



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
MAESTRÍA EN INGENIERÍA ELÉCTRICA – PROCESAMIENTO
DIGITAL DE SEÑALES

**DISEÑO E IMPLEMENTACIÓN DE UN CODIFICADOR
DECODIFICADOR PERCEPTUAL DE AUDIO DIGITAL**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
EDGAR TELLO PALETA

TUTOR
M.I. LARRY HIPÓLITO ESCOBAR SALGUERO
FACULTAD DE INGENIERÍA

CIUDAD DE MÉXICO, NOVIEMBRE DE 2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dra. Lucía Medina Gómez.

Secretario: Dr. Francisco García Ugalde.

1er. Vocal: M.I. Larry Escobar Salguero.

2do. Vocal: Dr. Fernando Arámbula Cosío.

3er. Vocal: Dr. Bohumil Psenicka.

Lugares donde se realizó la tesis:

Laboratorio de Procesamiento Digital de Señales, en el Posgrado de Ingeniería de la UNAM. Planta de transmisión del Departamento Técnico División FM de Grupo Formula.

TUTOR DE TESIS:

M.I. LARRY HIPÓLITO ESCOBAR SALGUERO

FIRMA

Agradecimientos

A mis padres Edmundo Tello y Teresita Paleta

Por su vida ejemplar, gran cariño y apoyo incondicional.

A mis tíos Arturo Paleta y Moisés Tello

Por sus consejos y apoyo.

Al profesor Larry Escobar

Por su apoyo, confianza y paciencia durante el desarrollo del presente trabajo.

Al Profesor Antonio Salvá

Por su sincera amistad.

A mi amigo Octavio Aguirre

Por animarme a estudiar la maestría.

A grupo Fórmula

Por las facilidades otorgadas para la realización del presente trabajo.

Resumen

En esta tesis se presenta el diseño e implementación de un codificador decodificador perceptual de audio digital con tasas de bits de audio comprimido a: 64, 96, 128, 192, 256 y 320 kbps. La señal de entrada para la etapa de codificación se toma de un archivo en formato WAV, conteniendo audio estereofónico en codificación PCM lineal, con frecuencia de muestreo de 44.1 kHz y 16 bits de cuantización. La señal de salida, proveniente de la etapa de decodificación, se almacena en un archivo en formato WAV, con las mismas características del archivo con la señal original. El códec implementado analiza la señal de entrada por bloques sucesivos, traslapados un cincuenta por ciento. Cada bloque de 1024 muestras se transforma al dominio de la frecuencia por medio de la Transformada de coseno discreta modificada (MDCT), obteniendo así, bloques de 512 coeficientes espectrales reales. Los coeficientes en cada uno de estos bloques son cuantizados de forma escalar, utilizando la información procedente de un modelo psicoacústico, y codificados utilizando la técnica de codificación entrópica empleada por el codificador LAME, versión 3.99.5. El modelo psicoacústico empleado en la etapa de codificación, determinado para cada bloque, utiliza los propios coeficientes MDCT para estimar la máxima energía de ruido de cuantización que puede ser introducida de forma imperceptible. Para tratar con la distorsión conocida como pre-eco, inherente a los códec de transformación por bloques, se emplean las técnicas de reservorio de bits y el moldeado de ruido temporal (TNS). Se espera que el presente trabajo sirva de referencia para todo aquel interesado en entender la arquitectura básica utilizada tanto por los códec de MP3 como los códec de AAC (Advanced Audio Coding).

Índice general

1. Introducción	1
2. Compresión de audio digital	5
2.1. Arquitectura genérica de un codificador perceptual de audio.....	6
2.2. Psicoacústica.....	7
2.2.1. Niveles de presión de sonido.....	7
2.2.2. Umbral auditivo.....	8
2.2.3. Enmascaramiento simultaneo.....	9
2.2.4. Bandas críticas y la escala Bark.....	10
2.2.5. Curvas de enmascaramiento.....	13
2.2.6. Umbral global de enmascaramiento.....	15
2.2.7. Enmascaramiento temporal.....	16
2.3. Banco de filtros uniforme máximamente decimado de M bandas.....	18
2.3.1. Estructura del banco de filtros.....	18
2.3.2. Banco de filtros FIR como transformada de bloque.....	20
2.3.3. Banco de filtros PQMF.....	22
2.3.4. Extended Lapped Transform (ELT).....	23
Resumen.....	24
3. Compresor perceptual de audio	25
3.1. Transformada de coseno discreta modificada.....	26
3.1.1. MDCT/IMDCT vía la DCT-IV y el origen del TDAC.....	28
3.1.2. Cálculo de la DCT-IV mediante la FFT.....	30
3.2. Modelo psicoacústico basado en la MDCT.....	31
3.2.1. Transformación al dominio de la frecuencia.....	32
3.2.2. Componentes enmascaradores.....	32
3.2.3. Índice de tonalidad y ajuste del enmascaramiento.....	33
3.3. Cuantización y codificación.....	33

3.3.1. Cuantización.....	34
3.3.2. Colocación del ruido de cuantización.....	35
3.4. Métodos de reducción del pre-eco.....	37
3.4.1. Reservorio de bits.....	37
3.4.2. Moldeado del ruido temporal (TNS).....	37
Resumen.....	40
4. Implementación del códec perceptual	41
4.1. Codificación.....	43
4.2. Banco de filtros MDCT/IMDCT.....	44
4.2.1. Ventana TDAC.....	46
4.2.2. Algoritmo rápido para la MDCT e IMDCT.....	48
4.3. Modelo psicoacústico.....	50
4.3.1. Espectro particionado de energía.....	50
4.3.2. Estimación del offset de enmascaramiento.....	51
4.3.3. Convolución con las funciones de propagación.....	52
4.3.4. Cálculo del umbral global de enmascaramiento.....	54
4.3.5. Entropía perceptual.....	56
4.4. Etapa de cuantización y codificación.....	57
4.4.1. Ciclo externo.....	59
4.4.2. Ciclo interno.....	61
4.4.3. Codificación Huffman.....	61
4.5. Etapas de análisis y síntesis TNS.....	65
4.5.1. Etapa de análisis TNS.....	67
4.5.2. Etapa de síntesis TNS.....	68
4.6. El bitstream.....	68
4.6.1. Encabezado de frame.....	69
4.6.2. Información complementaria.....	70
4.6.3. Datos principales.....	73
4.6.4. El reservorio de bits.....	73
4.7. Decodificación.....	76
4.7.1. Buffer de lectura de bitstream.....	77
4.7.2. Sincronización y lectura de encabezado.....	77
4.7.3. Lectura de información complementaria.....	77
4.7.4. Lectura de los datos principales.....	78
4.7.5. Decodificación de los datos principales.....	78
4.7.6. Recuperación de los coeficientes MDCT.....	79
4.7.7. Reconstrucción de señal de audio PCM lineal.....	79
Resumen.....	80

5. Evaluación y resultados del códec ACC	81
5.1. Método de prueba perceptual.....	82
5.2. Resultados perceptuales.....	83
5.3. Resultados objetivos.....	85
5.4. Discusión de resultados.....	90
6. Conclusiones	91
Apéndices	
A. Abreviaturas	93
B. Demostración de un FB-FIR como TBT	95
C. Demostración DCT-IV vía DFT	99
D. PAM: un ejemplo gráfico	103
Bibliografía.	107

Índice de figuras

2.1	Códec perceptual genérico de audio.....	7
2.2	Umbral auditivo.....	9
2.3	Ejemplo de enmascaramiento simultaneo.....	9
2.4	Aparato auditivo.....	11
2.5	Patrones de excitación para ruido de banda angosta a diferentes frecuencias.....	12
2.6	Índice de banda crítica.....	13
2.7	Patrones de excitación para ruido de banda angosta a diferentes niveles.....	13
2.8	Curva de enmascaramiento.....	14
2.9	Umbral global de enmascaramiento.....	16
2.10	Propiedades de enmascaramiento temporal del sistema auditivo.....	16
2.11	Pre-eco.....	17
2.12	Banco de filtros máximamente decimado de M bandas.....	19
2.13	Proceso de submuestreo en el dominio de la frecuencia.....	19
2.14	Banco de filtros como transformada de bloque.....	21
3.1	Transformada de coseno discreta modificada (MDCT).....	27
3.2	Aliasing en el dominio del tiempo (TDA).....	29
3.3	Cancelación de aliasing en el dominio del tiempo (TDAC).....	30
3.4	Doble ciclo anidado en la etapa de cuantización/codificación.....	36
3.5	Esquema de codificación por predicción de lazo abierto en el TNS.....	38
3.6	TNS utilizando la DCT-IV.....	39
3.7	Tres ventanas TDAC con diferente grado de traslape.....	39
4.1	Diagrama de bloques del códec ACC.....	42
4.2	Diagrama de flujo de la implementación del codificador.....	43
4.3	Implementación del banco de filtros MDCT.....	45
4.4	Implementación del banco de filtros IMDCT.....	46
4.5	Ventana Kaiser-Bessel Derivada.....	47

4.6	Funciones de propagación.....	54
4.7	Diagrama de flujo de la función II_ciclo_iteracion ().....	58
4.8	Diagrama de flujo de la implementación del ciclo de iteración externo.....	60
4.9	Diagrama de flujo de la implementación del ciclo de iteración interno.....	62
4.10	Estructura de un frame de audio estereofónico codificado.....	69
4.11	Estructura de datos complementarios.....	70
4.12	Reservorio de bits.....	74
4.13	Diagrama de flujo de la implementación del decodificador.....	76
5.1	Resultados de escucha generales.....	85
5.2	Comparación de SNR entre el ACC, el MP3 y el AAC-LC.....	86
5.3	Efectos de la compresión utilizando el ACC.....	87
5.4	Efecto del bloque TNS.....	89
D.1	Espectro MDCT de un bloque de 1024 muestras de audio.....	103
D.2	63 particiones del espectro MDCT.....	104
D.3	Espectro particionado de energía.....	104
D.4	Umbral de enmascaramiento.....	105
D.5	Umbral auditivo ATHmin.....	105
D.6	Umbral global de enmascaramiento.....	106
D.7	Umbral global de energía de enmascaramiento.....	106

Índice de tablas

4.1	Particiones del espectro MDCT utilizadas en el PAM.....	51
4.2	Bandas de factores de escala.....	55
4.3	Parámetros para convertir particiones a bandas de factores de escala.....	56
4.4	Par de tablas utilizadas para la codificación de la región count1.....	64
4.5	Ejemplo de tablas de código Huffman para la región bigvalues.....	64
4.6	Posibles “tablas”, para la región bigvalues.....	65
5.1	Escala de 5 grados de deterioro de audio ITU-R.....	82
5.2	Muestras de audio utilizadas en las pruebas de escucha.....	83
5.3	Resultados de escucha a 64 kbps.....	84
5.4	Resultados de escucha a 96 kbps.....	84
5.5	Resultados de escucha a 128 kbps.....	84
5.6	Resultados de escucha a 160 kbps.....	84
5.7	Resultados de escucha a 192 kbps.....	84
5.8	Resultados de escucha a 256 kbps.....	85
5.9	Resultados de escucha a 320 kbps.....	85
5.10	Comparación de SNR entre el ACC, el MP3 y el AAC-LC.....	86

Capítulo 1

Introducción

La investigación sobre codificadores perceptuales de audio comenzó hace poco más de 30 años. Estudios sobre el aparato auditivo del ser humano revelaron que la audición se basa, principalmente, en un análisis espectral de tiempo corto de las señales de audio. Se descubrió el llamado efecto de enmascaramiento, en el que el aparato auditivo humano no es capaz de percibir distorsiones que son enmascaradas por una señal más fuerte, en su entorno en el plano tiempo-frecuencia. Así, la idea básica de la codificación perceptual de audio, es modelar el ruido de cuantización en el dominio de la frecuencia, de tal forma que las distorsiones introducidas por cuantización, estén exactamente por debajo de un umbral de enmascaramiento. Esta investigación condujo al desarrollo de los códecs de audio perceptuales de hoy en día, con estándares definidos por el comité Moving Picture Experts Group (MPEG), tales como el MPEG-1 Layer-3 (o simplemente MP3) y el MPEG-4 Advanced Audio Coding-Low Complexity (AAC-LC). El desempeño de estos codificadores depende principalmente de dos aspectos: cuan precisamente puede ser aproximado el umbral de enmascaramiento y cuan eficientemente puede ser codificado el espectro cuantizado [1, 2, 3].

Justificación

En este trabajo se pretende entender y aplicar la teoría involucrada en la compresión perceptual de audio a bajas tasas de bits y, mejorar algunas soluciones que ya se encuentran en uso actualmente. En el códec objeto de presente trabajo, se utilizará como base la misma arquitectura fundamental empleada tanto en el MP3 como en el AAC-LC. Al explicar detalladamente la teoría y la implementación del códec propuesto, se pretende además, que el presente trabajo sirva de referencia, para aquellos alumnos de la FI de la UNAM, interesados en entender el algoritmo básico de compresión de audio, utilizado por el MP3 y el AAC-LC.

Objetivo

Diseñar e implementar un códec perceptual de audio digital a bajas tasas de bits. La implementación se lleva a cabo en el lenguaje C de programación. El códec implementado se aplica a varias señales de audio estereofónico en PCM lineal, con el fin de evaluar su desempeño.

Resultados esperados

- 1.- Obtener un codificador de audio estereofónico en PCM lineal, muestreado a 44.1kHz con 16 bits de cuantización, capaz de alcanzar una tasa de bits de 64 kbps.
- 2.- Obtener un decodificador capaz de decodificar un archivo de audio que previamente se haya codificado.
- 3.- Se pretende que el códec propuesto logre, desde un punto de vista perceptual, una calidad similar a la del disco compacto a los 128kbps.

Organización del trabajo

En el *capítulo 2* se presenta un panorama general sobre la codificación perceptual de audio digital. El *capítulo 3* describe a profundidad la transformada de coseno discreta modificada y como es utilizada para el diseño del códec perceptual de audio. El *capítulo 4* describe la implementación del códec perceptual propuesto. El *capítulo 5* muestra pruebas de sonido que miden la calidad subjetiva de los códecs MP3, AAC-LC y el códec implementado. Finalmente, se proporcionan las *conclusiones* y un *resumen* más concreto de los resultados obtenidos.

Capítulo 2

Compresión de audio digital

Las señales de audio de alta calidad sin compresión, por lo regular están muestreadas a 44.1kHz y codificadas con modulación por código de pulso (PCM) a 16 bits/muestra, esto da como resultado una alta tasa de bits de 1411kb/s para una señal estereofónica. En aplicaciones tales como la radio digital terrestre, es deseable tener una baja tasa de bits sin comprometer la calidad de la señal de audio.

La compresión de audio es una forma de compresión de datos, que consiste en la reducción del tamaño de los archivos de audio y existen dos tipos [3]: la compresión sin pérdida, que permite una reconstrucción exacta a costa de bajas razones de compresión, y la compresión con pérdida, que permite altas tasas de compresión a costa de sólo obtener una reconstrucción aproximada de la señal original.

A continuación se describirán los principales elementos que forman parte de los algoritmos de compresión con pérdida, pertenecientes a la categoría de codificación perceptual de audio, que para realizar la compresión aprovechan tanto irrelevancias perceptuales como redundancias estadísticas [3].

2.1 Arquitectura genérica de un codificador perceptual de audio

La mayoría de los compresores perceptuales de audio se basan en la arquitectura mostrada en la figura 2.1 [1, 2]. Por lo regular, estos codificadores segmentan las señales de entrada en bloques cuasi-estacionarios traslapados, cuya duración se encuentra entre los 3ms y los 45ms (entre 128 y 2048 muestras aprox., considerando una frecuencia de muestreo de 44.1kHz). Posteriormente, en la sección del banco de filtros de análisis se obtiene una representación espectral para cada bloque. Lo anterior se lleva a cabo con el fin de extraer del audio de entrada, un conjunto de parámetros en el dominio de la frecuencia que sea adecuado para su cuantización y codificación de acuerdo con una métrica de distorsión perceptual. La elección de un mapeo tiempo-frecuencia siempre involucra un compromiso entre los requerimientos de resolución en tiempo y frecuencia [4].

