parte determinada por el tasa de muestreo y la extensión de la historia de la señal de entrada.

### 3.6 Delta Modulation (DM)

Una implementación simple de Differential Coding es *Delta Modulation*, en la cual la palabra de código se reduce a un bit.

*DM* es entonces, otra forma de predicción lineal en la cual se utiliza una palabra de código de un bit. *DM* también recibe el nombre de Modulación Lineal Delta (Linear Delta Modulation, *LDM*) cuando el tamaño del paso es fijo.

Un único bit que representa el signo indica el sentido de la diferencia entre la forma de onda de entrada y la salida acumulada, en el momento del muestreo. El bit se

utiliza para determinar si debe incrementarse o decrementarse en un valor (el cual es constante) la forma de onda de salida. Cuando la pendiente de la forma de onda de entrada excede la pendiente máxima que el codificador puede alcanzar, se produce un efecto denominado sobrecarga de pendiente (Slope overload, Fig. 3.2 (a)). Tal efecto puede atenuarse aumentando la frecuencia de muestreo, agrandando el tamaño del paso, o ambos. Otra distorsión que puede surgir es el llamado ruido granular (Fig. 3.2 (b)), producido por la dimensión del valor que incrementa o decrementa, y se atenúa reduciendo dicha dimensión. Para el caso de señales de speech el sobremuestreo (oversampling) ayuda a controlar el problema de sobrecarga.

Con un balance adecuado de tasa de muestreo y tamaño de paso, *DM* puede producir para un nivel de relación señal ruido dada, una tasa de datos menor que la obtenida mediante cuantización logarítmica, sin embargo, esto se da para speech de baja calidad, con tasas menores a 16 Kbps.

Por lo tanto, a pesar de que es posible lograr alguna ganancia adaptando el tamaño del paso, *DM* se usa solamente en aquellos casos en los cuales es aceptable una tasa de datos inferior a la correspondiente a la calidad telefónica, y es deseable disponer de un codificador cuyo funcionamiento sea muy simple.

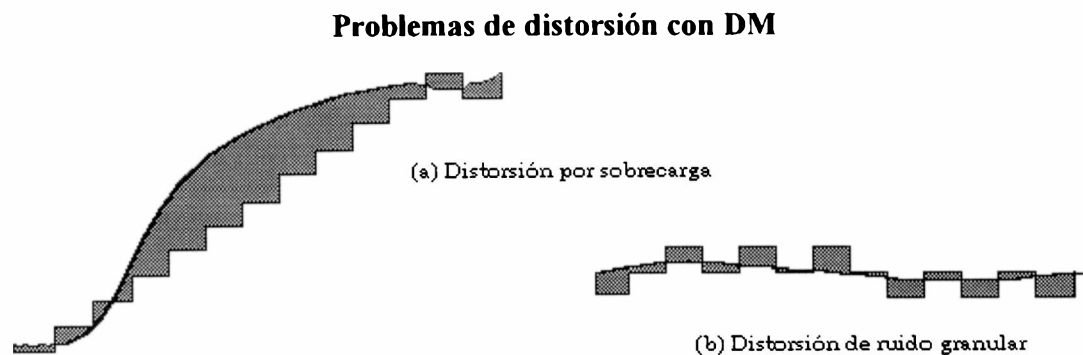


Fig. 3.2

### 3.7 Linear Predictive Coding (LPC)

El método de codificación lineal predictivo (*Linear Predictive Coding, LPC*) brinda aún mayores niveles de compresión, *pero es aplicable a voz en lugar de audio*

en general. *LPC* fue diseñado para procesar el speech humano, y usa un modelo de producción que aprovecha las características de la producción del habla, esto es, el tracto vocal humano, y la hipótesis de una única fuente de sonido. Usando esta técnica es posible lograr una tasa de compresión de 32:1 y mayores aún. Una de las variantes más recientes de *LPC* denominada *Codebook-Excited Linear Prediction (CELP)* ofrece una calidad superior.

## 3.8 Métodos de Compresión Analizados en el Dominio del Tiempo

Dado que nuestro objetivo no es analizar exhaustivamente algoritmos en el plano temporal, describiremos brevemente algunos métodos analizados para archivos de audio con forma de onda de muestras de 8 bits:

- truncamiento y conexión de dos muestras
- análisis de diferencias
- análisis de diferencias con exponente fijo
- análisis de diferencias con exponente variable

### 3.8.1 Truncamiento y Conexión de Dos Muestras

Truncar consiste en eliminar de cada muestra los 4 bits menos significativos. Una vez aplicado el truncamiento, quedan los 4 bits más significativos con información y los menos significativos quedan en 0, de este modo podemos agrupar dos muestras en un solo byte. Obviamente la tasa de compresión es 50%. El decompresor toma un byte (compuesto por dos muestras truncadas) y recupera las 2 muestras iniciales, pero con la pérdida de los 4 bits menos significativos.

La principal característica es la simpleza más que los resultados logrados, ya que se pierde la mitad de la información.

### 3.8.2 Análisis de Diferencias

Generalmente existe una correlación entre dos muestras consecutivas de un archivo. Dicha correlación nos permite deducir que las diferencias entre ellas no son grandes.

Codificamos sólo las diferencias y las almacenamos en 4 bits (hasta aquí no hay pérdida), pero si la diferencia es mayor que 15 almacenamos este último valor.

La reconstrucción consiste en partir de la muestra base inicial e ir sumando las diferencias para obtener los valores de entrada originales. Si en algún momento la muestra reconstruida no es igual a la original, se debe a que la diferencia fue mayor que 15, y a partir de allí la pérdida se propaga al resto del archivo reconstruido. Por lo tanto los resultados obtenidos no fueron buenos.

### 3.8.3 Análisis de Diferencias con Exponente Fijo

En este método se toman pares de muestras, a las que se le aplica un exponente fijo, el cual está predeterminado.

Es de suponer que en algunos casos, las diferencias entre muestras consecutivas debe ser mínima, por lo tanto podemos inferir que pueden almacenarse en los 4 bits menos significativos. Ahora bien, qué sucede si dentro de los 4 bits más significativos queda información, perderla representa recuperar más tarde, un archivo bastante distante del original. Por lo tanto, el exponente fijo produce un corrimiento hacia la derecha de los bits de la diferencia, en tantas posiciones como especifique su valor, reduciendo notablemente la probabilidad de pérdida de información.

El decompresor, deberá tomar un byte, descomponerlo en dos diferencias y luego rearmar las muestras originales, produciendo un corrimiento hacia izquierda (tal como lo indique el exponente).

El resultado obtenido es mejor que el anterior, pero la calidad obtenida está lejos de ser satisfactoria.

### 3.8.4 Análisis de Diferencias con Exponente Variable

El algoritmo compresor toma un conjunto de muestras formadas por 7 bytes, los cuales se compactarán de acuerdo al siguiente análisis [ROW94]:

- 1-. Se calculan las diferencias entre bytes consecutivos del conjunto.

2-. Se elige un exponente relativo a la mayor diferencia. Si es menor que 4, se asume este valor.

3-. Se aplica el exponente a cada diferencia, esto consiste en efectuar tantos corrimientos a derecha como él indique. Esta operación asegura que los 4 bits más significativos de cada diferencia quedarán en 0.

4. Se agrupan dos diferencias en un byte. De esta forma las 7 diferencias (56 bits) se transforman en 28 bits (3 bytes y medio).

5. Se agrega el exponente al principio para completar los 4 bytes.

El algoritmo descompresor realiza los pasos en orden inverso.

Los resultados obtenidos aquí fueron más aceptables, pero la compactación alcanzada, de casi un 50%, es poco significativa.



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



# Capítulo 4 : CODIFICACION DE FORMA DE ONDA EN EL DOMINIO FRECUENCIAL

## 4.1 Introducción

Los algoritmos de codificación en el dominio del tiempo, tratan la señal en forma completa [JAY94] eliminando la redundancia antes de la codificación mediante la aplicación de algún método predictivo. Las principales diferencias entre los algoritmos utilizados están dadas por el grado de predicción empleado, y si esos esquemas son adaptativos o no.

Por otro lado, los algoritmos de codificación en el dominio frecuencial dividen la señal de entrada en componentes de frecuencias separadas, y codifican cada uno de ellos separadamente.

## 4.2 Block Transform Coding (BTC).

### 4.2.1 Generalidades.

Para una representación eficiente de señales se utiliza Block Transform Coding (BTC). La idea central subyacente de esta técnica en el tratamiento de señales no independientes, no es codificar cada muestra individualmente, sino grupos de muestras los cuales se denominan bloques o frames.

BTC es una técnica general para codificación de señales digitalizadas en una o más dimensiones. Para audio, la señal es muestreada en el tiempo, subdividida en bloques de  $N$  muestras y transformada al dominio espectral para codificación y transmisión o almacenamiento (Fig. 4.1). En el decodificador, cada bloque de muestras espectrales se

transforma nuevamente al dominio temporal, y la señal de audio es sintetizada concatenando los bloques reconstruidos (Fig. 4.2). [BRI95]

BTC es una técnica de codificación en el dominio frecuencial, sin embargo, el problema de asignación de bits para cada línea espectral no es simple.

Varias técnicas pueden ser usadas para la codificación de señales de audio en el dominio espectral con diferente performance y complejidad de implementación. Entre ellas, *Zonal Transform Coding (ZTC)*, *Threshold Transform Coding (TTC)* y *Adaptive Transform Coding (ATC)*[BRI95].

Diferentes estrategias de asignación de bits han sido sugeridas en la literatura del tema. Las tendencias del momento, tratan de usar modelos psicoacústicos para controlar el cuantizador y el codificador.

Utilizando valores apropiados para el tamaño de los bloques y estrategias adecuadas de asignación de bits, BTC proporciona digitalizaciones de alta calidad a una tasa promedio de 2 bits/muestra, y digitalizaciones aceptables de 1 bit-muestra.

También surgen problemas de distorsión de bloque empleando la técnica de BTC como consecuencia de la cuantización. Estos son: *Clicking*, *Burbling* y *Pre-echoes* [TRI79][O'SH87]. La eficiencia de esta técnica dependerá del tipo de **transformada lineal** utilizada (Transformada Discreta de Fourier, Transformada Discreta Coseno, etc), y la **estrategia de asignación de bits** para los coeficientes transformados.

Por último se concluye que la principal ventaja de BTC es su buena resolución en el plano frecuencial.

### Pasaje del dominio temporal al dominio frecuencial

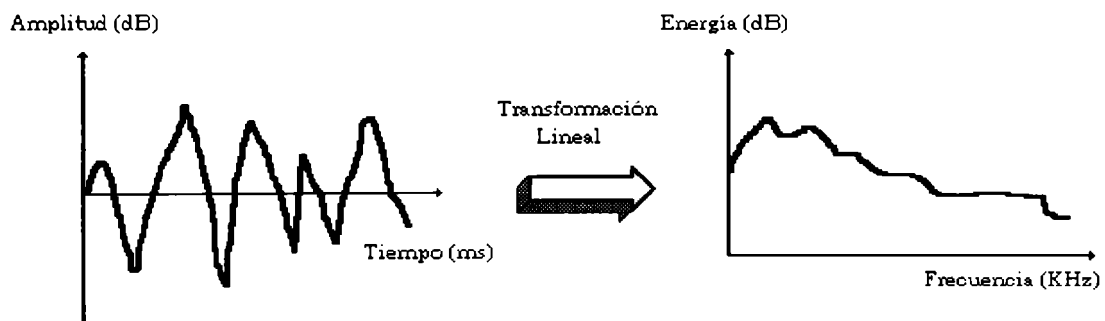


Fig. 4.1

### Esquema de Codificación

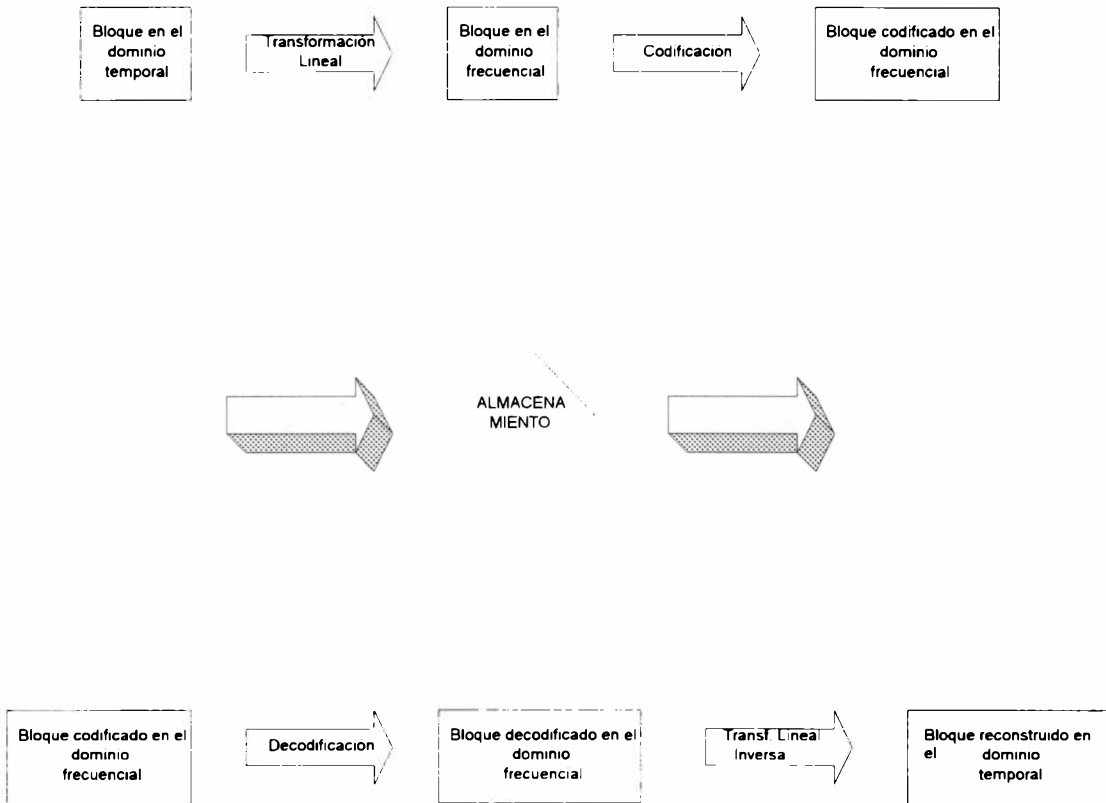


Fig. 4.2

#### 4.2.2 Transformaciones Lineales

Primeramente la señal de entrada se divide en bloques o "frames" de longitud fija  $N$ . Para cambiar de dominio se debe emplear una transformación lineal adecuada, ya que el objetivo es "decorrelacionar" las muestras.

Existen varias alternativas:

- **Transformada de Karhunen-Loéve (KLT)**: empaqueta la mayor parte de la energía de la señal dentro de una mínima cantidad de coeficientes para un nivel de error deseado, conduciendo teóricamente a la tasa mínima de bits.

Es óptima en el sentido que devuelve valores espectrales no correlacionados. Sin embargo, esta transformación es difícil de calcular ya que deben enviarse al receptor las funciones bases (las cuales dependen de los datos de entrada para cada cada frame),

como información adicional a fin de poder realizar la transformación inversa [DEL93] [O'SH87].

- **Transformada Discreta de Fourier (DFT)**: esta transformada se usa para calcular las funciones de correlación y producir el espectro de frecuencia de una forma de onda discreta de longitud finita. Tomando como entrada una forma de onda de N muestras, la DFT divide el espectro desde 0 Hz. hasta la frecuencia de muestreo en N intervalos similares, y provee un valor de energía para cada uno de ellos. Sin embargo, las salidas son simétricas con respecto al centro del rango de frecuencia, así solamente N/2 valores complejos son usados [PEL93].

DFT es una alternativa razonable, más simple para calcular y con funciones bases sinusoidales fijas, pero sólo aproxima la optimalidad de KLT en la ganancia de predicción para bloques de datos de gran tamaño [O'SH87].

- **Transformada Discreta Coseno (DCT)**: es la elección más común en BTC, ya que es la más cercana a KLT en optimalidad. Con esta transformada se obtiene una performance superior en la relación señal-ruido, y una reducción en los efectos de bordes en los bloques respecto a DFT. Además DCT produce un conjunto de N coeficientes reales en vez de los N/2 coeficientes complejos de DFT [COX81].

Para un bloque de N muestras, DCT se define como:

$$x = (x(0), x(2), \dots, x(N-1))$$

$$\theta(k) = \sqrt{\frac{2}{N}} \alpha(k) \sum_{i=0}^{N-1} x(i) \cos\left(\frac{(2i+1)k\pi}{2n}\right)$$

$$\alpha(0) = 1/\sqrt{2} \quad \text{con} \quad \alpha(k) = 1 \text{ para } k \neq 0$$

Varias técnicas han sido sugeridas para resolver los efectos de bloque, llamadas genéricamente "transformadas solapadas" como por ejemplo la Transformada Discreta Coseno Modificada (MDCT) [IWA92].

### 4.2.3 Técnicas de Codificación

***Zonal Transform Coding (ZTC)***: es una técnica de codificación simple para BTC en la cual las líneas de frecuencias más altas no son consideradas. Consecuentemente, la compresión obtenida con esta técnica es similar a la acción de un filtro pasabajos.