En el bloque del modelo psicoacústico (PAM) se obtienen umbrales de enmascaramiento, que cuantifican la máxima cantidad de distorsión permisible en cada punto en el plano tiempo-frecuencia, de tal forma que la cuantización de los parámetros espectrales no introduzca artefactos audibles. De esta forma, el modelo psicoacústico permite que el proceso de cuantización aproveche las irrelevancias perceptuales en el conjunto de parámetros espectrales. La cuantización puede ser uniforme o no uniforme, y puede ser llevada a cabo de forma escalar o de forma vectorial (VQ). Una vez que un conjunto compacto de parámetros cuantizados ha sido formado, las redundancias restantes por lo regular son removidas mediante técnicas de codificación entrópica [2, 3]. En la etapa de construcción del bitstream se empaquetan los datos que, por lo general, consisten de los coeficientes espectrales cuantizados y codificados, y de información complementaria necesaria para la correcta decodificación de los datos. El decodificador lleva a cabo los pasos inversos.

Sin importar los detalles de diseño, todos los códecs perceptuales de audio buscan lograr una calidad transparente (perceptualmente igual al original) a bajas tasas de bits, con una complejidad manejable y retardo aceptable.

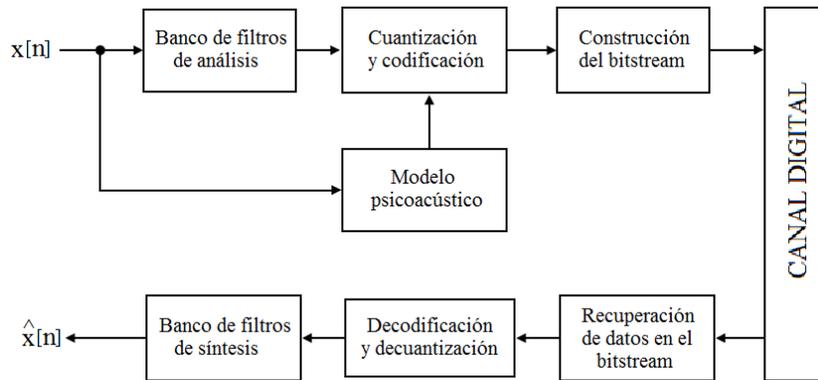


Figura 2.1. Códec perceptual genérico de audio.

2.2. Psicoacústica

La Psicoacústica es el estudio científico de la percepción del sonido y, gracias a ésta, se ha descubierto que el sistema auditivo humano posee características que pueden ser aprovechadas en la codificación perceptual de audio. Algunas de estas características se deben a limitantes en el propio oído, y otras son consecuencia de la forma de procesar el sonido en el cerebro. A continuación se procede a describir los principales aspectos de la psicoacústica que son útiles en el diseño de un codificador perceptual.

2.2.1. Niveles de presión de sonido

Los valores de presión de sonido relevantes para las aplicaciones de audio van desde 1×10^{-5} Pa hasta 1×10^2 Pa. El primer valor corresponde aproximadamente a los límites de la audición humana en las frecuencias más sensibles, y el segundo valor corresponde al umbral del dolor [5]. Para manejar este intervalo tan amplio de valores, se utiliza una escala logarítmica y se define el nivel de presión de sonido (SPL):

$$SLP = 10 \log_{10} \left(\frac{p}{p_0} \right)^2 = 10 \log_{10} \left(\frac{I}{I_0} \right) \quad [\text{dB}] \quad (2.1)$$

donde $p_0 = 20\mu\text{Pa}$ e $I_0 = 1 \times 10^{-12} \text{ W/m}^2$ son aproximadamente iguales, respectivamente, a la presión e intensidad de sonido, en el umbral de audición para tonos alrededor de 2kHz [5].

Por ejemplo, para determinar el SPL de una senoide (o tono), es común considerar como referencia, una senoide con una amplitud igual a medio paso de cuantización PCM ($A_0 = \Delta_{PCM}/2$) [6, 7]. Además, para palabras de 16 bits, una senoide con amplitud igual al nivel de sobrecarga del cuantizador ($A_{max} = (2^{16} - 1) \Delta_{PCM}/2$), tendrá un SPL de aproximadamente 96 dB (esto es, $20\log_{10}(A_{max}/A_0) \approx 96 \text{ dB}$). De esta forma, si se considera para el cuantizador un nivel de sobrecarga igual a 1, entonces el SPL de un tono con amplitud A , estará dado por [6]:

$$SPL_{Tono} = 96\text{dB} + 10\log_{10}(A^2) \quad (2.2)$$

2.2.2. Umbral auditivo

En el intervalo de frecuencias audibles, que va de 20Hz a 20kHz, el umbral auditivo caracteriza la cantidad mínima de energía necesaria en un tono puro para poder ser percibido. De acuerdo con [6], es posible obtener una buena aproximación del umbral auditivo utilizando la ecuación:

$$T_q(f) = 3.64 \left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000} - 3.3\right)^2} + 10^{-3} \left(\frac{f}{1000}\right)^4 \quad [\text{dB SPL}] \quad (2.3)$$

Que es representativa de un sujeto joven con un muy buen oído, ver figura 2.2. Es importante considerar que para el diseño de un códec no se tiene conocimiento a priori sobre el nivel real de reproducción y, por lo tanto, resulta congruente considerar que 0 dB en el umbral auditivo, corresponde a la energía en una senoide de amplitud de medio paso de cuantización PCM.

2.2.3. Enmascaramiento simultaneo

Enmascaramiento se refiere a un proceso donde un sonido se vuelve inaudible a causa de la presencia de otro sonido. El enmascaramiento simultáneo es un fenómeno en el dominio de la frecuencia que puede observarse cuando dos o más estímulos se presentan de forma simultánea en el sistema auditivo. Dependiendo de la forma de sus espectros de magnitud, la presencia de cierta energía espectral enmascarará la presencia de otra energía espectral. En la figura 2.3, además del umbral auditivo, se muestra una curva llamada umbral de enmascaramiento y representa el umbral de audibilidad de señales, en presencia de una señal enmascaradora. Entonces, otras señales o componentes frecuenciales que se encuentren por debajo esta curva, no serán percibidos cuando el enmascarador esté presente.

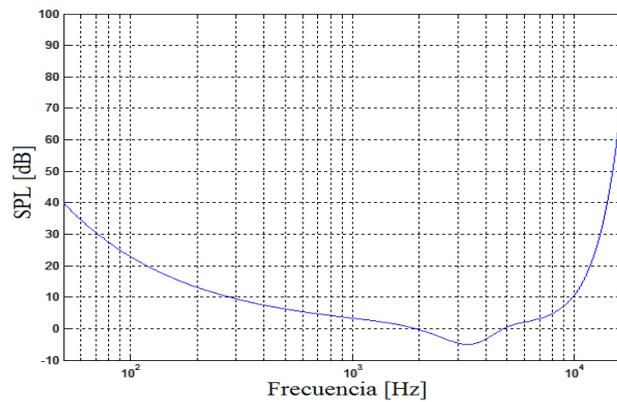


Figura 2.2. Umbral auditivo según la ecuación (2.3)

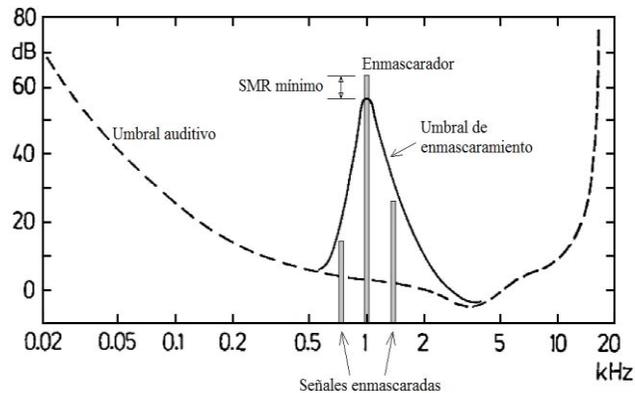


Figura 2.3. Ejemplo de enmascaramiento simultaneo.

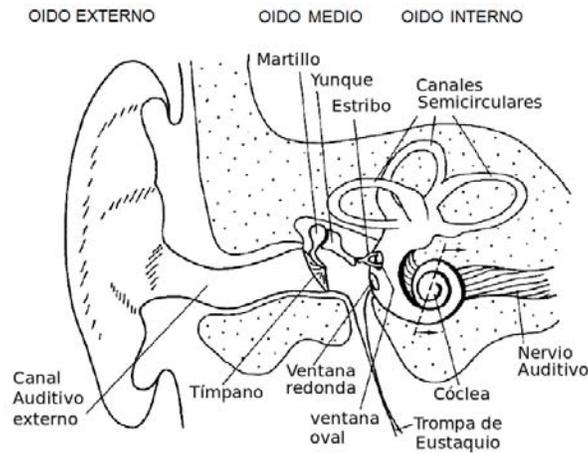
Para efectos de darle forma a las distorsiones por cuantización es suficiente considerar sólo dos tipos de señales enmascaradoras: las tipo tono y las tipo ruido de banda angosta [6]. En el primer caso, el enmascarador consiste de un solo tono, mientras que en el segundo caso, el enmascarador consiste de ruido cuyo ancho de banda no supera el llamado ancho de banda crítico (definido en la sección 2.2.4).

La diferencia de nivel entre un enmascarador y su umbral de enmascaramiento se conoce como razón señal a máscara (SMR). En la figura 2.3 se muestra el valor mínimo de SMR entre un enmascarador y el umbral de enmascaramiento que genera. El SMR mínimo es un parámetro muy importante en el diseño de códecs de audio y en general cuando el enmascarador es tonal, el SMR mínimo es mayor que cuando el enmascarador es ruido de banda angosta. Nótese que mayores niveles de SMR implican un menor nivel de enmascaramiento.

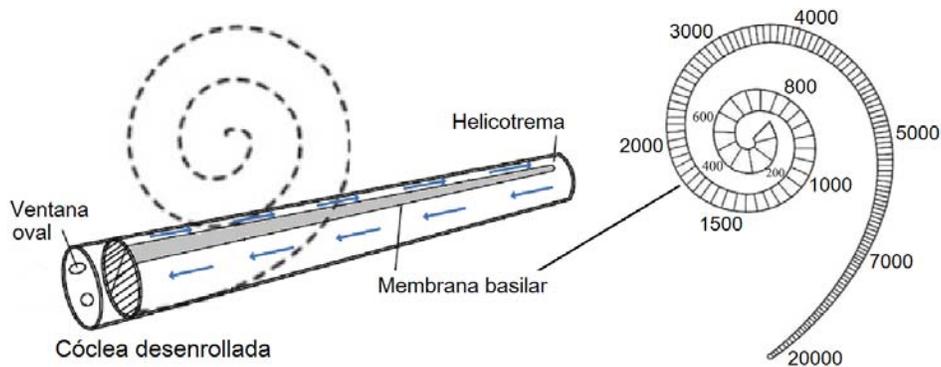
2.2.4. Bandas críticas y la escala Bark

Utilizar el umbral auditivo para dar forma al espectro de distorsión por cuantización, representa un primer paso en la codificación perceptual. Sin embargo, también es necesario considerar un umbral de enmascaramiento para el ruido de cuantización, cuya forma tome en cuenta los estímulos presentes en cualquier momento dado. Como en general los estímulos son variantes en el tiempo, el umbral de enmascaramiento también lo es. Con el fin de estimar este umbral, es necesario tener una idea de cómo el oído lleva a cabo el análisis espectral.

El oído puede dividirse en tres regiones principales: *el oído externo*, *el oído medio* y *el oído interno*, como se muestra en la figura 2.4a. El papel principal del *oído externo* es captar el sonido y canalizarlo por el canal auditivo externo hacia el oído medio a través del tímpano. El *oído medio* transforma la onda de presión que incide en el tímpano, en movimientos de fluido dentro de la cóclea en el *oído interno*. Posteriormente la cóclea traduce estos movimientos de fluido en señales eléctricas que son conducidas al cerebro a través del nervio auditivo.



(a)



(b)

Figura 2.4. Aparato auditivo: (a) Esquema del oído externo, medio e interno [6]. (b) Distribución de frecuencias resonantes a lo largo de la membrana basilar [7].

Se puede considerar que dentro de la cóclea, a lo largo de la membrana basilar, se lleva a cabo un proceso de transformación de frecuencia a posición. Esto debido a que distintas regiones de la membrana basilar están “entonadas” para diferentes bandas de frecuencia [7], como se muestra en la figura 2.4b. Al medir experimentalmente los umbrales de enmascaramiento, se descubrió que existe un intervalo de frecuencias angosto alrededor de la frecuencia de un enmascarador, dónde el umbral de enmascaramiento es aproximadamente plano en vez de decreciente. Este intervalo se conoce como ancho de banda crítico. De hecho, la cóclea se puede considerar como un banco de filtros paso-banda densamente traslapados [6]. Las respuestas de estos filtros (proporcionales a los umbrales

de enmascaramiento) son asimétricas y sus bandas de paso tienen un ancho de banda no uniforme que aumenta al aumentar la frecuencia, como se muestra en la figura 2.5a. No obstante, si se considera el banco de filtros anterior en función de la distancia a lo largo de la membrana basilar dentro de la cóclea (desde la helicotrema hasta la ventana oval), entonces la forma de los filtros cocleares es bastante independiente de la frecuencia del enmascarador y los anchos de banda críticos son iguales, ver figura 2.5b. La distancia correspondiente a una banda crítica se conoce como “un bark” y 25 veces esa distancia corresponde a la longitud total de la membrana basilar [6]. La función [5]

$$z(f) = 13 \arctan(0.00076f) + 3.5 \arctan \left[\left(\frac{f}{7500} \right)^2 \right], \quad (2.4)$$

cuya gráfica se muestra en la figura 2.6, es bastante adecuada para convertir frecuencia en Hertz a la escala Bark.

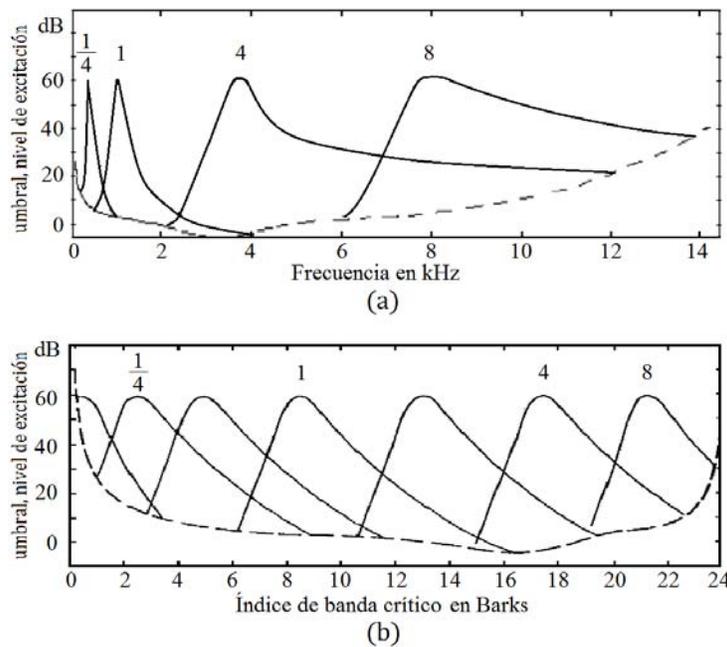


Figura 2.5. Patrones de excitación para señales de ruido de banda angosta, con un SPL de 60dB, centradas en las frecuencias 0.25, 1, 4 y 8 kHz. a) Escala lineal de frecuencia, b) Escala Bark [5].