***Threshold Transform Coding (TTC)*** : es una técnica de codificación general para BTC, en la cual se descartan las líneas con menor energía que un umbral (Fig. 4.3). Un efecto general pasabajos aparece también con esta técnica. Según la implementación realizada, el umbral puede ser fijo o no.

***Adaptative Transform Coding (ATC)***: similar a TTC en cuanto a la selección de líneas a ser codificadas, pero con un algoritmo específico que mantiene fija la cantidad de información por bloque. Para lograr tal performance el algoritmo adapta dinámicamente, no sólo el umbral para cada bloque, sino también el número de bits asignados para codificar cada línea espectral dentro de cada frame. Por lo tanto, se asignan más bits a los coeficientes espectrales más importantes y menos a los de menor relevancia. Usando este tipo de alocaión dinámica en la asignación del número total de bits para los coeficientes espectrales, se obtiene una adaptación conveniente a los cambios en la señal de audio codificada. Según la tasa de bits utilizada, algunas frecuencias con baja energía pueden anularse ( asignación de 0 bits ), así sus valores no se transmiten, lo cual produce "huecos" imperceptibles en el espectro de la señal reconstruida. Además, distintas alternativas pueden derivarse de esta técnica de codificación básica.

## Umbral de Codificación

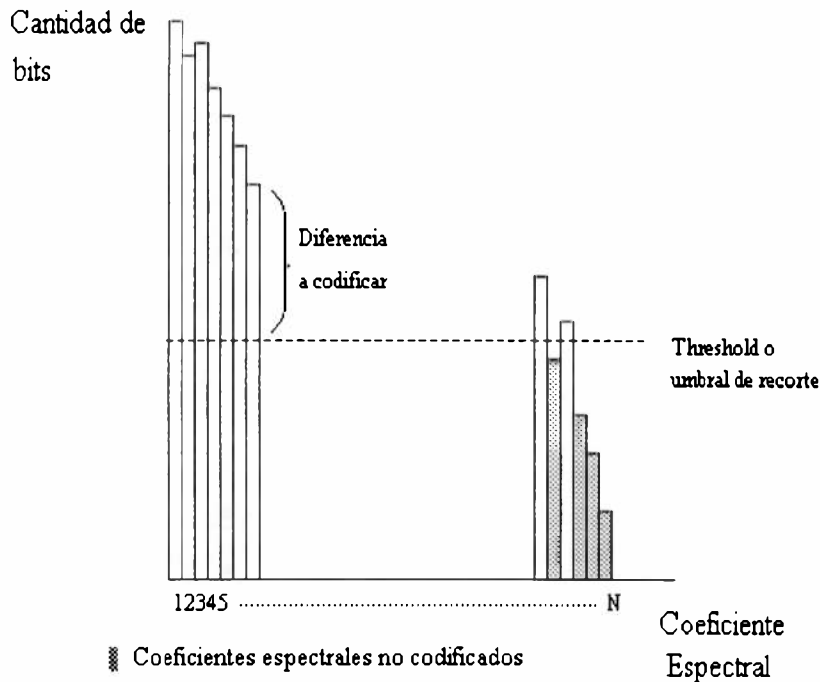


Fig. 4. 3

### 4.2.4 Alocación Óptima de Bits para BTC

Los algoritmos de codificación de señales de audio en el dominio frecuencial, dividen la señal de entrada en una cantidad de componentes de frecuencias distintas y codifican cada uno de ellos en forma separada (cuantizan cada línea espectral independientemente una de otra). Ello se puede realizar debido a la “decorrelación” existente entre los coeficientes en el espectro, lo cual no ocurre en el plano temporal.

Dicha “decorrelación” depende de la transformación lineal usada. La transformada KLT es la óptima; aunque la que se usa es DCT, la cual es apropiada para audio. Por lo tanto se asegura la “decorrelación” como hipótesis de trabajo.

En el dominio frecuencial, es posible en general, no considerar algunas muestras ( las de baja energía) , obteniendo así cierto nivel de **compresión** con una pequeña pérdida de información.

La teoría de BTC revela que todos los coeficientes transformados deben sufrir la misma distorsión, para una distorsión total mínima. Para codificar los coeficientes transformados de “alta” varianza se usa un “alto” número de bits y viceversa[ZEL79].

La calidad de señales codificadas usando BTC, se puede medir al igual que PCM, mediante la relación señal-ruido ( SNR).

La performance de BTC puede ser caracterizada por una ganancia de performance sobre PCM, denotada como  $G_{BTC}$ . [ JAY84]

$$SNR_{BTC}(dB) = SNR_{PCM}(dB) + 10 \log G_{BTC} \quad (\text{para una señal analógica})$$

La ganancia de BTC dependerá de :

- la estadística fuente de la señal ( lo que garantiza que la varianza de cada elemento del bloque de entrada sea la misma, y que los coeficientes sean independientes)
- el tipo de transformada lineal aplicada
- la longitud del bloque a transformar

Denotaremos:

vector  $x$  : bloque a transformar en el dominio del tiempo

$A$  : matriz ortogonal ( permite invertirse ), generalmente unitaria u ortonormal (dependiendo si se trabaja con elementos complejos o reales), de transformación.

$\theta$  : bloque transformado en el dominio espectral =  $Ax$

$Q_i$  : cuantizador del coeficiente- $i$  transformado (  $i: 0, \dots, N-1$  )

$Q$  : cuantizador del bloque transformado

vector  $u$  : bloque de coeficientes transformados cuantizados =  $Q(\theta)$

vector  $v$  : recuperación de  $u$  (según si  $u$  fue transmitido o almacenado)

vector  $y$  : vector de salida reconstruido en el dominio del tiempo

$A^{-1}$  : transformación lineal inversa

$$y = A^{-1}u = A'u = \sum_{k=0}^{N-1} u(k)B(k)$$

$B_k$  = vectores básicos de la transformación usada

Se considera como hipótesis que : cada vector  $x$  es una fuente escalar estacionaria de media 0.

La ortogonalidad es una propiedad claramente necesaria para los vectores básicos, los cuales son usados para descomponer la señal de entrada en componentes no correlacionados en un espacio  $N$ -dimensional.

$\theta$  : var. aleatoria del bloque transformado

La ortonormalidad de los vectores básicos es una propiedad más fuerte, la cual lleva a transformaciones que son necesarias en Transform Coding, para hacer *la suma promedio de las varianzas de los elemento de  $\theta$  igual a  $\sigma_x^2$  (varianza de los elementos de  $x$ ).*

Esto se puede expresar de la siguiente forma:

$\sigma_{\theta k}^2$  : varianza del  $k$ -ésimo elemento del bloque transformado

$\frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\theta k}^2$  : promedio de la suma de las varianzas de los elementos de  $\theta$

$\sigma_x^2$  : varianza de cada elemento del bloque de entrada  $x$  (para todos es la misma)

$\frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\theta k}^2 = \sigma_x^2$  ( *el promedio de la suma de las varianzas de los elemento de  $\theta$  es igual a la varianza de cada elemento del bloque de entrada  $x$  , debido a la ortonormalidad de la matriz* )

$\sigma_{r k}^2$  : varianza del error de la  $k$ -ésima muestra



$\sigma_r^2$  : varianza del error promedio del bloque codificado

$\sigma_{qk}^2$  : varianza introducida por la cuantización de la k-ésima muestra  
( varianza de la k-ésima muestra cuantizada)

$\sigma_q^2$  : varianza introducida por la cuantización del bloque transformado (promedio de la varianza del bloque)

$r = x - y$  : vector de errores

$\frac{1}{N} \sum_{k=0}^{N-1} \sigma_{rk}^2$  : promedio de la suma de las varianzas de los errores de cada muestra

$\sigma_r^2 = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{rk}^2$  ( la varianza del error del bloque codificado es igual al promedio de la suma de la varianza del error de cada una de las muestras )

$\sigma_q^2 = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{qk}^2$  ( la varianza introducida por la cuantización del bloque transformado es igual al promedio de la suma de las varianzas introducidas al cuantificar cada muestra)

$\sigma_r^2 = \sigma_q^2$  ( la varianza del error es igual a la varianza introducida en la cuantización del bloque, debido a la ortonormalidad mencionada anteriormente)

Con esto se concluye que : **la varianza del error de codificación en BTC es igual a la introducida por el conjunto de cuantizadores de coeficientes.**

Los componentes de la varianza introducida en la cuantización del bloque pueden escribirse de la siguiente forma:

$$\sigma_{ijk}^2 = \varepsilon_{ijk}^2 \sigma_{\theta k}^2 \quad [\text{JAY84}]$$

$\varepsilon_{ijk}^2$  : factor de performance del cuantizador correspondiente ( k-ésimo ), su valor ideal es 0

Este factor depende de la varianza del elemento de entrada, de la función de distribución de probabilidad (pdf) del k-ésimo coeficiente transformado y de la característica del cuantizador, especialmente la cantidad de niveles (aquí está centrado nuestro trabajo, o sea, como obtener una buena relación de compresión teniendo en cuenta bandas de cuantización para cada uno de los bloques o frames).

Dicho factor puede ser aproximado como sigue :

$$\varepsilon_{ijk}^2 = \varepsilon^2 2^{-2R_k}$$

$\varepsilon^2$  : factor de corrección variable que depende de la performance del cuantizador, para una distribución específica de la entrada.

Entonces puede deducirse que :

$$\sigma_{ijk}^2 \approx \varepsilon^2 2^{-2R_k} \cdot \sigma_{\theta k}^2$$

$R_k$  : cantidad de bits con que se cuantiza el k-ésimo coeficiente

Si tomamos la aproximación como una igualdad ( aunque dicha aproximación sea grosera para niveles bajos de  $R_k$  ) :

$$R_k = \frac{1}{2} \log_2 \varepsilon^2 + \frac{1}{2} \log_2 \frac{\sigma_{\theta k}^2}{\sigma_{ijk}^2} \quad k = 0, 1, \dots, N-1$$

El problema de optimización de un esquema BTC, puede establecerse como:

- hallar una matriz ortogonal - en general unitaria - ( representa a la transformada )
- hallar una distribución de bits  $R_k$  tal que :

$$\sigma_q^2 = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{qk}^2 \quad (\text{varianza introducida por la cuantización del bloque})$$

*transformado es igual al promedio de la suma de las varianzas introducidas al cuantificar cada muestra)*

esta sumatoria debe minimizarse con la restricción que el promedio de bits usados para cuantizar los coeficientes de un bloque, debe ser un valor constante:

$$R = \frac{1}{N} \sum_{k=0}^{N-1} R_k \quad (\text{Restricción})$$

De esto también se deduce que R debe ser igual para todos los bloques a codificar.-

Para concluir con este desarrollo, escribiremos la varianza del error del bloque codificado  $\sigma_r^2$  como una función de los coeficientes transformados  $\sigma_{\theta k}^2$  y de  $R_k$  :

$$\sigma_r^2 = \sigma_q^2 = \frac{1}{N} \sum_{k=0}^{N-1} \epsilon^2 \cdot 2^{-2R_k} \cdot \sigma_{\theta k}^2$$

Por lo tanto el problema es minimizar la varianza introducida al cuantizar el bloque ( mediante la alocaión óptima de bits ). Para ésto se puede utilizar multiplicadores de Lagrange, con la *Restricción*

$$\frac{\delta}{\delta R_k} \left[ \sigma_r^2 - \lambda \left( R - \frac{1}{N} \sum_{k=0}^{N-1} R_k \right) \right] = 0 \quad \text{para } k=0,1,\dots,N-1 ; \text{ donde}$$

$$\sigma_{\hat{c}_k}^2 = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\hat{c}_k}^2 = \frac{1}{N} \varepsilon^2 \sum_{k=0}^{N-1} 2^{-2R_k} \cdot \sigma_{\theta_k}^2$$

**Mediante la resolución del sistema de ecuaciones anterior, se obtiene una asignación óptima de bits para el el k-ésimo coeficiente, expresada por :**

$$R_k = R + \frac{1}{2} \log_2 \frac{\sigma_{\theta_k}^2}{\left[ \prod_{j=0}^{N-1} \sigma_j^2 \right]^{\frac{1}{N}}} \quad \text{(A.O.B.)}$$

A partir de esta fórmula se ve que  $R_k$  depende solamente de la distribución de la varianza de los coeficientes.

(A.O.B. hace que el error de cada cuantizador sea igual para todos los coeficientes).

$L_k$  : número de niveles de cuantización (por cada coeficiente se elije entre  $L_k$  niveles)

$$L_k = 2^{\hat{L}_k} \text{ es proporcional a la varianza } \sigma_{\theta_k}^2 \quad (L_k \propto \sigma_{\theta_k}^2)$$

La asignación de bits también implica varianzas de error idénticas en los coeficientes de cuantización. Distribuye bits de tal modo que los errores sean “parejos”.

$$\min\{\sigma_{\hat{c}_k}^2\} = \min\{\sigma_{\hat{c}_j}^2\} = \sigma_{\hat{c}_k, \text{ óptimo}}^2 = \varepsilon^2 2^{-2R} \left[ \prod_{j=0}^{N-1} \sigma_j^2 \right]^{\frac{1}{N}} \quad \text{con } k = 0, 1, \dots, N-1$$

Para el caso de cuantizadores uniformes, surge un mismo error de cuantización si todos tienen el mismo “paso”  $\Delta$ , y  $\sigma_{\theta_k}^2 = \frac{\Delta^2}{12}$  [RAB78]. Por lo tanto, un

coeficiente con varianza más alta necesita más niveles que uno con menor varianza; esto es garantizado por (A.O.B.).

Como fue expresado anteriormente, la ganancia de BTC ( $G_{BTC}$ ) sobre PCM para bloques de N-muestras es:

$$SNR_{BTC}(dB) = SNR_{PCM}(dB) + 10 \log G_{BTC}$$

La relación señal-ruido (SNR) nos sirve para evaluar la calidad de la señal codificada.

Ahora, cómo obtenemos  $G_{BTC}$  :

$\sigma_{1,PCM}^2$  : varianza del error promedio de un bloque en PCM

$\sigma_{1,BTC}^2$  : varianza del error promedio de un bloque en BTC

$\sigma_x^2$  : varianza del un bloque de entrada x

$$\frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\theta k}^2 = \sigma_x^2 \quad (\text{el promedio de la suma de las varianzas de los elemento de } \theta \text{ es igual a } \sigma_x^2)$$

Entonces, usando lo antes mencionado :

$$\max\{G_{BTC}\} = \frac{\sigma_{1,PCM}^2}{\min\{\sigma_{1,BTC}^2\}} = \frac{\epsilon_{PCM}^2 2^{-2k} \sigma_x^2}{\epsilon_{BTC}^2 2^{-2k} \left[ \prod_{k=0}^{N-1} \sigma_{\theta k}^2 \right]^{1/N}} = \frac{\epsilon_{PCM}^2 \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\theta k}^2}{\epsilon_{BTC}^2 \left[ \prod_{k=0}^{N-1} \sigma_{\theta k}^2 \right]^{1/N}}$$

A partir de aquí, se ve que la  $\max\{G_{BTC}\}$  tiene la forma de una relación entre una media aritmética (la sumatoria) y una media geométrica (la productoria), multiplicada por una constante (factor de corrección variable de PCM sobre el factor de BTC). Esta relación puede expresarse como:

$$\max\{G_{BTC}\} \propto \frac{\overline{\oplus\{\sigma_{\theta k}^2\}}}{\overline{\otimes\{\sigma_{\theta k}^2\}}}$$

*Se debe tener en cuenta que el máximo valor de  $G_{BTC}$  expresado anteriormente, es con respecto a alocaación de bit y al tamaño del bloque, no con respecto al tipo de transformada lineal utilizada.*

La ganancia de BTC sobre PCM es una función del tamaño del bloque o frame, por una lado, y de las estadísticas fuente y el tipo de transformada, por el otro.

Para finalizar, los cuantizadores de coeficientes tienen la restricción de usar valores enteros no negativos para representar las cantidades de bits  $R_k$ . Pero es necesario remarcar que existe la posibilidad de obtener valores negativos para  $R_k$  (caso en que la varianza del k-ésimo coeficiente transformado es “suficientemente” más pequeña que la media geométrica, en A.O.B.). Para estos casos se realiza una reasignación del valor de  $R_k$ ,  $R_k = 0$ , para los cuales  $\sigma_{ijk}^2 = \sigma_{\theta k}^2$  (no se introduce error de cuantización). Claramente, cuando los  $R_k$  negativos son llevados a 0, los valores de  $R_k$  positivos también deben ser ajustados para mantener el valor de  $R$  (cantidad promedio de bits por bloque). Un algoritmo para esta reoptimización es dado por Adaptive Transform Coding (ATC), del cual se obtiene que :

$$R_k = \max\{0, (1/2)\log_2(\sigma_{\theta k}^2 / D)\}$$

en la cual  $D$  es la distorsión media, o sea que se ajustará de modo tal, que el valor asignado a  $R_k$  no altere  $R$ .