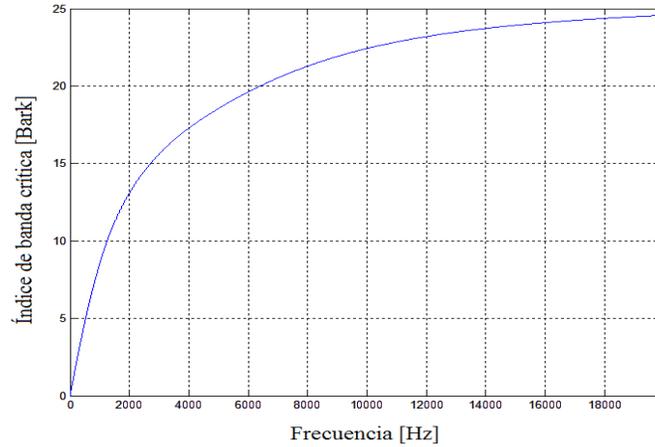


Figura 2.6. Índice de banda crítica según la ecuación (2.4)

Además, como se ilustra en la figura 2.7, cada filtro coclear es no lineal, en el sentido de que su magnitud depende del SPL del enmascarador a una frecuencia dada.

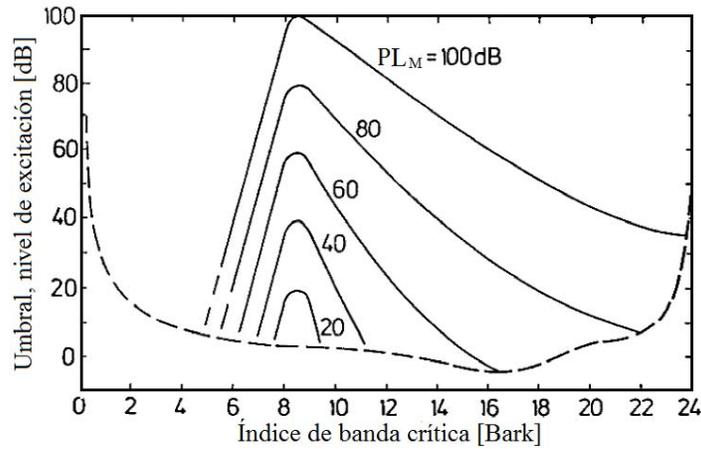


Figura 2.7. Patrones de excitación para señales de ruido de banda angosta, centradas en 1 kHz y a diferentes niveles [5].

2.2.5 Curvas de enmascaramiento

De lo expuesto se puede concluir que, modelar los umbrales de enmascaramiento resulta más sencillo si se utiliza la escala Bark, en lugar de la escala de frecuencias en Hertz.

Como lo sugiere la figura 2.3, con el fin de representar analíticamente el umbral de enmascaramiento para un enmascarador en la frecuencia $z_j = z(f_j)$, se puede utilizar la ecuación:

$$T_M(z, z_j) = PL_M(z_j) + 10\log_{10} \left(SF(z - z_j) \right) - \Delta_M(z_j) \quad [\text{dB}] \quad (2.5)$$

donde $PL_M(z_j)$ es el nivel de presión de sonido del enmascarador; $\Delta_M(z_j)$ es el mínimo SMR del enmascarador y su valor depende, principalmente, de si el enmascarador es tipo ruido o tipo tono [6]; $SF(z - z_j)$ es la *función de propagación* y modela la forma de la curva del umbral de enmascaramiento.

Un ejemplo de función de propagación, es la definida por el modelo psicoacústico 2 del estándar ISO/IEC MPEG 1 capa 3 [6], ver figura 2.8:

$$10\log_{10}SF(dz) = 15.81 + 7.5(\beta dz + 0.474) - 17.5\sqrt{1 + (\beta dz + 0.474)^2} + 8 \times \text{MIN}(0, (\beta dz - 0.5)^2 - 2(\beta dz - 0.5)) \quad (2.6)$$

donde $dz = z - z_{\text{enmascarador}}$.

$$\beta = 3 \text{ con } dz \geq 0 \quad \text{o} \quad \beta = 1.5 \text{ con } dz < 0$$

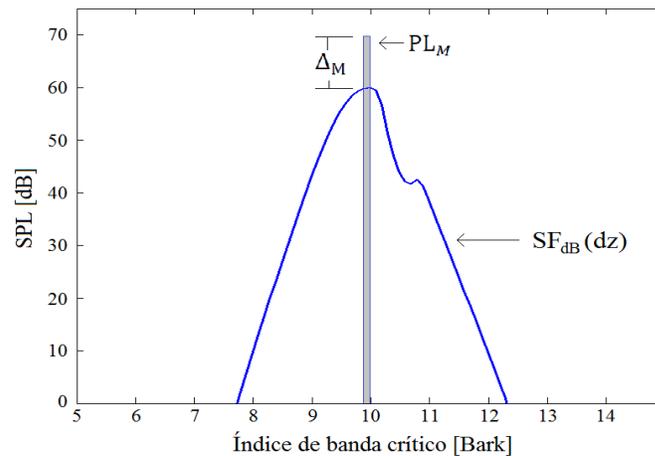


Figura 2.8. Curva de enmascaramiento según la ecuación (2.5), considerando la función de propagación definida por la ecuación (2.6).

2.2.6 Umbral global de enmascaramiento

Por lo regular, las señales de audio contienen una gran cantidad de componentes tipo tono y tipo ruido. Entonces, después de que los tipos de componentes son identificados en un intervalo dado de tiempo, se procede a calcular sus curvas individuales de enmascaramiento. Para modelar el efecto total de enmascaramiento que se produce, es necesario combinar de forma “adecuada”, las curvas individuales de enmascaramiento previamente obtenidas. El método utilizado para combinar las curvas individuales de enmascaramiento en los modelos psicoacústicos 1 y 2 del MPEG 1 audio, consiste simplemente en sumar las intensidades individuales de las curvas. Otro método para combinar las curvas individuales de enmascaramiento consiste en conservar, para cada frecuencia, el valor máximo de todas las curvas. Este método se utiliza en el códec Dolby AC-2A [6].

Después de que los umbrales individuales han sido combinados, también se debe tomar en consideración los efectos del umbral auditivo, para lo cual se crea el llamado *umbral global de enmascaramiento*. Un método para obtener este umbral global, consiste en tomar el máximo valor entre el umbral auditivo y la combinación de los umbrales individuales. A manera de ejemplo, en la figura 2.9b se muestra el umbral global de enmascaramiento obtenido mediante la ecuación (2.7), para las diez curvas de enmascaramiento mostradas en la figura 2.9a.

$$T_g(z) = \text{MAX} \left(T_q(z), 10 \log_{10} \left(\sum_{j=1}^{10} 10^{0.1 T_M(z, z_j)} \right) \right) \quad [\text{dB}] \quad (2.7)$$

donde $T_q(z)$ es el umbral auditivo en la escala Bark, y los $T_M(z, z_j)$ son las curvas individuales de enmascaramiento definidas por la ecuación (2.5).

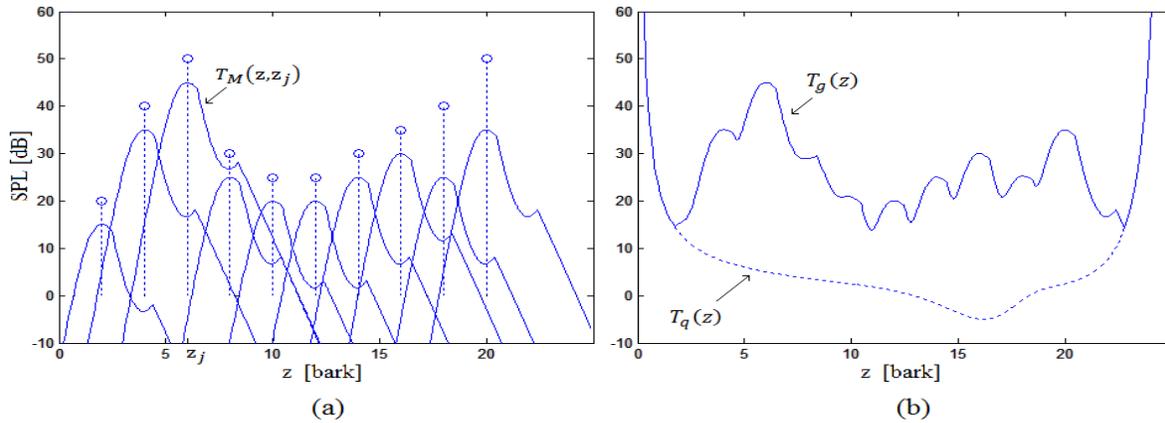


Figura 2.9. Umbral global de enmascaramiento.

2.2.7 Enmascaramiento temporal

El fenómeno de enmascaramiento también se presenta en el dominio del tiempo, como se muestra en la figura 2.10. Dado un enmascarador de duración finita, este tipo de enmascaramiento no simultáneo, ocurre antes del comienzo y después de la desaparición del enmascarador. En el caso de señales de audio, los transitorios abruptos crean regiones de pre y post enmascaramiento temporales, durante las cuales no se percibirán señales por debajo de la elevación en los umbrales de audición producida por el enmascarador. El pre-enmascaramiento dura aproximadamente 20ms, aunque tiende a ser efectivo solo durante 2ms aproximadamente. Por otro lado, la duración del post-enmascaramiento varía entre los 50ms y los 300ms dependiendo de la intensidad y duración del enmascarador [5].

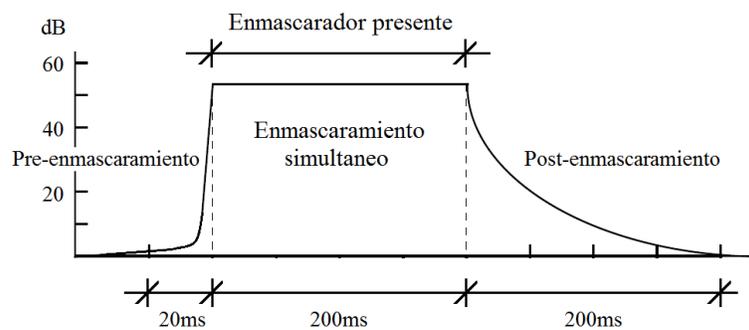


Figura 2.10. Propiedades de emascaramiento temporal del sistema auditivo.

Los codificadores perceptuales utilizan una descomposición espectral por bloques de la señal de entrada, para la cuantización y codificación. Esto implica que el error de cuantización, introducido en el dominio de la frecuencia, se extiende en el dominio del tiempo a lo largo del bloque en análisis (Principio de incertidumbre tiempo/frecuencia) [8]. Esto produce un artefacto conocido como pre-eco, cuando la señal por codificar contiene fuertes componentes, solo en algunas partes del bloque en análisis. Un ejemplo de pre-eco se muestra en la figura 2.11.

Debido a las propiedades del sistema auditivo los “pre-ecos” son enmascarados, solo si no existe ninguna cantidad significativa de ruido de cuantización, más allá de 2 ms antes del comienzo de la señal. Esto por lo regular implica que la etapa de análisis tiempo/frecuencia de un códec, disponga de una alta resolución en el dominio del tiempo.

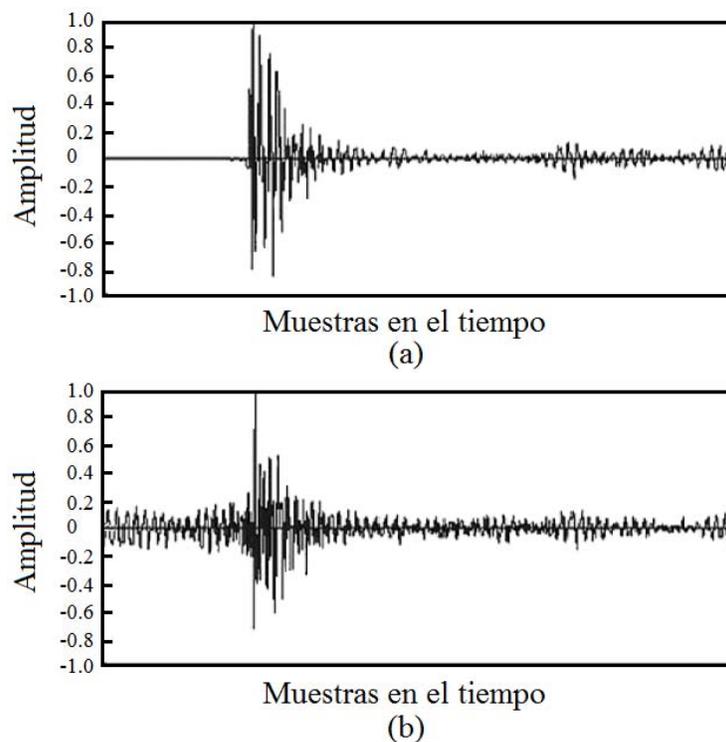


Figura 2.11. Pre-eco. (a) Señal original. (b) Efecto sobre la señal en el dominio del tiempo, debido a la cuantización de su espectro DFT [9].

2.3. Banco de filtros uniforme máximamente decimado de M bandas

Los códecs de audio poseen una etapa de análisis tiempo-frecuencia que extrae de la señal de entrada, un conjunto de parámetros adecuado para la cuantización y codificación, de acuerdo con una métrica de distorsión perceptual. La herramienta comúnmente empleada para este análisis es un banco de filtros paso-banda, que descompone la señal de entrada en sus componentes frecuenciales, lo cual hace posible el análisis psicoacústico, el modelado perceptual del ruido de cuantización, y la reducción de redundancias estadísticas [6]. A continuación se describe el banco de filtros uniforme máximamente decimado de M bandas, y dos casos particulares ampliamente utilizados en la codificación perceptual de audio [1, 6].

2.3.1. Estructura del banco de filtros

La figura 2.12 muestra un banco de filtros uniforme máximamente decimado de M canales [10]. Los filtros de análisis $H_0(z)$, $H_1(z)$, ..., $H_{M-1}(z)$ dividen la señal $x[n]$ en M bandas de frecuencia, y cada señal de subbanda es decimada por un factor de M para obtener las secuencias $X_k[r]$. Lo anterior implica que el sistema es críticamente muestreado. Para la compresión, cada una de las señales $X_k[r]$ puede ser cuantizada con un número limitado de bits, colocando la mayor parte del ruido de cuantización en bandas de frecuencia donde sea perceptualmente inaudible, obteniendo así las secuencias $\hat{X}_k[r]$. En el decodificador, las señales $\hat{X}_k[r]$ son decuantizadas, se insertan M-1 ceros entre sus muestras, y luego son filtradas por los filtros de síntesis $F_0(z)$, $F_1(z)$, ..., $F_{M-1}(z)$, para obtener las secuencias $v_k[n]$. Por último, al sumar las M secuencias $v_k[n]$ se obtiene la señal $\hat{x}[n]$, que es una versión reconstruida de $x[n]$ y se desea que, perceptualmente hablando, sea igual a $x[n]$.

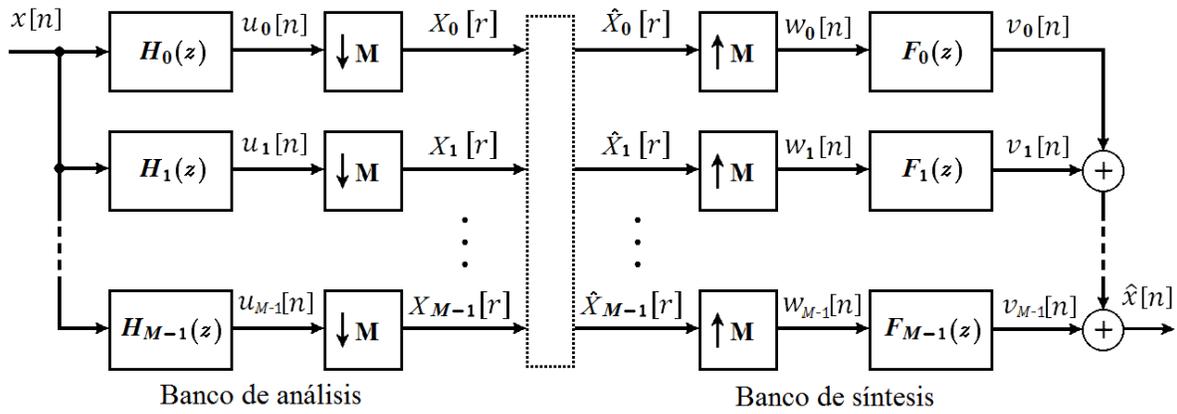


Figura 2.12. Banco de filtros maximamente decimado de M bandas.