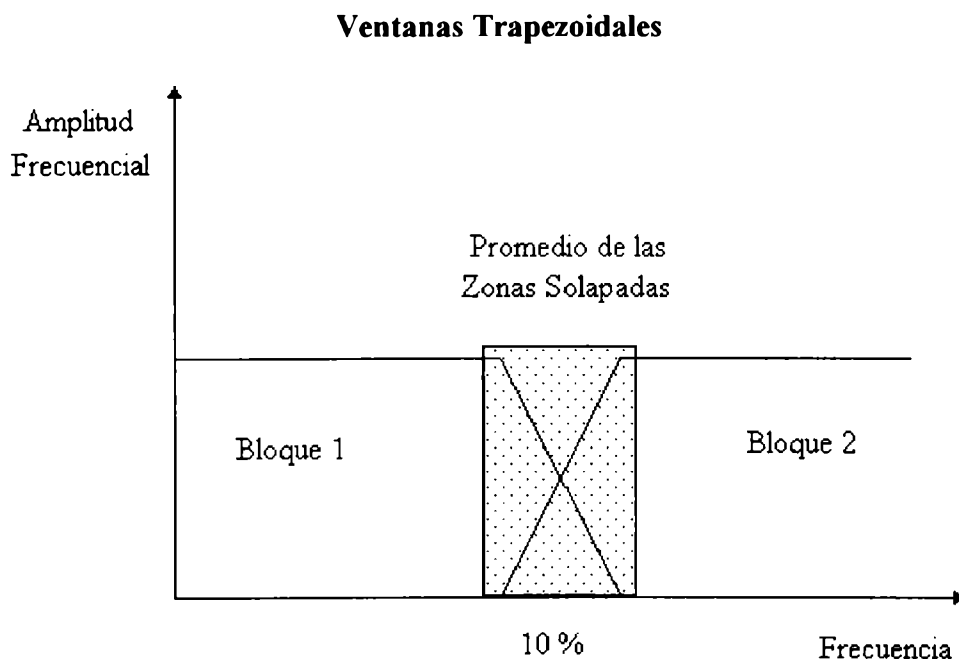
#### 4.2.5 Problemas de Codificación Utilizando BTC

Como expresamos anteriormente, debido a la cuantización, surgen problemas de distorsión de bloque. Este problema es una consecuencia natural del procesamiento independiente de cada bloque. La distorsión de bloque se percibe como tonos extraños.

Tipos de distorsión:

- **Clicking y Bubbling**: cuando los bloques o frames reconstruidos se concatenan en el decodificador surgen discontinuidades en los bordes. En los límites de cada bloque existen saltos que se escuchan como "clicks" o "bubbles" [ O'SH87]. Esta distorsión puede ser parcialmente eliminada mediante la superposición de muestras consecutivas. La porción de bloque a superponer es el 10% de cada lado del mismo. Esto se realiza mediante el uso de ventanas trapezoidales. Las mismas son frecuentemente usadas en la práctica durante el cambio de dominio para reducir la distorsión de la señal producida por la discontinuidad de bloques. Con esta técnica, el receptor promedia las muestras reconstruidas de bloques consecutivos en las áreas superpuestas, atenuando de ese modo el efecto de bordes (Fig. 4.4) [DEL93]. La desventaja de esta solución es el incremento de la cantidad de muestras procesadas, incrementado también así el bit rate. Como se mencionó anteriormente, utilizando DCT, tales efectos son menos severos.

- **Pre-echoes** : es una distorsión producida por la cuantización, para los casos en los que un período de silencio es seguido por un sonido de percusión, dentro de un mismo bloque. En ATC, la transformación lineal inversa distribuye dicha distorsión en todo el bloque, especialmente a tasa de bits bajas [NOL93]



**Fig. 4.4**

# Capítulo 5 : CODIFICACION EN EL ESPECTRO UTILIZANDO BTC

## 5.1 Etapas de Codificación en el Espectro

El proceso de codificación y decodificación de una señal utilizando **BTC** se divide en 4 etapas [BST95]:

- I. cambio del dominio temporal al dominio frecuencial o espectral
- II. codificación de las líneas espectrales (muestras en el espectro)
- III. decodificación de las líneas espectrales
- IV. cambio del dominio frecuencial al dominio temporal

### 5.1.1 Cambio de Dominio Temporal al Dominio Frecuencial

Primeramente se divide la señal de audio de entrada en **bloques o frames** de longitud fija  $N$ . La longitud de cada bloque es de 1025 muestras, de las cuales 25 son solapadas entre bloques consecutivos (**ventanas trapezoidales**) para atenuar los problemas de distorsión de lados (**clicking y burbling**). En nuestro caso la señal está digitalizada con una resolución de 16 bits utilizando 1 o 2 canales (mono o estéreo) y muestreada a 44.100 Hz. Para cambiar de dominio se debe emplear una transformación lineal adecuada ya que el objetivo es "decorrelacionar" las muestras. A fin de lograr tal objetivo, adoptamos el uso de **DCT** (mencionada anteriormente), ya que además atenúa los efectos de bordes en los límites de bloques.

La decorrelación es necesaria ya que nos permite eliminar algunas líneas sin causar una pérdida significativa en la información de la señal. Por lo tanto de aquí en más la consideramos como hipótesis de trabajo para las etapas siguientes.



### 5.1.2 Codificación de las Líneas Espectrales

La técnica de codificación que utilizamos es ATC, ya que es la que más aproxima la forma de onda original, al establecer umbrales diferentes para cada bloque, o sea , se adapta a los cambios de la señal, y usa una cantidad fija de bits para cada frame, los cuales se distribuyen en forma proporcional al valor real discreto de cada una de las muestras.

Esta etapa se subdivide en 2 pasos que, si bien no son totalmente disjuntos, clarifican el proceso de codificación. Dichos pasos son:

a. Asignación de bits a las líneas espectrales de un frame

b. Cuantización de líneas espectrales de un frame

Teniendo en cuenta que nos surgieron diferentes ideas para lograr mayor compresión durante la codificación, podemos establecer una serie de variantes que marcaron el progreso de nuestro trabajo. Tales variantes fueron:

1) Codificar cada línea espectral de un frame con un número de bits específico

2) Dividir cada bloque en bandas de long fija

- 2.1. Usando cantidad máxima de Bits (**BMAX**)
- 2.2. Usando cantidad promedio de Bits (**BPROM**)
- 2.3. Usando la Desviación Standard (**BDS**)
- 2.4. Usando la Desviación Media (**BDM**)

3) Dividir cada bloque en bandas de longitud variable

- 3.1. Usando cantidad máxima de Bits (**BMAX**)
- 3.2. Usando cantidad promedio de Bits (**BPROM**)
- 3.3. Usando la Desviación Standard (**BDS**)
- 3.4. Usando la Desviación Media (**BDM**)

4) Establecer cantidades variables de bits para codificar cada frame según la tasa de compresión deseada, combinado con 3.

1) Debido a que BTC proporciona teóricamente digitalizaciones de alta calidad a una tasa promedio de 2 bits por muestra, utilizamos 2050 ( $2 \cdot 1025$ ) bits para codificar todo el frame. A partir de esta hipótesis inicial, establecimos el *umbral* del bloque y la *cantidad de bits* para codificar cada coeficiente del mismo, del siguiente modo:

**Proceso 1 : Establecer Umbral y Asignación de Bits**

- $N$  = 1025 ( tamaño del bloque )
- $i$  = 1...N
- $Sumalg_2$  =  $\sum \lg_2(\text{Muestra}_i)$
- $TotB$  = cantidad de bits para codificar todo el bloque
- $Umbral$  =  $( Sumalg_2 - TotB ) / N$
- $Int(x)$  = parte entera más cercana superior(  $x$  )
- $Bits[i]$  = cantidad de bits para codificar la  $Muestra_i$

**Proc 1.1**

Comienzo **Proc 1.1**

- $Bits[i] = Int(\lg_2(\text{Muestra}_i) - Umbral)$  si  $\lg_2(\text{Muestra}_i) - Umbral > 0$
- $Bits[i] = 0$  si  $\lg_2(\text{Muestra}_i) - Umbral \leq 0$
- $Signo[i] = 1$  si  $\text{Muestra}_i < 0$
- $Signo[i] = 0$  si  $\text{Muestra}_i \geq 0$

$SumaB = \sum Bits[i]$

$Mposit$  = cantidad de muestras con  $Bits[i] > 0$

Fin **Proc 1.1**

Mientras (  $SumaB > TotB$  )

$Umbral = Umbral + ( SumaB - TotB ) / Mposit$  {distribuir la diferencia}

**Proc 1.1**

Fin Mientras

**Proceso 2 : Cuantización**

$$Diferencia_i = \lg_2( Muestra_i ) - Umbral$$

Si  $Bits[i] > 0$

$$MuestraCuantizada_i = \text{Int} ( ( Bits[i] - Diferencia_i ) * 2^{Bits[i]} )$$

sino

$$MuestraCuantizada_i = 0$$

Fin Si

**Proceso 3: Almacenamiento**

El formato de la palabra de datos a grabar, la cual representa cada línea espectral, es el siguiente:

Palabra de Datos 1

Si  $Bits[i] > 0$

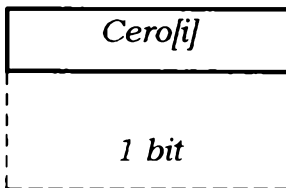
$$Cero[i] = 0$$

$Cero[i]$	$Bits[i]$	$Signo[i]$	$MuestraCuantizada_i$
1 bit	4 bits	1 bit	2 bits promedio

Palabra de Datos 2

Si  $Bits[i] = 0$

$$Cero[i] = 1$$



2) Como se ve en el punto 1), cada coeficiente espectral tiene una cantidad de bits asociada. Este tipo de codificación es óptima en lo referido a calidad de reproducción, pero no lo es desde el punto de vista del espacio requerido. Entonces vimos la necesidad de cambiar la codificación de manera tal que, varias muestras usen la misma cantidad de bits para ser representadas ( la cantidad indicada por la banda).

De este modo surgió la idea de subdividir cada bloque en subbloques de longitud fija, los cuales denominamos *bandas*. Obviamente a menor cantidad de bandas menor pérdida, pero luego de varias pruebas deducimos que usando entre 20 y 27 la degradación era aceptable. Por lo tanto establecimos 25 bandas por frame. Cada una de ellas con 41 muestras.

Para establecer el umbral y realizar la asignación de bits agregamos al **Proceso 1** :

**Proc 1.2 : Establecer las Bandas de longitud fija**

$K = \text{cantidad de bandas} = 25$

$k = 1 \dots 25$

$Mueband = \text{cantidad de muestras por banda} = 41$

$Bandas[k] = \text{cantidad de bits para codificar las muestras de la Banda}_k$

$Bmax_k = \text{máx}(Bits[i]) \quad \forall i \text{ tal que } k * Mueband \leq i < (k+1) * Mueband$

$Bprom_k = \text{promedio}(Bits[i]) \quad \forall i \text{ tal que } k * Mueband \leq i < (k+1) * Mueband$

$DM_k = \text{Desviación Media para la } k\text{-ésima banda}$

$DS_k = \text{Desviación Standard para la } k\text{-ésima banda}$

$DM_k = ( \sum \text{valor absoluto} ( Bits[i] - Bprom_k ) ) / Mueband$

$DS_k = \text{raiz cuadrada} ( ( \sum ( Bits[i] - Bprom_k ) ^ 2 ) / Mueband )$

$Bandas[k] = Bmax_k$  para 2.1

$Bandas[k] = Bprom_k$  para 2.2

$Bandas[k] = Bprom_k + DM_k$  para 2.3

$Bandas[k] = Bprom_k + DS_k$  para 2.4

### **Proceso 2 : Cuantización**

$Diferencia_i = \lg_2(Muestra_i) - Umbral$

Si  $Diferencia_i > 0$  .y  $Bandas[k] > 0$

$MuestraCuantizada_i = Int ( ( Diferencia_i / Bandas[k] ) * 2^{Bandas[k]} )$

sino

$MuestraCuantizada_i = 0$

Fin Si

### **Proceso 3: Almacenamiento**

El nuevo formato de la palabra de datos a grabar, es el siguiente:

Palabra de Datos 1

Si  $Bandas[k] > 0$

$$Cero[k] = 0$$

$Bandas [k]$	$Cero[k]$
4 bit	1 bits

$Signo[1]$	$MuestraCuantizada_1$
1 bit	2 bits promedio

.....  
.....

$Signo[N]$	$MuestraCuantizada_N$
1 bit	2 bits promedio

Palabra de Datos 2

Si  $Bandas[k] = 0$

$$Cero[k] = 1$$

$Cero[k]$
1 bit

3) Con el mismo propósito de codificación en bandas quisimos obtener mayores ventajas de esta técnica. Una alternativa fue cambiar el número de bandas, pero vimos que la ganancia no era lo suficiente como para justificar dicho cambio. Por lo tanto pensamos en bandas de longitud variable que, combinadas con la Desviación Media o Standard nos diera mayores beneficios; entonces adecuamos esta idea a los principios psicoacústicos. Con esta meta agrupamos los coeficientes transformados en bandas no uniformes que reflejan el sistema auditivo humano.

Este tipo de análisis indica cuáles frecuencias son críticas y deben ser codificadas con alta precisión, y cuáles son menos sensitivas y pueden tolerar algún ruido de cuantización sin degradación de la calidad percibida del sonido. La sensibilidad del oído varía con las frecuencias, es más sensible a frecuencias cercanas a 4 kHz. [TSU92]. Las bandas no uniformes surgen naturalmente a partir de experiencias en audición humana, y pueden también derivarse a partir de la distribución de las células sensoriales en el oído interno. Por lo tanto las bandas pueden pensarse como la escala de frecuencias usada por el oído.

Dichas bandas son más estrechas en frecuencias bajas que en frecuencias altas, en efecto, el 75% de ellas están ubicadas por debajo de los 5 kHz. Esto indica que el oído recibe más información de las bajas frecuencias que de las altas.

La tabla 1 muestra el tamaño de las bandas en Hz.

**TABLA 1: BANDAS NO UNIFORMES**

<b>Bandas</b>	<b>Lím. Inferior (Hz.)</b>	<b>Lím. Superior (Hz.)</b>	<b>Ancho (Hz.)</b>	<b>Cantidad de muestras</b>
0	0	100	100	4
1	100	200	100	4
2	200	300	100	5
3	300	400	100	5
4	400	510	110	5
5	510	630	120	6
6	630	770	140	6
7	770	920	150	7
8	920	1080	160	8
9	1080	1270	190	9
10	1270	1480	210	9
11	1480	1720	240	11
12	1720	2000	280	13
13	2000	2320	320	15
14	2320	2700	380	18
15	2700	3150	450	21
16	3150	3700	550	25
17	3700	4400	700	33
18	4400	5300	900	42
19	5300	6400	1100	51
20	6400	7700	1300	60
21	7700	9500	1800	84
22	9500	12000	2500	116
23	12000	15500	3500	163
24	15500	22050	6550	305

Con el esquema hasta aquí descrito los **Procesos 1, 2 y 3** son similares al punto 2), con la excepción que :



$Mueband = \text{cantidad de muestras por banda ( ver TABLA 1 )}$

4) Los mejores resultados, que veremos en detalle más adelante, los obtuvimos con la codificación con bandas de longitud variable. A partir de ello, buscamos aumentar la tasa de compresión y que el algoritmo sea capaz de recibir un parámetro con la tasa deseada (obviamente con restricciones). Hasta este punto la cantidad inicial de bits para cada frame fue uniforme, de aquí en más dependerá del nivel de compresión recibido como parámetro.

Según el nivel requerido (entre 3 y 12 veces) la cantidad de bits por bloque es la siguiente:

**TABLA 2: CANTIDAD DE BITS POR FRAME O BLOQUE**

<b>Tasa</b>	<b>Cantidad de Bits</b>
3	4150
4	2800
5	2050
6	1450
7	1100
8	800
9	500
10	290
11	150
12	100



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

Los **Procesos 1, 2 y 3** son similares al punto 3), con la excepción que :

$TotB = \text{cantidad de bits para codificar todo el bloque ( ver TABLA 2 )}$

Las cantidades establecidas son resultados de sucesivas pruebas, y aún podrían ajustarse más si se quisieran aumentar los niveles propuestos.

### 5.1.3 Decodificación de las Líneas Espectrales

Este proceso es el inverso a la etapa II., y teniendo en cuenta la palabra de datos almacenada, el modo de reconstruir las muestras es el siguiente:

$$DiferenciaReconstruida_i = Bandas[k] * ( MuestraCuantizada_i / 2^{Bandas[k]} )$$

$$\lg_2Reconstruido(Muestra_i) = DiferenciaReconstruida_i + Umbral$$

$$MuestraReconstruida_i = 2^{\lg_2Reconstruido(Muestra_i)} * Signo[i]$$

Todas las muestras de una misma banda se reconstruyen con la misma cantidad de bits indicada por  $Bandas[k]$ .

Lógicamente el algoritmo aplicado en esta etapa está supeditado al esquema de codificación, ya que aquí solamente se leen las muestras grabadas y se reconstruyen aplicando el método inverso.

### 5.1.4 Cambio del Dominio Frecuencial al Dominio Temporal

Esta etapa depende también de la transformada usada en el paso I. . Aquí sólo se aplica la transformación inversa.