En ausencia de cuantización, esto es, cuando $\hat{X}_k[r] = X_k[r]$, se dice que el banco de filtros de la figura 2.12 tiene la propiedad de reconstrucción perfecta (PR), si $\hat{x}[n] = x[n - n_0]$. Sin embargo, cuando el banco de filtros no posee la propiedad de reconstrucción perfecta, la señal $\hat{x}[n]$ sufre de tres distorsiones [10]: distorsión de amplitud, distorsión de fase y, distorsión por aliasing. Esta última, debida al proceso de decimación/interpolación en cada canal, ver figura 2.13.

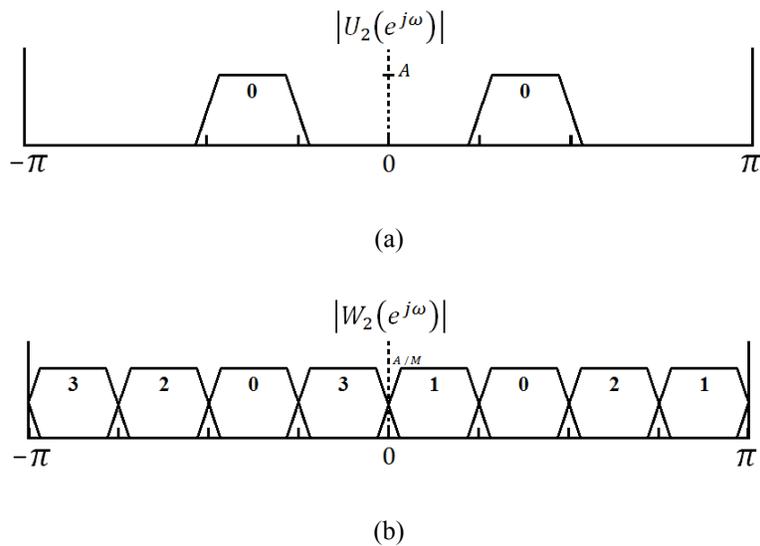


Figura 2.13. Representación en el dominio de la frecuencia, del proceso de submuestreo para la 2da. banda de un banco de filtros de $M = 4$ bandas. a) Espectro de salida del 2do. filtro de análisis. b) Espectro de la secuencia $w_2[n]$.

2.3.2. Banco de filtros FIR como transformada de bloque

Sí en el banco de filtros de la figura 2.12, se consideran filtros FIR de longitud $L = QM$, tales que satisfagan la ecuación:

$$f_k[m] = h_k[(L - 1) - m] \quad (2.8)$$

Y sí, a partir de $x[n]$ y de las M señales de subbanda $X_k[r]$, se definen respectivamente, los vectores:

$$\mathbf{x} = [\dots, x[-4], x[-3], x[-2], x[-1], x[0], x[1], x[2], x[3], x[4], \dots]^T \quad (2.9)$$

$$\mathbf{X} = [\dots, X_0[0], \dots, X_{M-1}[0], X_0[1], \dots, X_{M-1}[1], X_0[2], \dots, X_{M-1}[2], \dots]^T \quad (2.10)$$

Entonces, como se demuestra en el apéndice B, la operación del banco de análisis en la figura 2.12, puede ser descrita mediante la ecuación

$$\mathbf{X} = \mathbf{T}^T \mathbf{x} \quad (2.11)$$

Y, la operación del banco de síntesis en la figura 2.12, puede ser descrita mediante la ecuación

$$\hat{\mathbf{x}} = \mathbf{T} \mathbf{X} \quad (2.12)$$

donde

$$\mathbf{T} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots \\ \ddots & P_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots \\ \ddots & P_1 & P_0 & \mathbf{0} & \mathbf{0} & \ddots \\ \ddots & \vdots & P_1 & P_0 & \mathbf{0} & \ddots \\ \ddots & P_{Q-1} & \vdots & P_1 & P_0 & \ddots \\ \ddots & \mathbf{0} & P_{Q-1} & \vdots & P_1 & \ddots \\ \ddots & \mathbf{0} & \mathbf{0} & P_{Q-1} & \vdots & \ddots \\ \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & P_{Q-1} & \ddots \\ \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (2.13)$$

Además, si se combinan las ecuaciones (2.11) y (2.12) se obtiene

$$\hat{\mathbf{x}} = \mathbf{T}\mathbf{T}^T\mathbf{x} \quad (2.16)$$

La ecuación anterior implica que el banco de filtros de la figura 2.12, tendrá la propiedad de reconstrucción perfecta si la matriz \mathbf{T} representa un operador ortogonal, esto es, si:

$$\mathbf{T}^T = \mathbf{T}^{-1} \quad (2.17)$$

De acuerdo con Malvar en [12], la restricción anterior es equivalente a

$$\mathbf{P}^T\mathbf{W}^m\mathbf{P} = \delta[m]\mathbf{I}_M, \quad m = 0, 1, \dots, Q - 1 \quad (2.18)$$

donde \mathbf{W} es la matriz de desplazamiento de un bloque, definida por

$$\mathbf{W} = \begin{bmatrix} \mathbf{0}_{(QM/2) \times M} & \mathbf{I}_{(Q-1)M} \\ \mathbf{0}_{(QM/2) \times M} & \mathbf{0}_{M \times (Q-1)M} \end{bmatrix}_{QM \times QM} \quad (2.19)$$

Las restricciones en (2.18), son las condiciones necesarias y suficientes para que una matriz de $QM \times M$ defina una transformada Lapped (LT), [12].

2.3.3. Banco de filtros PQMF

La modulación por coseno de un filtro prototipo paso-bajas FIR ha sido utilizada desde principios de la década de los 80s [13], para implementar bancos de filtros máximamente decimados de M bandas con cuasi reconstrucción perfecta, conocidos como bancos pseudo quadrature mirror filter (PQMF). Los filtros de análisis y síntesis en un banco PQMF están dados por [6]:

$$h_k[n] = h[n] \cos \left[\frac{2\pi}{4M} (2k + 1) \left(n + \frac{1}{2} - \frac{L}{2} \right) + \varphi_k \right], \quad k = 0, \dots, M - 1 \quad (2.20)$$

$$f_k[n] = h_k[(L - 1) - n]$$

donde L es la longitud de la secuencia $h[n]$, un filtro prototipo FIR paso-bajas de fase lineal con frecuencia de corte normalizada de $\pi/2M$. La fase φ_k es tal que el aliasing entre bandas adyacentes prácticamente se cancela y satisface la relación [6]:

$$\varphi_k - \varphi_{k-1} = \frac{\pi}{2}(2l + 1)$$

Nótese que en un banco de filtros PQMF, la distorsión de fase es completamente eliminada de la función de transferencia global, al forzar que los filtros de análisis y síntesis satisfagan la ecuación (2.8). Dado que las distorsiones por aliasing y de fase han sido eliminadas, el procedimiento de diseño del banco de filtros se reduce al diseño de la ventana $h[n]$, de tal forma que la distorsión global de amplitud se minimice [6].

El banco PQMF es una pieza clave en varios códecs perceptuales de audio. Por ejemplo, los códecs de audio MPEG-1/2 en sus capas I, II y III (en esta última, como parte de su banco de filtros híbrido) utilizan un banco PQMF de 32 bandas. En este caso el filtro prototipo, $h[n]$, tiene una longitud de $L = 512$, lo que implica una atenuación de más de 96-dB en la banda de rechazo de cada filtro de análisis. El rizo de salida (distorsión AMD) del banco PQMF es menor de 0.07 dB [6].

2.3.4. Extended Lapped Transform (ELT)

Los bancos de filtros PQMF requieren compensar la distorsión que implica su falta de reconstrucción perfecta, con el fin de evitar artefactos audibles. Por esta razón es preferible el uso de bancos de filtros uniformes máximamente decimados con PR, los cuales limitan las fuentes de distorsión a la etapa de cuantización.

A principios de los años noventa, Malvar [14, 15] demostró la existencia de bancos de filtros modulados por coseno, que poseen la propiedad de reconstrucción perfecta. Esta clase de bancos es, de hecho, un tipo de transformada Lapped que Malvar en [15], denomina Extended Lapped Transform (ELT).

De acuerdo con Malvar [15], las funciones base de la ELT están definidas por la ecuación:

$$p_{nk} = f_k[n] = h[n] \sqrt{\frac{2}{M}} \cos \left[\frac{2\pi}{4M} (2k+1) \left(n + \frac{1}{2} + \frac{M}{2} \right) \right] \quad (2.21)$$

para $k = 0, 1, \dots, M-1$, $n = 0, 1, \dots, L-1$ y $L = 2KM$. Con el fin de obtener la propiedad PR con la ELT, se debe escoger una secuencia $h[n]$ tal que la matriz correspondiente \mathbf{P} , satisfaga las condiciones de ortogonalidad en (2.18). Como lo demuestra Malvar en [14, 15], lo anterior se cumple sí la secuencia $h[n]$ tiene simetría par

$$h[L-1-n] = h[n] \quad (2.22)$$

y satisface el siguiente sistema de ecuaciones no lineales:

$$\sum_{i=0}^{2K-2m-1} h[n+iM]h[n+iM+2mM] = \delta[m], \quad (2.23)$$

para $m = 0, 1, \dots, K-1$ y $n = 0, 1, \dots, \frac{M}{2}-1$

En el presente trabajo se utilizará para el mapeo tiempo-frecuencia la MDCT, un caso particular de la ELT con $L=2M$. Esto debido a que la MDCT se ha convertido en una pieza clave en los algoritmos actuales de codificación perceptual de audio (e.g. AAC-LC).

Resumen

En este capítulo se presentó la estructura genérica de un códec perceptual de audio. Así mismo, se describieron los principales aspectos de la Psicoacústica y se presentó un análisis detallado del banco de filtros uniforme máximamente decimado, elemento fundamental en la mayoría de los compresores perceptuales de audio. De esta clase de bancos, se presentaron dos casos particulares: el banco de filtros PQMF, que ha sido de gran importancia para el desarrollo de la compresión perceptual de audio; y la transformada ELT, de la cual un caso particular, la MDCT, es utilizado para el mapeo tiempo-frecuencia en el códec desarrollado en el presente trabajo.

Capítulo 3

Compresor perceptual de audio

En este capítulo se describen los elementos principales que forman parte del compresor de audio objeto del presente trabajo. En primer lugar, se presentan la MDCT e IMDCT y se describe su propiedad de reconstrucción perfecta, basada en la cancelación de aliasing en el del tiempo (TDAC). Después, se presenta la relación existente entre la transformada discreta de coseno tipo IV (DCT-IV) y la MDCT e IMDCT. Así mismo, se explica cómo, gracias a esta relación, es posible calcular la MDCT directa e ~~inversa~~ inversa mediante un algoritmo FFT. Posteriormente, se procede a describir aspectos relevantes del modelo psicoacústico (PAM), el cual se basa en los coeficientes espectrales obtenidos mediante la MDCT. Luego, se presenta el esquema de cuantización a utilizar, que se aplica sobre los coeficientes de la MDCT. Éste esquema adapta su precisión de cuantización, de acuerdo a las condiciones de enmascaramiento de ruido, estimadas por el PAM. Por último, en este capítulo se presentan el reservorio de bits y el TNS, los cuales son métodos utilizados para la reducción de los artefactos de pre-eco.

3.1. Transformada de coseno discreta modificada

Como se mencionó en el capítulo anterior, la MDCT es un caso particular de la transformada ELT, con $L = 2M$. Entonces, a partir de las ecuaciones (2.21) y (B.3), se tiene que la transformada MDCT de un bloque $x_b[m]$, de longitud $2M$, está dada por:

$$X_b[k] = \sqrt{\frac{2}{M}} \sum_{m=0}^{2M-1} x_b[m] h[m] \cos \left[\frac{2\pi}{4M} (2k+1) \left(m + \frac{1}{2} + \frac{M}{2} \right) \right] \quad (3.1)$$

para $k = 0, 1, \dots, M-1$. Y, de acuerdo con la ecuación (B.9), la transformada inversa (IMDCT), está dada por:

$$y_b[m] = h[m] \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} X_b[k] \cos \left[\frac{2\pi}{4M} (2k+1) \left(m + \frac{1}{2} + \frac{M}{2} \right) \right] \quad (3.2)$$

para $m = 0, 1, \dots, 2M-1$.

Vale la pena mencionar que las condiciones PR para $h[n]$ en este caso especial de ELT, ecuaciones (2.22) y (2.23) con $K=1$, fueron primero demostradas por Princen y Bradley en [16, 17], cuando utilizaban argumentos en el dominio del tiempo, para el desarrollo de un banco de filtros con cancelación de aliasing en el dominio del tiempo (TDAC)".

En la figura 3.1 se muestra la interpretación de la MDCT, como una transformada por bloques traslapados, considerando $M = 6$ y el filtro prototipo o ventana TDAC [18]

$$h[m] = \text{sen} \left[\frac{2\pi}{4M} \left(m + \frac{1}{2} \right) \right], \quad m = 0, 1, \dots, 2M-1 \quad (3.3)$$

En primer lugar, la señal $x[n]$ en la etapa de análisis, es procesada por bloques de longitud $L = 2M$ con un traslape de un 50%. Cada bloque se transforma mediante la MDCT (ecuación 3.1), en un conjunto de M coeficientes en el dominio de la frecuencia. En la etapa de síntesis, cada conjunto de M coeficientes es transformado mediante la IMDCT

(ecuación 3.2), en una señal temporal de longitud $2M$, que se traslapa un 50% con los bloques adyacentes. Finalmente, con todos los bloques obtenidos se calcula la secuencia $\hat{x}[n] = x[n - (2M - 1)]$, mediante un método de suma con traslape (OLA).