## 5.2 Criterios de Fidelidad

Cualquier evaluación de la calidad de señal implica una medición de la fidelidad. Para la mayoría de los sistemas de comunicación es difícil de especificarla cuantitativamente, debido a que esto involucra la percepción humana. Por lo tanto, la calidad de audio es tradicionalmente evaluada por el criterio de quien escucha.

Mientras la calidad de cada técnica de compresión es subjetiva, es implícito que a medida que el radio de compresión aumenta la calidad disminuye, para una frecuencia de muestreo establecida.

Básicamente, cada técnica de compresión tiene un límite de calidad para diferentes radios de la misma. Esto significa que no existe una única técnica de

compresión que sea la ideal para todas las aplicaciones potenciales. La selección de una técnica apropiada es un elemento clave de cualquier aplicación en la que participen señales de audio.[SOS95]

### 5.3 Resultados Obtenidos

Las pruebas realizadas nos indicaron que este método de codificación no es aplicable con los mismos resultados a cualquier tipo de señal de audio. Específicamente, a un nivel de compresión deseado, la calidad de reproducción obtenida no es igual en todos los casos.

El estudio de la performance del método se realizó con los siguientes tipos de señales de audio:

- música pop           ( i )
- música clásica       ( ii )
- speech               ( iii )
- efectos sonoros      ( iv )

Los resultados se reflejan en la siguientes tablas, y valen tanto para señales mono como estéreo:

**Cuadro 1 : Asignación de bits individual a cada línea espectral**

	<b>2 bits promedio</b>
Compresión	1:2
Calidad (i)	Excelente
Calidad (ii)	Excelente
Calidad (iii)	Excelente
Calidad (iv)	Excelente

**Cuadro 2 : Bandas de longitud fija**

	<b>BPROM</b>	<b>BMAX</b>	<b>BDM</b>	<b>BDS</b>
Compresión	de 1:3 a 1:4	de 1:2.5 a 1:3	de 1:3 a 1:4	de 1:3 a 1:3.5
Calidad (i)	Mala	Excelente	Muy Buena	Muy Buena
Calidad (ii)	Mala	Excelente	Buena	Muy Buena
Calidad (iii)	Mala	Excelente	Muy Buena	Muy Buena
Calidad (iv)	Mala	Excelente	Muy Buena	Muy Buena

**Cuadro 3 : Bandas de longitud variable**

	<b>BPROM</b>	<b>BMAX</b>	<b>BDM</b>	<b>BDS</b>
Compresión	de 1:3 a 1:4	1:2.5	1:5	de 1:3 a 1:4
Calidad (i)	Mala	Excelente	Muy Buena	Muy Buena
Calidad (ii)	Mala	Excelente	Buena	Buena
Calidad (iii)	Mala	Excelente	Buena	Muy Buena
Calidad (iv)	Mala	Excelente	Buena	Muy Buena

**Cuadro 4 : Bandas de longitud variable, con variación de la cantidad de bits por frame**

Tasa	Cantidad de Bits	Calidad (i)	Calidad (ii)	Calidad (iii)	Calidad (iv)
3	4150	Excelente	Muy Buena	Excelente	Excelente
4	2800	Muy Buena	Buena	Muy Buena	Muy Buena
5	2050	Muy Buena	Regular	Buena	Buena
6	1450	Buena	Mala	Regular	Buena
7	1100	Buena	Mala	Regular	Regular
8	800	Buena	Mala	Mala	Regular
9	500	Buena	Mala	Mala	Mala
10	290	Regular	Mala	Mala	Mala
11	150	Regular	Mala	Mala	Mala
12	100	Regular	Mala	Mala	Mala

Obviamente estos resultados dependen del oído subjetivo del receptor y de las muestras analizadas dentro de cada tipo de señal; por lo tanto no los consideramos absolutos sino como un referente o patrón. Aunque, cabe destacar que los mejores logros los encontramos codificando música pop y según el ejemplo usado, la compresión 1:10 o 1:12 es aceptable en lo referido a calidad de reproducción.

Para mostrar *objetivamente* que tales logros fueron exitosos, calculamos la relación señal ruido total y la relación señal ruido segmental, para los resultados obtenidos en el cuadro 4, con lo que se observa que la codificación de música pop fue la más aceptable para los distintos niveles de compresión:

***SNR para Bandas de longitud variable, con variación de la cantidad de bits por frame.***

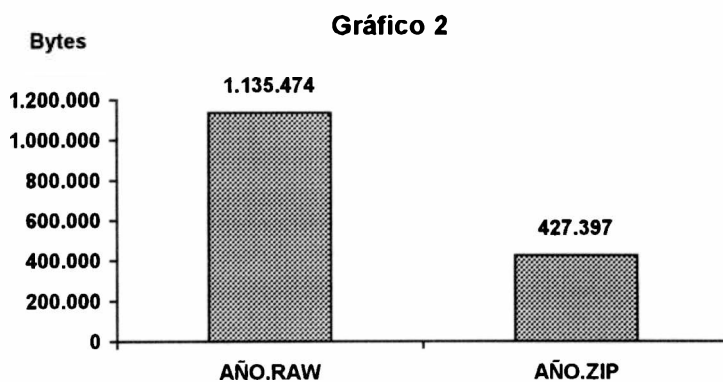
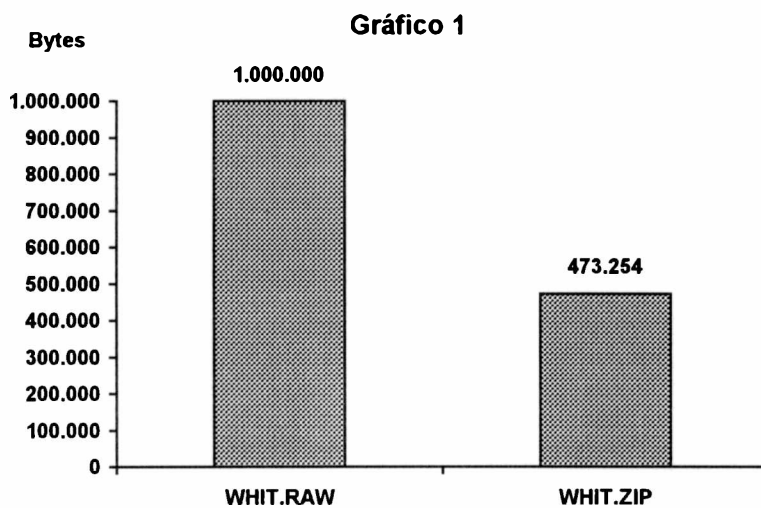
Tasa	Cantidad de Bits	Calidad (i)	Calidad (ii)	Calidad (iii)	Calidad (iv)
3	4150	14.24509 dB	2.57614 dB	19.49338 dB	2.60884 dB
5	2050	14.68057 dB	2.58396 dB	20.13647 dB	3.53323 dB
10	290	13.46096 dB	2.58777 dB	18.34249 dB	1.92300 dB

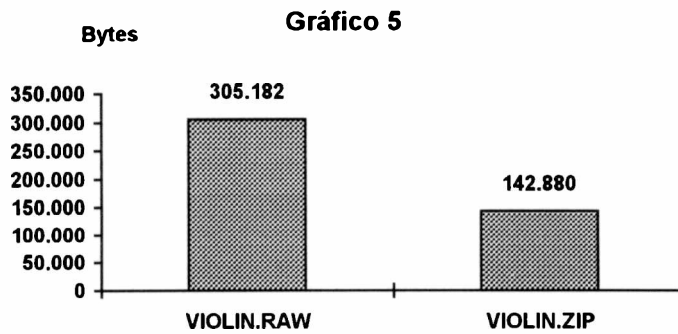
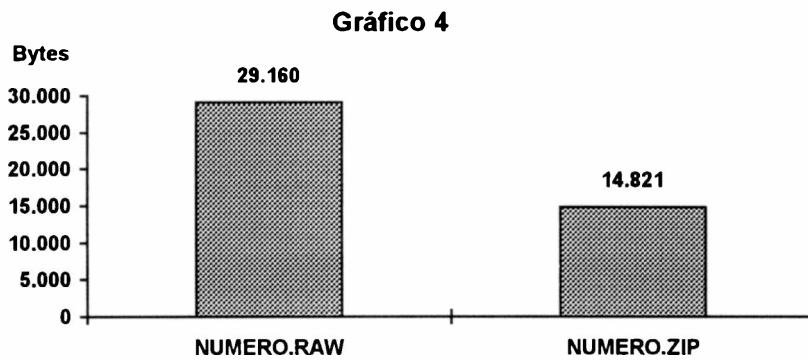
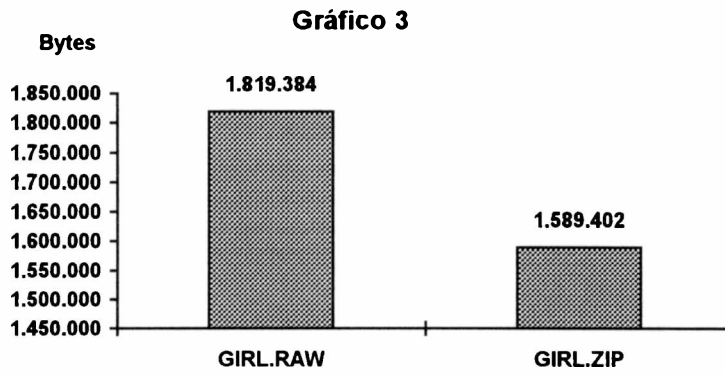
***SNR segmental para Bandas de longitud variable, con variación de la cantidad de bits por frame.***

Tasa	Cantidad de Bits	Calidad (i)	Calidad (ii)	Calidad (iii)	Calidad (iv)
3	4150	20.91071 dB	2.39821 dB	20.72236 dB	2.57475 dB
5	2050	21.55829 dB	2.43880 dB	21.27642 dB	3.49948 dB
10	290	19.05656 dB	2.39646 dB	17.56410 dB	1.89441 dB

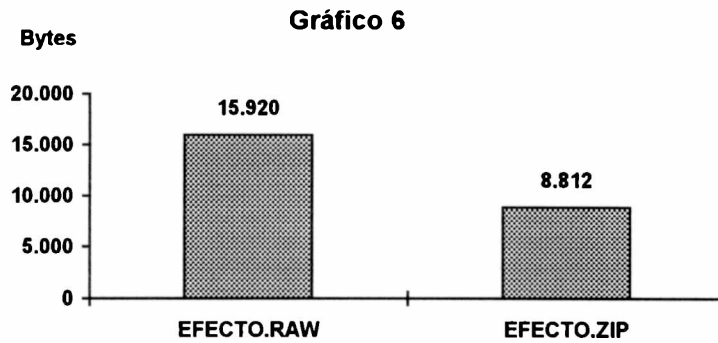
También realizamos la comparación con los resultados de un algoritmo de compresión sin pérdida tipo Pkzip como se muestra en la siguiente tabla y los gráficos correspondientes.

Tipo de Señal	Gráfico	Compresión
Música Pop Mono	1 y 2	de 2 a 2.5 veces
Música Pop Estéreo	3	1.14 veces
Música Clásica	5	2.13 veces
Speech	4	1.96 veces
Efectos Sonoros	6	1.80 veces









## 5.4 Limitaciones Prácticas en PC's. Implementaciones en Tiempo Real

Por su naturaleza, la compresión de audio es un proceso de tiempo real. Históricamente, la arquitectura de las PC's (sistema operativo y hardware) no fue diseñada para soportar operaciones en tiempo real en forma óptima.

Con la demanda creciente sobre las PC's para soportar manejo sofisticado de gráficos y proveer acceso a sistemas de archivos, el procesamiento de audio fue relegado a un plano secundario. Con este panorama existen dos alternativas posibles: ajustar el procesamiento de audio en las MIPS (millones de instrucciones por segundo) restantes, o tener un coprocesador dedicado que pueda satisfacer los requerimientos de tiempo real para dicho procesamiento.

Aquí es donde se manifiestan los beneficios de los procesadores de señales digitales (DSPs). La arquitectura de los DSPs ha sido optimizada desde sus comienzos para soportar las necesidades de los sistemas de tiempo real. El DSP está diseñado para realizar una gran cantidad de operaciones en cada ciclo de reloj. Esta unidad de trabajo es consistente e incorpora la capacidad de realizar un cálculo acumulativo múltiple (MAC) y al mismo tiempo efectuar sumas u otro tipo de operaciones ALU.

Debido a esto, podemos decir que la esencia de las variadas técnicas de compresión de audio es un tipo de cálculo conocido como "suma de productos", y los DSP's proporcionan un soporte perfecto para ellas [REI94].

## CONCLUSION

La Codificación Digital de Audio en el Espacio Transformado, es una metodología muy útil para obtener excelentes resultados en lo referente a compresión sin degradación de la calidad sonora.

Con dicha metodología a través de la combinación de varias técnicas incluyendo: división de la señal de audio en frames, separación del espectro en bandas de longitud variable, y asignación de bits según la longitud de las bandas y principios psicoacústicos, logramos obtener una relación de compresión entre 3 y 12 veces. La calidad obtenida depende del tipo de información que contiene la señal que se comprime, siendo la música pop la que soporta la tasa más alta ( entre 10 y 12 ) sin pérdida significativa para el oído humano.

---

## ***Bibliografía***

[AHM83] Nasir Ahmed and T. Natarajan, *Discrete-Time Signals and Systems*, Prentice-Hall, 1983.

[BRI95] Oscar N. Bria, "**Block Transform Coding for Audio Signals**", Reporte Técnico CETAD, Marzo 1995.

[BST95] Oscar N. Bria, Ramiro F. Sosa y Pablo J. Thomas, "**Compresión de Audio en el Espacio Transformado**", 1<sup>er</sup> Congreso Argentino de Ciencias de la Computación (CACIC '95), 3<sup>er</sup> Ateneo de Profesores Universitarios Investigadores (APUI '95) y 2<sup>do</sup> Workshop sobre aspectos teóricos de la Inteligencia Artificial (ATIA '95). Octubre 1995.

[COX81] R. Cox and R. Crochiere, "**Real-Time Simulation of Adaptive Transform Coding**", *IEEE Transaction on Acoustics, Speech, and Signal Processing*, April 1981.

[FLA79] J. Flanagan et. al., "**Speech Coding**", *IEEE Transaction on Communications*, April 1979.

[DEL93] J.R. Deller et. al., *Discrete-Time Processing of Speech Signals*, Macmillan, 1993.

[IWA92] M. Iwadare et. al., "**A 128 Kb/s Hi-Fi Audio CODEC Based on Adaptive Transform Coding with Adaptive Block Size MDCT**", *IEEE Journal on Selected Areas in Comm*, January 1992.

[JAY84] S.N. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984.

- 
- [NEL94] J. Nelson, *The Data Compression Book*, 1994.
- [NIE89] H.Niemann, *Pattern Analysis and Understanding*, Springer-Verlag, September 1989.
- [NOL93] Peter Noll, "**Wideband Speech and Audio Coding**", *IEEE Communications Magazine*, November 1993.
- [O'SH87] D. O'Shaughnessy, *Speech Communication: Human and Machine*, Addison Wesley, 1987.
- [PEL93] G.E. Pelton , *Voice Processing*, McGraw-Hill, 1993.
- [RAB78] L. Rabiner and R. Schafer, *Digital Processing of Speech Signal*, Prentice Hall, 1978.
- [REI94] J.B. Reimer, "**DSP and Audio Compression**", *Dr. Dobb's Multimedia Sourcebook*, Winter 1994.
- [ROW94] N.G. Rowland, "**Compressing Waveform Audio Files**", *Dr. Dobb's Multimedia Sourcebook*, Winter 1994.
- [SIM86] J.-C.Simon, *Patterns and Operators*, McGraw-Hill, 1986.
- [SOS95] Ramiro F.Sosa, Oscar N.Bria y Pablo J. Thomas, "**Audio Compression in the Transformed Space**", ICIE '95 ( II International Congress on Information).  
Noviembre 1995.
- [SPI94] M. R. Spiegel , *Estadística*, McGraw-Hill, 1991.

[THO94] Pablo J. Thomas, Ramiro F. Sosa y Rodolfo Bertone, "**Análisis de Audio, Algoritmos de Compresión y Descompresión de Archivos de Sonido**". Informe Técnico LIDI. Octubre 1994.

[TRI79] Tribolet and Crochiere, "**Frecuency Domain Coding of Speech**", *IEEE Transaction on Acoustic, Speech, and Signal Processing*, October 1979.

[TSU92] K. Tsutsui et. al. , "**ATRAC: Adaptative Acoustic Coding for Minidisc**", *Audio Eng. Soc. Preprint*, presented at the 93rd Convention AES, October 1992.

[ZEL79] R.Zelinki and P. Noll, "**Approaches to Adaptative Transform Speech Coding at Low Bit Rates**", *IEEE Transactions on Acoustics, Speech and Signal Processing*, February 1979.



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

DONACION..... TES  
\$..... 96/11  
Fecha..... 26-8-05  
Inv. E. ~~16-27~~ Inv. B. 1956

## Agradecimientos

- Al Ingeniero Oscar N. Bria, por su dedicación en la dirección de este trabajo.
- A Licenciado Rodolfo Bertone, que participó inicialmente en la investigación junto a nosotros.
- Al L.I.D.I. (Laboratorio de Investigación y Desarrollo en Informática), por permitirnos utilizar su infraestructura.