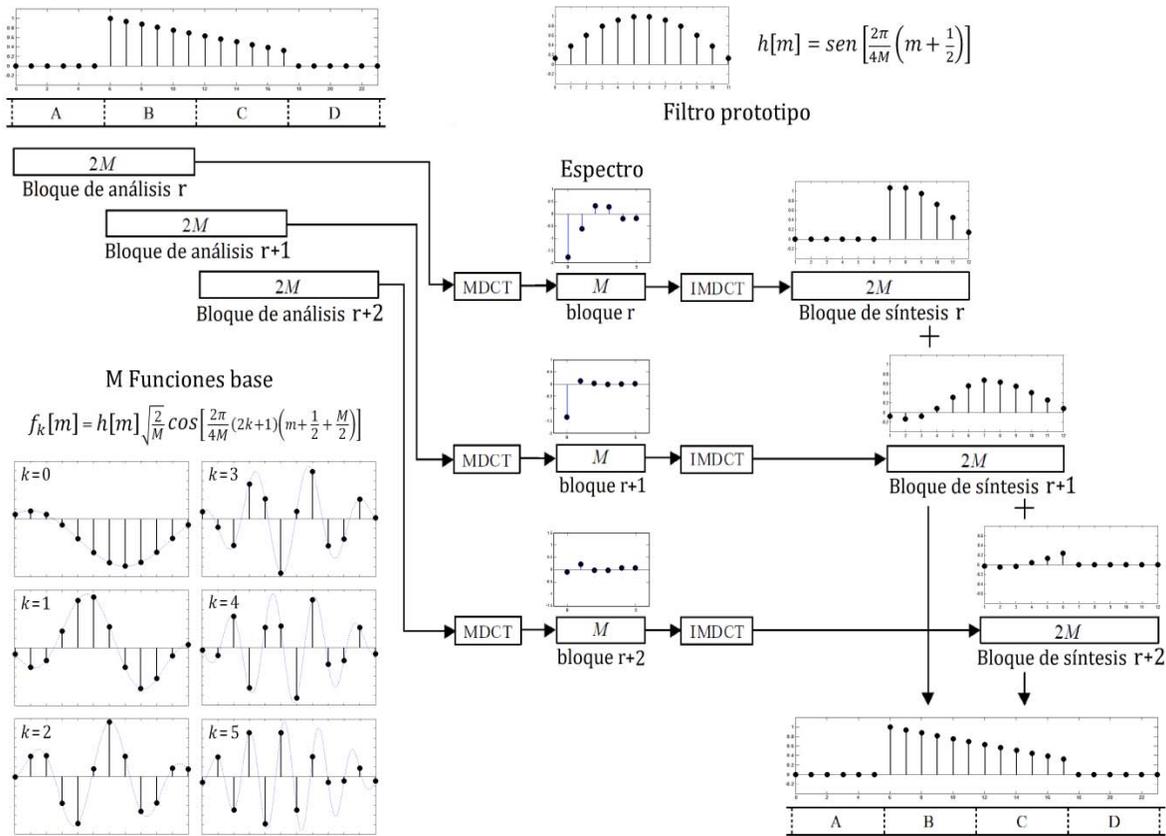


Figura 3.1. Transformada de coseno discreta modificada (MDCT).

Para visualizar cómo la MDCT descompone un bloque de audio en “tonos”, de acuerdo con la figura 3.1, se puede seguir el siguiente razonamiento. Cada bloque de audio, de longitud $2M$, sometido al proceso MDCT/IMDCT, es “transformado” en un bloque de audio de la misma longitud, pero con una distorsión conocida como aliasing en el dominio del tiempo (TDA) [16, 17]. Este bloque, de acuerdo con la ecuación (3.2), es una combinación lineal de M funciones base o “tonos” (ver figura 3.1), donde los escalares o amplitudes de los “tonos”, son los coeficientes espectrales que arroja la MDCT para cada bloque de $2M$ muestras. La distorsión que supone el proceso MDCT/IMDCT, se cancela al ir sumado todos los bloques con un método OLA.

3.1.1. MDCT/IMDCT vía la DCT-IV y el origen del TDAC

Con el fin de describir el fenómeno de aliasing en el dominio del tiempo (TDA), resulta útil considerar la relación existente entre la MDCT y la transformada discreta de coseno tipo IV (DCT-IV) [21]. Además, esta relación da pie a varios algoritmos rápidos para el cálculo de la MDCT e IMDCT, vía la DCT-IV [22-25]. Un algoritmo rápido para esta última, se considerará en el punto 3.1.2.

Suponiendo una MDCT de M canales, con M par, de acuerdo con [12, 19], las ecuaciones (3.1) y (3.2) pueden ser factorizadas en forma matricial, respectivamente, como:

$$\mathbf{X}_b = \mathbf{C}_M^{IV} \mathbf{M}_{TDA} [\mathbf{x}_b \circ \mathbf{h}] \quad (3.4)$$

$$\mathbf{y}_b = \mathbf{h} \circ [\mathbf{M}_{TDA}^T \mathbf{C}_M^{IV} \mathbf{X}_b] \quad (3.5)$$

Donde

- Denota el producto Hadamard [20]

$$\mathbf{x}_b = (x_b[0], x_b[1], \dots, x_b[2M-1])^T$$

$$\mathbf{y}_b = (y_b[0], y_b[1], \dots, y_b[2M-1])^T$$

$$\mathbf{h} = (h[0], h[1], \dots, h[2M-1])^T$$

$$\mathbf{X}_b = (X_b[0], X_b[1], \dots, X_b[M-1])^T$$

\mathbf{M}_{TDA} Es la matriz de TDA, dada por:

$$\mathbf{M}_{TDA} = \begin{bmatrix} \mathbf{0}_{M/2 \times M/2} & \mathbf{0}_{M/2 \times M/2} & -\mathbf{J}_{M/2} & -\mathbf{I}_{M/2} \\ \mathbf{I}_{M/2} & -\mathbf{J}_{M/2} & \mathbf{0}_{M/2 \times M/2} & \mathbf{0}_{M/2 \times M/2} \end{bmatrix} \quad (3.6)$$

\mathbf{C}_M^{IV} Es la representación matricial de la DCT-IV, cuyos elementos son:

$$C^{IV}(n, k) = \sqrt{\frac{2}{M}} \cos \left[\frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(k + \frac{1}{2} \right) \right], \quad n, k = 0, \dots, M-1 \quad (3.7)$$

En la figura 3.2 se ilustra el TDA causado por el proceso MDCT/IMDCT, descrito por ecuaciones (3.4) y (3.5), sobre las secuencias $x[m] = 1 - 2m/3(2M - 1)$ y $h[m] = 1/\sqrt{2}$, mostradas en la fig. 3.2a. En las figuras 3.2b y 3.2c se ilustran, respectivamente, la secuencia $x_b[m]h[m]$ y el vector $\mathbf{M}_{TDA} [\mathbf{x}_b \circ \mathbf{h}]$. Debido a que es un endomorfismo simétrico y ortogonal, la DCT-IV es su propia inversa, lo que implica que $\mathbf{M}_{TDA} [\mathbf{x}_b \circ \mathbf{h}] = \mathbf{C}_M^{IV} \mathbf{X}_b$. En la figura 3.2c, también ilustra la extensión periódica que implica el proceso DCT-IV/IDCT-IV, para $\mathbf{M}_{TDA} [\mathbf{x}_b \circ \mathbf{h}]$. El producto $\mathbf{M}_{TDA}^T \mathbf{C}_M^{IV} \mathbf{X}_b$ completa el TDA en la secuencia $x_b[m]h[m]$, como se ilustra en el lado izquierdo de la figura 3.2d. Finalmente, en el lado derecho de la figura 3.2d se muestra la secuencia $y_b[m]$, una versión de $x_b[m]$ que sufre de TDA, y del efecto compuesto análisis/síntesis de la secuencia ventana.

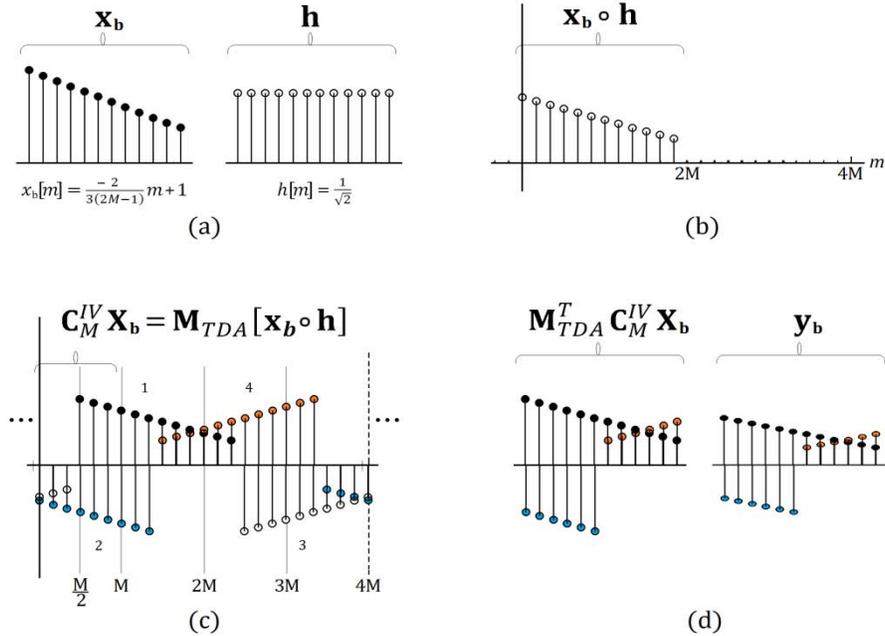


Figura 3.2. Aliasing en el dominio del tiempo.

A continuación se describe el proceso de TDAC. De acuerdo con [16], para cualquier ventana $h[n]$ que cumpla con las ecuaciones (2.22) y (2.23) con $K=1$, la reconstrucción perfecta de cualquier $x[n]$ ocurre, ya que la respuesta compuesta de la ventana, en el proceso análisis/síntesis de cada bloque, se traslapa y suma con la de los bloques adyacentes, de tal forma que el resultado es plano y cualquier TDA introducido por el

banco de filtros, es removido durante el proceso de síntesis. Con el fin de ilustrar lo anterior, en la figura 3.3 se muestra en que consiste, el fenómeno de cancelación del aliasing en el dominio del tiempo, para la secuencia “plana” $x[n] = 1$.

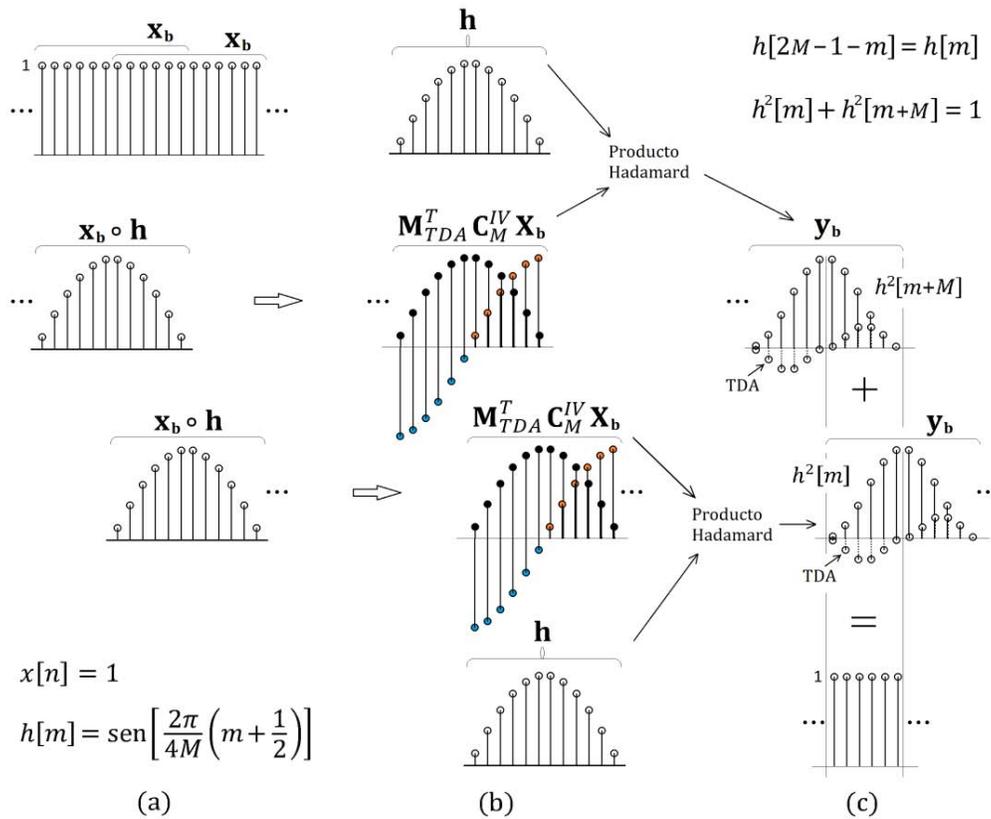


Figura 3.3. TDAC. a) Ventaneo de análisis. b) Secuencias $x_b[m]h[m]$, con TDA. c) Ventaneo de síntesis y método OLA.

3.1.2. Cálculo de la DCT-IV mediante la FFT

En el presente trabajo, de acuerdo con las ecuaciones (3.4) y (3.5), se calcularán tanto la MDCT como la IMDCT, mediante la DCT-IV. A continuación se describe el método propuesto por Malvar en [12], para el cálculo de la DCT-IV, mediante un algoritmo rápido para la DFT.

Sea $x[n]$ una secuencia real de longitud M . Si se define la secuencia compleja

$$c[n] = \sqrt{\frac{2}{M}}(x[2n] + jx[M - (2n + 1)]) e^{-j\frac{2\pi}{8M}(4n+1)}, \quad n = 0, 1, \dots, \frac{M}{2} - 1 \quad (3.8)$$

Entonces, los M coeficientes X_k , de la DCT-IV de $x[n]$, se pueden calcular mediante [12]:

$$X_{2k} = \text{Re} \left\{ e^{-j\frac{2\pi k}{2M}} C_k \right\} \quad (3.9)$$

$$X_{M-(2k+1)} = -\text{Im} \left\{ e^{-j\frac{2\pi k}{2M}} C_k \right\} \quad (3.10)$$

Donde

$$C_k = \sum_{n=0}^{M/2-1} c[n] e^{-j\frac{2\pi}{M/2}kn}, \quad k = 0, 1, \dots, \frac{M}{2} - 1 \quad (3.11)$$

es la DFT de longitud $M/2$, de la secuencia compleja $c[n]$, definida en la ecuación (3.8). En el apéndice C se muestra la deducción de las ecuaciones 3.8-3.11. Para el cálculo de la DFT, se utilizará el algoritmo FFT Radix-2 DIT.

3.2. Modelo psicoacústico basado en la MDCT

Los códecs de audio MPEG-1/2 capas I, II III y el AAC-LC, entre otros, utilizan la DFT (vía una FFT) como base de su modelo psicoacústico (PAM) [6, 26]. Estos códecs perceptuales efectúan, por bloques ventaneados de la señal de entrada, una DFT de alta resolución y, con los espectros obtenidos, determinan las curvas de enmascaramiento de cada bloque a codificar (sólo se calculan curvas para los enmascaradores principales: tipo tono o tipo ruido [26]). Cada bloque de entrada para la DFT, abarca y se encuentra centrado en tiempo, con el bloque de datos espectrales a cuantizar [6].

Un modelo psicoacústico para el MPEG-2 AAC-LC basado en la MDCT, fue presentado por primera vez por Takamizawa en [27]. Él encontró que la resolución en frecuencia del espectro generado por la MDCT, es suficiente para utilizarlo en el PAM del AAC-LC, sin afectar la calidad del sonido. Utilizar un PAM basado en el espectro de la MDCT, también ha demostrado ser útil en la reducción de la complejidad de los códecs MPEG-2/4 AAC [28-30]. En vista de lo antes mencionado, en el presente trabajo se adopta la idea de emplear los coeficientes de la MDCT, para los cálculos en el PAM. La estructura básica de este último, se inspira en el modelo psicoacústico 2 del estándar ISO/IEC MPEG [26]. A continuación se consideran algunos aspectos importantes del PAM basado en la MDCT. En el siguiente capítulo, se describirá detalladamente la implementación del mismo, como parte del códec objeto del presente trabajo.

3.2.1. Transformación al dominio de la frecuencia

En el PAM se emplean los coeficientes espectrales obtenidos en el banco de análisis o MDCT. Dado que la longitud de la MDCT a utilizar es de 1024 muestras y la frecuencia de muestreo de la señal de entrada es de 44.1kHz, la resolución del banco de filtros será de 23.2ms en el dominio del tiempo y de 43Hz en el dominio de la frecuencia.

3.2.2. Componentes enmascaradores

Se puede considerar que la mayoría del enmascaramiento en un bloque dado de señal, es producido por unos cuantos componentes enmascaradores predominantes, tipo tono y tipo ruido [6, 31, 32]. Entonces, con el fin de limitar el número de enmascaradores y, simplificar los cálculos en el PAM, las M líneas del espectro MDCT se agrupan en particiones, cuyos anchos son de una línea MDCT o de aproximadamente $1/3$ de banda crítica, dependiendo de cuál sea más ancha [7]. Para obtener la energía de cada componente enmascarador, se suman los valores $X^2[k]$ en cada partición. Al arreglo de energías resultante se le denomina “espectro particionado de energía”.

3.2.3. Índice de tonalidad y ajuste del enmascaramiento

Los enmascaradores tipo ruido tienen una mayor capacidad de enmascaramiento que los enmascaradores tipo tono [6, 7]. Para tomar en cuenta lo anterior, el espectro particionado de energía debe ser disminuido en cada partición, por un offset cuyo valor dependerá de la "tonalidad" del componente enmascarador en la partición.

El modelo psicoacústico II, utiliza la llamada medida de impredecibilidad para estimar la tonalidad de los componentes enmascaradores de un bloque dado en análisis [6]. La predictibilidad de la magnitud y fase de cada uno de los coeficientes espectrales de la DFT, a partir de sus contrapartes en las dos ventanas temporales anteriores, se utiliza para evaluar la tonalidad de los componentes enmascaradores más recientes [6]. Una alta predictibilidad temporal de las magnitudes y fases de los coeficientes espectrales se asocia con la presencia de componentes tonales. En contraste, una baja predictibilidad implica componentes tipo ruido.

En el caso del PAM basado en la MDCT, al ser ésta una transformada real, no es posible utilizar la medida de impredecibilidad para evaluar la tonalidad de un componente enmascarador. Por esta razón, en el presente trabajo se utiliza un método alternativo para determinar el offset de cada componente enmascarador. En este caso, el offset dependiente de la tonalidad se calcula utilizando el método propuesto por Naoki Shibata, empleado en el PAM NSpsyTune del codificador LAME de MP3 [33]. El NSpsyTune se enfoca sólo en el bloque en análisis presente y emplea una razón pico/promedio como estimador de tonalidad. En este caso, cuanto más "picudo" sea el espectro, más tonal será un componente enmascarador.

3.3. Cuantización y codificación

La cuantización y codificación son bloques de construcción esenciales en la implementación de cualquier códec de audio. La meta conjunta de estas dos etapas es cuantizar los datos espectrales con ayuda del PAM, de tal forma que el ruido de cuantización resulte

inaudible, y que el número de bits necesarios para codificar la señal cuantizada, permanezca por debajo un cierto valor, dependiente de la frecuencia de muestreo y de la tasa de bits deseada. A continuación, se procede a describir las partes principales involucradas en la etapa de cuantización/codificación del códec perceptual.

3.3.1. Cuantización

En la etapa de cuantización de un códec perceptual, la resolución (paso de cuantización) se debe modificar a lo largo de la frecuencia, de acuerdo con el umbral de enmascaramiento estimado en el PAM del códec. La información acerca de la resolución del cuantizador, debe ser comunicada al decodificador, lo que supone un incremento de la información complementaria en el flujo de bits. Para mantener la cantidad de información secundaria dentro de límites razonables, es común utilizar una misma resolución de cuantización, para grupos de coeficientes espectrales, adyacentes en frecuencia, denominados bandas de factores de escala (SFBs) [6].

En la etapa de codificación tanto del MP3 como la del AAC-LC, se utiliza un esquema de cuantización escalar no-uniforme de ley exponencial, en donde la resolución efectiva del cuantizador se controla mediante los llamados “factores de escala” [6]. Un cuantizador de este tipo, reduce el rango dinámico de los coeficientes espectrales y tiende a distribuir la distorsión total hacia los coeficientes de gran magnitud, donde la probabilidad de enmascaramiento perceptual es mayor. En este esquema, para especificar el cuantizador utilizado, solo se requiere de transmitir los factores de escala al decodificador.

En el presente trabajo, se utilizará un cuantizador escalar no-uniforme, cuyo comportamiento puede ser descrito mediante la ecuación [6, 34, 35]:

$$X_{qi} = nint \left[\left(\frac{2^{scf(i)/2} |X|}{2^{glsf/4}} \right)^{\frac{3}{4}} + 0.0946 \right] \quad (3.12)$$

donde i es la i -ésima banda de factores de escala, X es uno de los valores espectrales dentro de la banda a cuantizar, $2^{glsf/4}$ es el factor global de escala (que controla la tasa de transferencia), y $2^{scf(i)/2}$ es el factor local de escala (que controla la distorsión por bandas). Además, para la decuantización se tiene la expresión:

$$Xdqi = \text{signo}(X) \times (Xqi)^{4/3} \times \frac{2^{glsf/4}}{2^{scf(i)/2}} \quad (3.13)$$

3.3.2. Colocación del ruido de cuantización

Con el fin de lograr una cuantización y codificación óptimas, tanto en un sentido perceptual, como con respecto a la tasa de bits, es posible utilizar un método iterativo con dos ciclos anidados, similar al empleado por el AAC-LC [6]. Este proceso, mostrado en la figura 3.4, controla tanto la cantidad de ruido de cuantización colocado en cada banda, como la cantidad total de bits utilizados para codificar un bloque dado de coeficientes espectrales. A grandes rasgos, la tarea del ciclo de iteración interno, es la de cambiar el tamaño del paso de cuantización, modificando $glsf$ en (3.12), hasta que los datos espectrales puedan ser codificados con el número de bits disponible. Dentro de este ciclo, se lleva a cabo la cuantización de todos los coeficientes espectrales mediante (3.12), la codificación de los coeficientes cuantizados y, el cálculo del número total de bits requeridos por esta última. Por otro lado, la tarea principal del ciclo de iteración externo, es la de amplificar las bandas de factores de escala, modificando $scf(i)$ en (3.12), de tal forma que las demandas del PAM sean cumplidas en la medida de lo posible. En el siguiente capítulo, se describirá detalladamente la implementación del algoritmo utilizado en el códec objeto del presente trabajo.

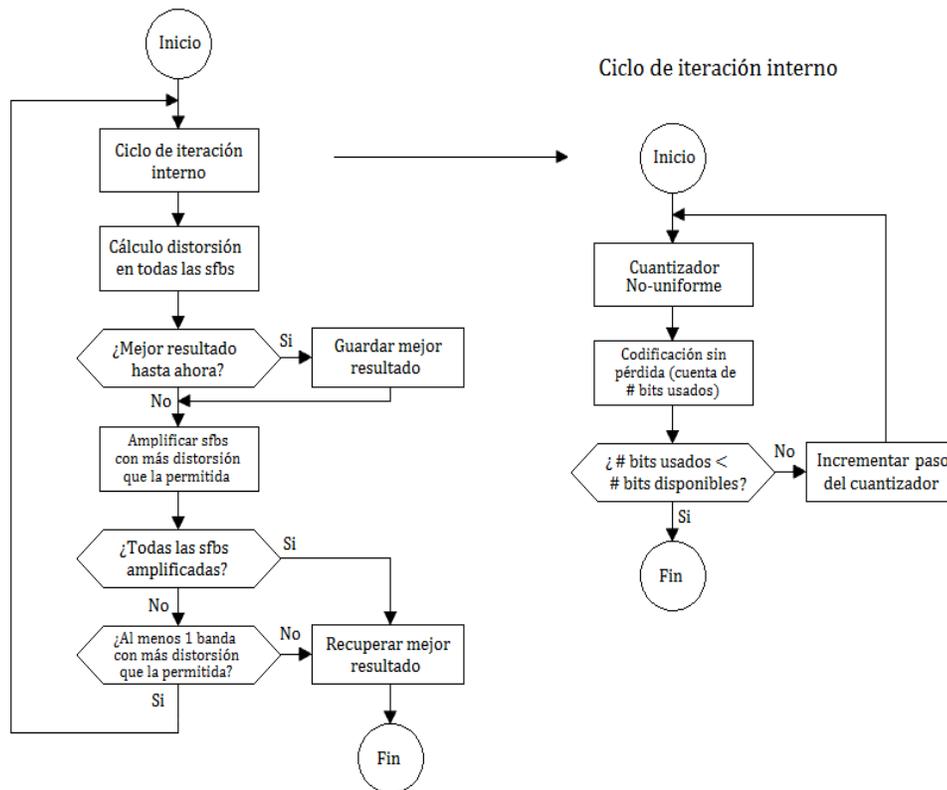


Figura 3.4. Doble ciclo anidado en la etapa de cuantización/codificación [6].

Para disminuir la redundancia estadística en la representación de los coeficientes espectrales cuantizados, es posible utilizar alguna técnica de codificación entrópica [3, 6]. La idea principal de ésta, es encontrar una función invertible, entre un conjunto de valores cuantizados $\{a_i\}$ y un nuevo conjunto $\{b_i\}$, de tal forma que el número promedio de bits por símbolo se minimice. En este caso, los símbolos más probables se representan con códigos cortos, mientras que los símbolos menos probables se representan utilizando códigos largos [3]. Existen varios esquemas de codificación entrópica, entre los que se destacan la codificación aritmética y la codificación de Huffman. En el presente trabajo, para la codificación sin pérdida de los coeficientes espectrales cuantizados, se utiliza una adaptación de la codificación Huffman del MP3 [33], misma que será descrita en el próximo capítulo.

3.4. Métodos de reducción del pre-eco

Varios métodos han sido propuestos para mitigar los artefactos de pre-eco, los cuales tienden a plagar los esquemas de codificación por bloques y se vuelven distintivamente audibles a bajas tasas de bits [8]. A continuación, se procede a describir los métodos para la reducción del pre-eco que se utilizan en el presente trabajo: el reservorio de bits y el moldeado del ruido temporal (TNS). Éste último también es utilizado por el AAC-LC [6].

3.4.1. Reservorio de bits

Las tasas de bits instantáneas requeridas para satisfacer los umbrales de enmascaramiento en cada bloque, son en general variantes en el tiempo [6]. En el caso de un algoritmo de compresión de tasa fija, el reservorio de bits es una técnica de buffer a corto plazo, que se utiliza para satisfacer la mayor demanda de bits asociada con los sonidos transitorios [6]. La idea básica es guardar en “el reservorio”, los bits sobrantes durante periodos cuasi-estacionarios de baja demanda y luego utilizarlos convenientemente durante los periodos de alta demanda. Esto resulta en una tasa de bits instantánea variante en el tiempo pero en una tasa de bits promedio fija [8]. En el códec implementado se emplea una adaptación del reservorio de bits utilizado en el MP3 [32], la cual será descrita a detalle en el próximo capítulo.

3.4.2. Moldeado del ruido temporal (TNS)

Como se mencionó en la sección 2.2.7., debido a las propiedades del sistema auditivo, los pre-ecos son enmascarados solo si ninguna cantidad significativa de ruido de cuantización está presente más allá de 2ms antes del comienzo de una señal transitoria. De lo contrario, el ruido de cuantización se percibirá como un artefacto de pre-eco. Con el fin de evitar tales artefactos, se debe procurar moldear las características temporales del ruido de cuantización, de tal forma que éste satisfaga las condiciones de enmascaramiento temporal. El problema de moldeado del ruido temporal hace difícil lograr una buena calidad

perceptual a bajas tasas de bits, para señales transitorias tales como el sonido de las castañuelas, el “glockenspiel”, el triángulo o para ciertos tipos de señales de voz [8].

Si un esquema LPC es aplicado a una señal en el dominio del tiempo, se sabe que el error de cuantización en la señal final decodificada, estará adaptado en su densidad de potencia espectral (PSD), a la forma de la PSD de la señal de original [8]. Combinando esta observación con la dualidad tiempo/frecuencia existente entre la PSD y la envolvente de Hilbert al cuadrado [8], el siguiente enunciado dual puede ser derivado: si un esquema LPC es aplicado a los datos espectrales a lo largo de la frecuencia, la forma temporal del error de cuantización aparecerá a la salida del decodificador, adaptada a la forma de la envolvente temporal de la señal de entrada. Utilizando el principio anterior, la técnica de moldeado del ruido temporal (TNS) permite al codificador, bloque por bloque, ejercer cierto control sobre la estructura temporal fina del ruido de cuantización [8].

Como se muestra en la figura 3.5, el TNS es una técnica en el dominio de la frecuencia que opera sobre los coeficientes espectrales generados por un mapeo tiempo/frecuencia. El TNS solo se aplica durante transitorios susceptibles a pre-ecos. La idea básica es aplicar codificación lineal predictiva (LPC) a lo largo de la frecuencia, ya que para una señal temporal impulsiva, la ganancia de codificación en el dominio de la frecuencia se maximiza al utilizar técnicas de predicción lineal [8]. El método trabaja de la siguiente manera. Los parámetros de un filtro LPC de predicción espectral, $A(z)$, son estimados mediante la aplicación a los coeficientes espectrales $X[k]$, de algún método estándar de estimación del MSE mínimo (e.g. Levinson-Durbin). El residual de predicción resultante, $e[k]$, es cuantizado y codificado utilizando la codificación perceptual estándar de acuerdo con el umbral de enmascaramiento original [8]. Los coeficientes de predicción son transmitidos al receptor como parte de la información complementaria para permitir la recuperación de la señal original.

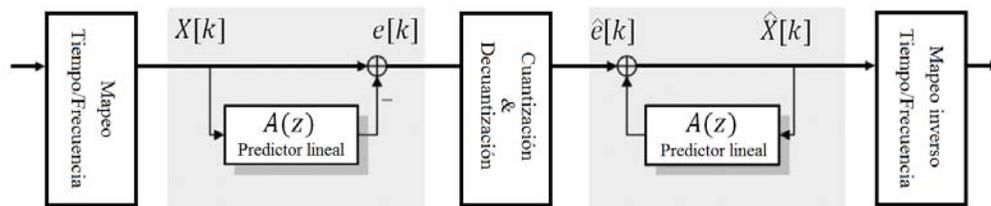


Figura 3.5. Esquema de codificación por predicción de lazo abierto en el TNS [9].

Hasta ahora la discusión del moldeado del ruido temporal estuvo basada en la noción de la DFT. En el presente trabajo, la MDCT es preferida sobre la DFT, ya que es un banco de filtros críticamente muestreado y entrega una excelente eficiencia de codificación. Al extender las consideraciones anteriores más allá de la DFT, se puede demostrar que el TNS también proporciona un efecto directo de moldeado del ruido temporal para el caso de transformadas de bloque ortogonales, como la DCT-IV [36], ver figura 3.6. No obstante, en el caso de la MDCT el moldeado del ruido temporal resultante, también está sujeto a los efectos del TDA inherente a este banco de filtros [9]. Sin embargo, debido a la naturaleza del TDAC en el banco de filtros MDCT, figura 3.3, los componentes indeseables de TDA en el TNS, son atenuados si se utiliza una ventana de “bajo” traslape entre bloques subsecuentes [8, 9], ver figura 3.7.

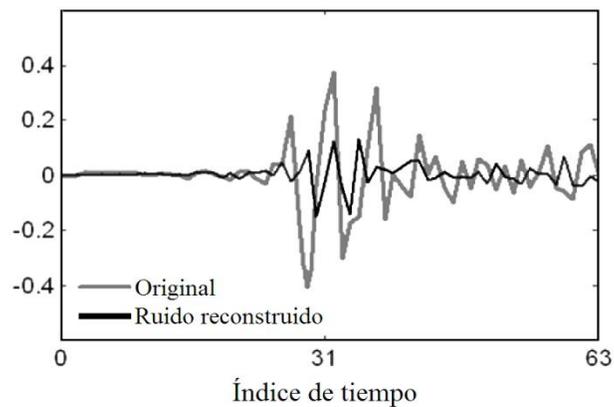


Figura 3.6. TNS utilizando la DCT-IV en un segmento de audio de 64 muestras a 8 kHz [36].

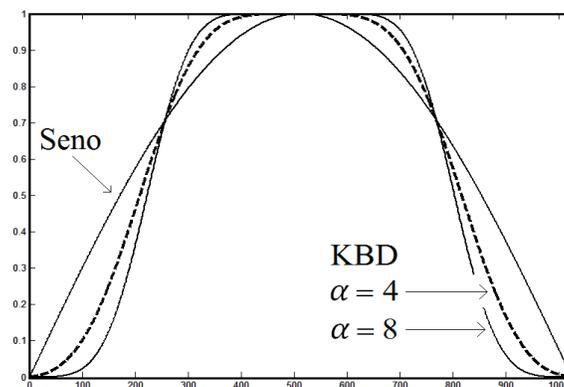


Figura 3.7. Tres ventanas TDAC con diferente grado de traslape.

Resumen

En este capítulo se describieron los elementos principales que conforman el compresor de audio propuesto. En primer lugar, se presentaron la MDCT e IMDCT y se describió su propiedad de reconstrucción perfecta, basada en el TDAC. Después, se presentó la relación existente entre la DCT-IV y la MDCT e IMDCT, y se describió como ésta es útil para calcular la MDCT directa e “inversa” mediante un algoritmo FFT. Posteriormente, se comentaron aspectos relevantes del PAM a utilizar, el cual se basa en los coeficientes espectrales obtenidos mediante la MDCT. Luego, se presentó el esquema de cuantización a utilizar, basado en el paradigma de colocación de ruido de cuantización con ayuda del PAM. Por último, se presentaron el reservorio de bits y el TNS, los cuales son métodos utilizados para la reducción del pre-eco en el códec implementado.

Capítulo 4

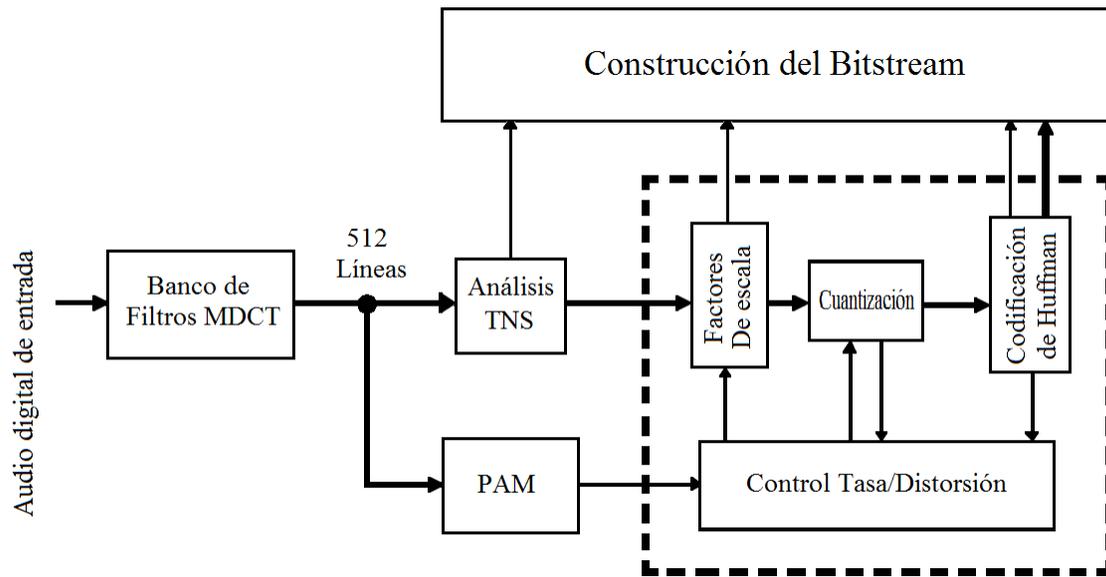
Implementación del códec perceptual

En la figura 4.1 se muestra el diagrama de bloques del códec de audio ACC, objeto del presente trabajo. El ACC opera sobre una señal estereofónica de audio en PCM lineal, muestreada a 44.1 kHz con 16 bits de cuantización, contenida en un archivo en formato WAV. La tasa de bits del audio de entrada es de $44.1 \left[\frac{\text{kilo muestras}}{\text{s}} \right] \times 16 \left[\frac{\text{bits}}{\text{muestra}} \right] \times 2 = 1411.2$ kbps.

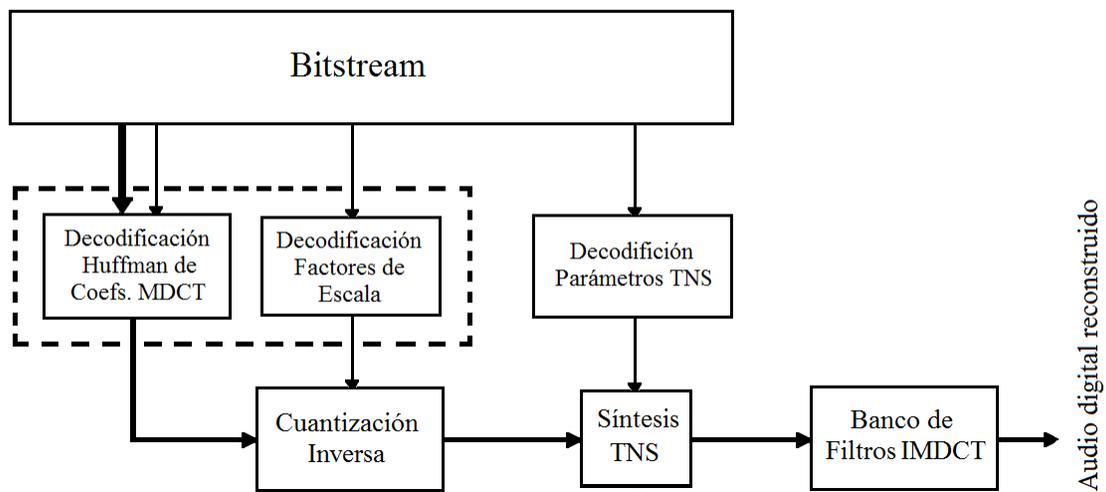
La etapa de codificación crea un archivo de audio codificado de tamaño reducido, a partir de una señal de audio en formato WAV, de mayor tamaño. La etapa de decodificación recupera, a partir del archivo comprimido, la señal de audio reconstruida y, almacena las muestras resultantes en un archivo de salida en formato WAV. La señal de audio de salida esta codificada en PCM lineal con una frecuencia de muestreo de 44.1 kHz y 16 bits de cuantización.

Las tasas de bits disponibles para la codificación son 64 kbps, 96 kbps, 128 kbps, 160 kbps, 192 kbps, 256 kbps y 320 kbps. Estas tasas proporcionan una razón de compresión de 22:1, 14:1, 11:1, 8:1, 7:1, 5:1 y 4:1, respectivamente. Para las tasas de bits de 64 kbps, 96 kbps, 128 kbps y 160 kbps, el códec ACC utiliza el TNS para reducir los artefactos de pre-eco.

A continuación se procede a describir la implementación de cada uno de los bloques que componen el códec ACC, mostrados en la figura 4.1. La implementación, llevada a cabo en lenguaje C, utiliza aritmética de punto flotante de 32 bits, a excepción del banco de filtros MDCT/IMDCT, el cual está implementado en aritmética de punto fijo formato Q22.



(a)



(b)

Figura 4.1. Diagrama de bloques del códec ACC.
a) Etapa de codificación. b) Etapa de decodificación.

4.1. Codificación

La etapa de codificación del ACC opera sobre una señal de audio PCM en un archivo formato WAV, con el fin de obtener un archivo de audio codificado de tamaño reducido, denominado bitstream. En la figura 4.2 se muestra el diagrama de flujo general de la implementación de la etapa de codificación del códec ACC.

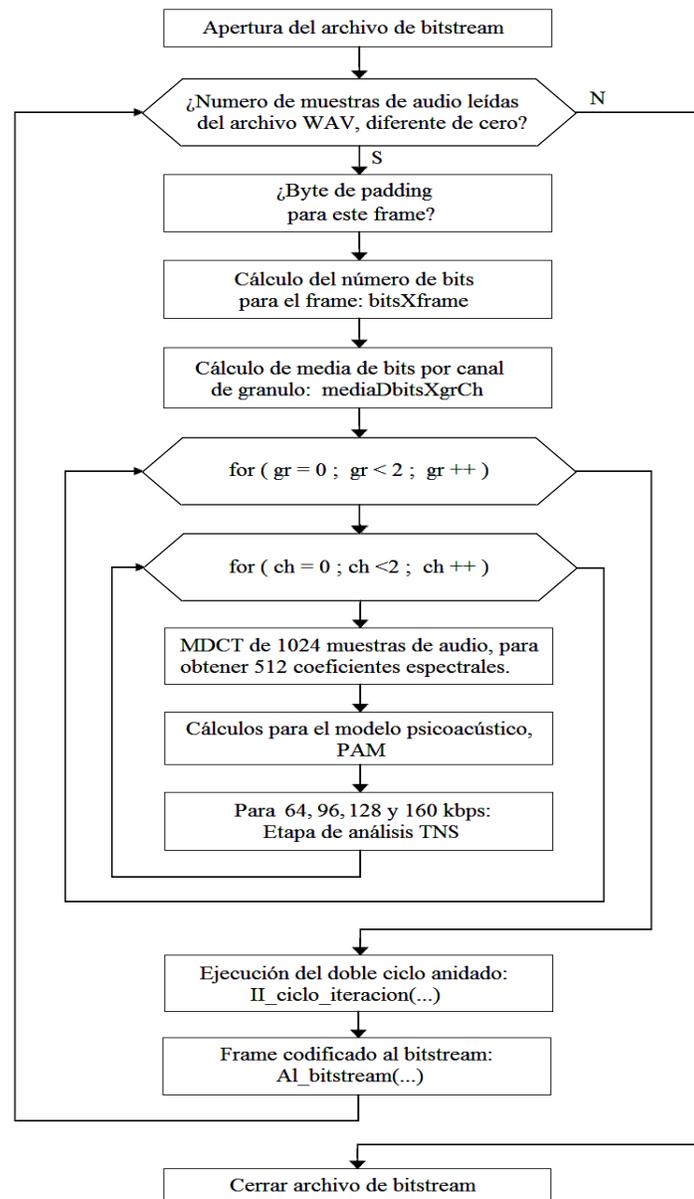


Figura 4.2 Diagrama de flujo de la implementación del codificador

4.2. Banco de filtros MDCT/IMDCT

En la figura 4.3 se muestra la implementación del bloque “Banco de filtros MDCT” de la etapa de codificación en la figura 4.1a. En primer lugar, las muestras de audio contenidas en formato Q15 en el archivo WAV, son leídas por bloques contiguos de 512 muestras por canal. Cada bloque es guardado temporalmente en un arreglo llamado `bufferDgrsWav`. Para obtener un bloque de 512 coeficientes espectrales, las muestras en formato Q22 contenidas en la segunda mitad del buffer `Xmq22` de longitud 1024, se copian a la primera mitad del mismo y luego, el contenido del arreglo `bufferDgrsWav` se convierte a formato Q22 y se guarda en la segunda mitad del arreglo `Xmq22`. Posteriormente se aplica el producto de Hadamard entre el arreglo `Xmq22` y el arreglo `hm_q22` que contiene la ventana TDAC presentada en el punto 4.2.1, en formato Q22. Al arreglo obtenido, se le aplica el algoritmo rápido para la MDCT en Q22 descrito en la sección 4.2.2. Como resultado se obtienen 512 coeficientes espectrales en formato Q22, los cuales se convierten a punto flotante de 32 bits y se almacenan temporalmente en el arreglo `Xk` para su procesamiento y codificación. Lo anterior se repite mientras se tenga por lo menos una muestra leída en el arreglo `bufferDgrsWav`.

El proceso anterior, como se muestra en la figura 4.2, se lleva a cabo para los canales izquierdo y derecho de una señal estereofónica. De un par de bloques simultáneos de 1024 muestras de audio (canales izquierdo y derecho), se obtiene un par de bloques de 512 coeficientes espectrales, el cual forma un “gránulo” de coeficientes espectrales. Para el proceso de codificación se trabaja por pares de gránulos llamados “frames”. Vale la pena mencionar que la nomenclatura de gránulos y frames se retoma de la utilizada en el formato MP3 [6, 32, 33], ya que para construir el archivo comprimido, se adaptó el constructor de bitstream MP3 implementado en el códec LAME [33].

En la figura 4.4 se muestra la implementación del bloque “Banco de filtros IMDCT” de la etapa de decodificación en la figura 4.1b. En primer lugar, cada bloque de 512 coeficientes espectrales contenido en el archivo comprimido, es decodificado decuantizado y guardado temporalmente, en formato Q22, en el arreglo `Xkrq22`. A los coeficientes en este buffer se les aplica el algoritmo rápido para la IMDCT en Q22 descrito en la sección 4.2.2, con el fin

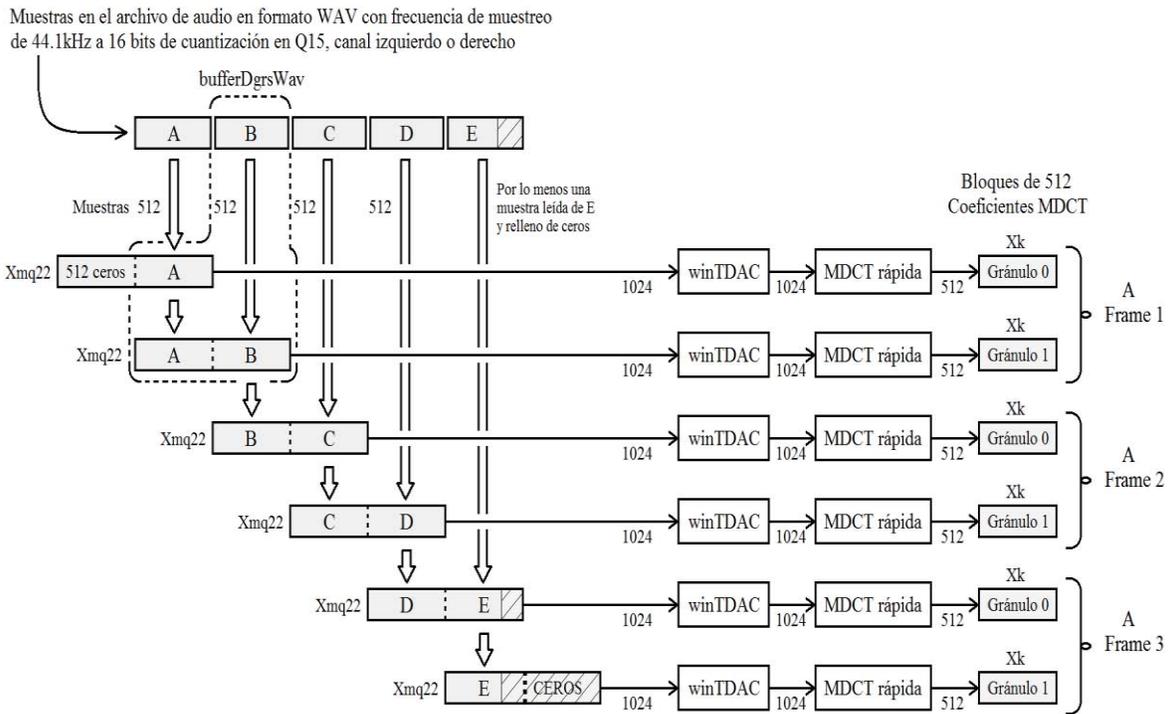


Figura 4.3. Implementación del banco de filtros de MDCT

de obtener 1024 muestras temporales en Q22, a las que se les aplica el producto de Hadamard con la ventana TDAC contenida en el arreglo `hm_q22`. A las 1024 muestras temporales con TDA resultantes, se les almacena temporalmente en el arreglo `yoh_q22`. Para obtener un bloque de 512 muestras de audio reconstruido, en formato Q22, se suma el contenido de `yoh_2daMitad` con el contenido de la primera mitad de `yoh_q22`. Para ser utilizada en la reconstrucción del siguiente bloque de 512 muestras de audio, la segunda mitad del arreglo `yoh_q22` se guarda en el arreglo `yoh_2daMitad` de longitud 512. Las 512 muestras de audio reconstruidas son convertidas al formato Q15 a 16 bits y guardadas en el archivo WAV de salida. El proceso anterior se lleva a cabo para los canales izquierdo y derecho de la señal estereofónica comprimida.

representa la ventana Kaiser-Bessel de longitud $M+1$ con factor α , e I_0 es la función Bessel modificada de primer tipo de orden cero, definida por [6]:

$$I_0[x] = \sum_{k=0}^{\infty} \left[\frac{(x/2)^k}{k!} \right]^2$$

Al ser una ventana TDAC, de acuerdo con la ecuación (2.22), la segunda mitad de la ventana KBD se obtiene por simetría.

Como se menciona en [6], el factor alfa del filtro en la ecuación (4.1) permite controlar el “trade-off” que se tiene entre el ancho de banda del lóbulo principal y la energía de los lóbulos laterales de su espectro, tal y como se muestra en la figura 4.5a. Así mismo, en la figura 4.5b se muestra cómo el factor alfa determina, en el dominio del tiempo, el nivel de traslape de la ventana TDAC.

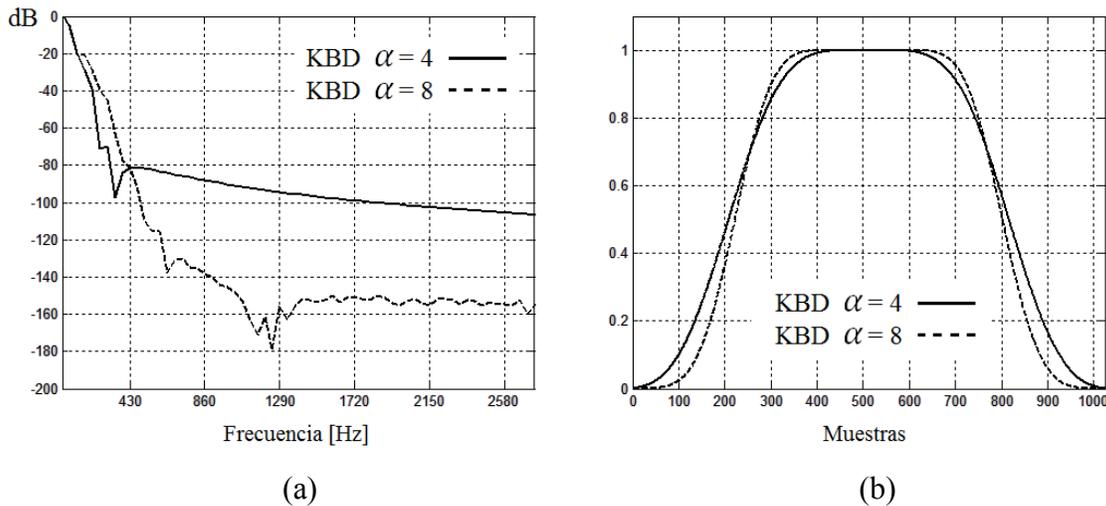


Figura 4.5. Ventana Kaiser-Bessel Derivada.

Específicamente, en el códec objeto del presente trabajo se utiliza una ventana KBD de longitud $2M = 1024$ con $\alpha = 4$, representada en formato Q22, cuyas respuestas en frecuencia y al impulso se muestran, respectivamente, en las figuras 4.5a y 4.5b.

4.2.2. Algoritmo rápido para la MDCT e IMDCT

Para llevar a cabo la MDCT se utiliza la ecuación (3.4), cuya implementación en lenguaje C se muestra en el Programa 4.1. En esta función, los apuntadores `XbloqWin` y `Xcoef_MDCT` corresponden, respectivamente, a los vectores $[\mathbf{x}_b \circ \mathbf{h}]$ y \mathbf{X}_b en la ecuación (3.4). El argumento `Log2dM` es el logaritmo base 2 de $M = 512$.

```
void MDCT_32qi(long *XbloqWin, long *Xcoef_MDCT, unsigned short Log2dM)
{
    unsigned short i3,M;
    M=1<<Log2dM; //M=2^Log2dM
    //"multiplicación" por la matriz de TDA
    for(i3=0;i3<(M>>1);i3++)
    {
        Xcoef_MDCT[i3+(M>>1)]=XbloqWin[i3]-XbloqWin[M-1-i3];
        Xcoef_MDCT[(M>>1)-1-i3]=-XbloqWin[i3+M]-XbloqWin[(M<<1)-1-i3];
    }
    DCTiv32qi(Xcoef_MDCT, Log2dM); //DCT-IV
}
```

Programa 4.1. Código en LENGUAJE C para calcular la MDCT.

Por otro lado, para llevar a cabo la IMDCT se emplea la ecuación (3.5), cuya implementación en C se muestra en el Programa 4.2. En esta función, los apuntadores `Xcoef_MDCT` y `YbloqAWin` corresponden, respectivamente, a los vectores \mathbf{X}_b y $[\mathbf{M}_{TDA}^T \mathbf{C}_M^{IV} \mathbf{X}_b]$ en la ecuación (3.5). El argumento `Log2dM` es el logaritmo base 2 de $M = 512$.

```
void IMDCT_32qi(long *Xcoef_MDCT, long *YbloqAWin, unsigned short Log2dM)
{
    unsigned short i4,M;
    M=1<<Log2dM; //M=2^Log2dM
    DCTiv32qi(Xcoef_MDCT, Log2dM); //DCT-IV
    //"multiplicación" por la matriz de TDA transpuesta
    for(i4=0;i4<(M>>1);i4++)
    {
        YbloqAWin[i4]=Xcoef_MDCT[i4+(M>>1)];
        YbloqAWin[i4+(M>>1)]=-Xcoef_MDCT[M-1-i4];
        YbloqAWin[i4+M]=-Xcoef_MDCT[(M>>1)-1-i4];
        YbloqAWin[i4+(M>>1)+M]=-Xcoef_MDCT[i4];
    }
}
```

Programa 4.2. Código en LENGUAJE C para calcular la IMDCT.

Para calcular la DCT-IV en los Programas 4.1 y 4.2, se emplea la función `DCTiv32qi()`, cuya implementación se muestra en el Programa 4.3.

```

//In place computation
//El Mmax esta determinado por el periodo del cuarto de coseno en
//CosNmax[], definido en CosNmax2048_b32qi22.h. Mmax=Nmax/2
void DCTiv32qi(long *Xdn, unsigned short Log2dM)
{
    unsigned short i2,M,SubSampFact1,Difaux;
    Difaux = Log2dNma-(Log2dM+1);
    SubSampFact1 = 1<<(Difaux);

    M=1<<Log2dM; //M=2^Log2dM

    long *Cre = malloc( ((M>>1)+1)*sizeof(long) );
    if (NULL == Cre) {
        fprintf(stderr, "malloc failed\n");
        return;
    }
    long *Cim = malloc( ((M>>1)+1)*sizeof(long) );
    if (NULL == Cim) {
        fprintf(stderr, "malloc failed\n");
        return;
    }

    long long Fesc,cos1,sen1,Acu1,Acu2,Acu3,Acu4,Acu5,Acu6;
    //para M=1024,512,256,128,64,32,16,8
    cos1=cos8M[Difaux]; //cos(2pi/8M) en Q22
    sen1=sen8M[Difaux]; //sen(2pi/8M) en Q22
    Fesc=FescM[Difaux]; //sqrt(2/M) en Q22

    for (i2=0;i2<(M>>1);i2++) {
        Acu1=(long long)Xdn[2*i2];
        Acu2=(long long)Xdn[M-(2*i2+1)];
        Acu3=(long long)CosNmax[i2*SubSampFact1];
        Acu4=(long long)CosNmax[(M>>1)-i2]*SubSampFact1];

        Acu5=((Acu3*cos1)>>b32qi)-((Acu4*sen1)>>b32qi); //cos(2pi_n/2M + 2pi/8M)
        Acu6=((Acu4*cos1)>>b32qi)+((Acu3*sen1)>>b32qi); //sen(2pi_n/2M + 2pi/8M)

        Acu3=((Acu1*Acu5)>>b32qi) + ((Acu2*Acu6)>>b32qi);
        Acu4=((Acu2*Acu5)>>b32qi) - ((Acu1*Acu6)>>b32qi);

        Cre[i2]=(long)((Fesc*Acu3)>>b32qi);
        Cim[i2]=(long)((Fesc*Acu4)>>b32qi);
    }

    FFT_32qi(Cre, Cim, (Log2dM-1));
}

```

```

for (i2=0;i2<(M>>1);i2++) {
    Acu1=(long long)Xdn[2*i2];
    Acu2=(long long)Xdn[M-(2*i2+1)];
    Acu3=(long long)CosNmax[i2*SubSampFact1];
    Acu4=(long long)CosNmax[(M>>1)-i2]*SubSampFact1];

    Acu5=((Acu3*cos1)>>b32qi)-((Acu4*sen1)>>b32qi); //cos(2pi_n/2M + 2pi/8M)
    Acu6=((Acu4*cos1)>>b32qi)+((Acu3*sen1)>>b32qi); //sen(2pi_n/2M + 2pi/8M)

    Acu3=((Acu1*Acu5)>>b32qi) + ((Acu2*Acu6)>>b32qi);
    Acu4=((Acu2*Acu5)>>b32qi) - ((Acu1*Acu6)>>b32qi);

    Cre[i2]=(long)((Fesc*Acu3)>>b32qi);
    Cim[i2]=(long)((Fesc*Acu4)>>b32qi);
}

```

Ecuación 3.8

```

FFT_32qi(Cre, Cim, (Log2dM-1));

```

Ecuación 3.11

```

for (i2=0;i2<(M>>1);i2++) {
    Acu1=(long long)Cre[i2];
    Acu2=(long long)Cim[i2];
    Acu3=(long long)CosNmax[i2*SubSampFact1]; //cos(2pi_k/2M)
    Acu4=(long long)CosNmax[(M>>1)-i2]*SubSampFact1]; //sen(2pi_k/2M)

    Acu5=((Acu1*Acu3)>>b32qi)+((Acu2*Acu4)>>b32qi);
    Acu6=(-Acu2*Acu3)>>b32qi+((Acu1*Acu4)>>b32qi);
    Xdn[2*i2]=(long)Acu5;
    Xdn[M-(2*i2+1)]=(long)Acu6;
}

```

Ecuaciones 3.9 y 3.10

```

free(Cre);
free(Cim);
}

```

Programa 4.3. Código en LENGUAJE C para calcular la DCT-IV.

La función `DCTiv32qi()` está implementada en aritmética de punto fijo en formato Q22, y obtiene la DCT-IV, según las ecuaciones 3.8-3.11, de una secuencia de longitud $M=1024$, 512, 256, 128, 64, 32, 16 o 8. Esta función utiliza el arreglo `cosNmax`, que contiene los valores en formato Q22 del $\cos[2\pi n/2048]$ para $0 \leq n \leq 512$, como look-up table para la evaluación de las secuencias $\cos[2\pi n/2M]$ y $\sin[2\pi n/2M]$ utilizadas en la función `DCTiv32qi()`. De forma similar, se utilizan los arreglos `cos8M`, `sen8M` y `FescM`, como look-up tables para la obtención, en formato Q22, de las constantes $\cos(2\pi/8M)$, $\sin(2\pi/8M)$ y $\sqrt{2/M}$. En el programa 4.3, la función `FFT_32qi()` implementa el algoritmo FFT Radix-2 DIT en formato Q22.

4.3. Modelo psicoacústico

En el códec implementado, para los cálculos realizados en el PAM, las líneas del espectro MDCT se agrupan en 63 particiones [33], cada una cubriendo aproximadamente 1/3 de banda crítica, como se muestra en la tabla 4.1.

4.3.1. Espectro particionado de energía

El espectro particionado de energía se obtiene al sumar la energía de cada partición o componente enmascarador, mediante la ecuación [31]:

$$EP(i) = \sum_{k=kba\grave{a}ja(i)}^{k=kalta(i)} X^2[k] \quad (4.2)$$

donde $X[k]$ corresponde al k -ésimo coeficiente MDCT dentro de la partición i .

Part. i	Núm. de líneas MDCT	Frec bark media Zi	Ancho escala bark	Part. i	Núm. de Líneas MDCT	Frec bark media Zi	Ancho escala bark
1	1	00.213	0.426	33	3	14.003	0.351
2	1	00.638	0.425	34	4	14.395	0.432
3	1	01.062	0.423	35	4	14.809	0.397
4	1	01.484	0.421	36	4	15.191	0.367
5	1	01.903	0.418	37	4	15.545	0.341
6	1	02.319	0.413	38	5	15.913	0.396
7	1	02.730	0.409	39	5	16.295	0.367
8	1	03.136	0.403	40	5	16.650	0.343
9	1	03.536	0.397	41	6	17.013	0.385
10	1	03.929	0.390	42	6	17.386	0.360
11	1	04.316	0.383	43	6	17.736	0.340
12	1	04.695	0.375	44	7	18.093	0.374
13	1	05.067	0.367	45	7	18.457	0.354
14	1	05.430	0.359	46	7	18.802	0.336
15	1	05.785	0.351	47	8	19.152	0.364
16	1	06.131	0.342	48	8	19.506	0.344
17	2	06.633	0.658	49	9	19.860	0.365
18	2	07.273	0.622	50	9	20.214	0.342
19	2	07.878	0.588	51	10	20.563	0.355
20	2	08.449	0.554	52	11	20.920	0.360
21	2	08.986	0.521	53	12	21.279	0.358
22	2	09.492	0.490	54	13	21.633	0.350
23	2	09.968	0.461	55	14	21.975	0.336
24	2	10.415	0.434	56	16	22.312	0.337
25	2	10.835	0.408	57	19	22.653	0.344
26	2	11.231	0.384	58	22	22.992	0.335
27	2	11.604	0.362	59	27	23.328	0.336
28	2	11.956	0.342	60	35	23.667	0.343
29	3	12.367	0.478	61	46	24.008	0.338
30	3	12.825	0.440	62	65	24.345	0.336
31	3	13.248	0.406	63	63	24.627	0.228
32	3	13.639	0.377				

Tabla 4.1. Particiones del espectro MDCT utilizadas en el PAM

4.3.2. Estimación del offset de enmascaramiento

El offset para el espectro particionado de energía, en la partición i -ésima, está dado por la expresión [33]:

$$\Delta(i) = \text{DeltaArray}[\text{index}(i)] \quad (4.3)$$

donde

$$\Delta_{prom}[] = \{10^{0.0}, 10^{-0.1}, 10^{-0.2}, 10^{-0.2}, 10^{-0.2}, 10^{-0.2}, 10^{-0.2}, 10^{-0.6}, 10^{-0.93}\}$$

$$index(i) = MIN \left(nint \left(\left\lfloor 20 * \frac{Epico/Eprom - 1}{TotLin_{parts}} \right\rfloor \right), 8 \right) \quad (4.3a)$$

siendo

$$Epico = Max(X^2[k]; kbaja(i-1) \leq k \leq kalta(i+1))$$

$$Eprom = \frac{EP(i-1)/N_L(i-1) + EP(i)/N_L(i) + EP(i+1)/N_L(i+1)}{3}$$

$TotLin_{parts}$: Total de coeficientes MDCT en las particiones $i-1, i, i+1$

con

$X^2[k]$: Energía del k -ésimo coeficiente MDCT

$EP(i)$: Espectro particionado de energía, ecuación (4.2)

$N_L(i)$: Número de coeficientes MDCT en la partición i -ésima, tabla 4.1

4.3.3. Convolución con las funciones de propagación

El umbral de enmascaramiento para cada partición en el bloque de señal en análisis, se calcula mediante “la convolución” de una función de propagación, con cada uno de los elementos del espectro particionado de energía [6, 31, 32, 33]:

$$T(a) = \Delta_{prom}(a) Rn(a) \sum_{b=1}^{Np} SF(z_a - z_b) EP(b) \Delta(b), \quad a = 1, 2, \dots, Np \quad (4.4)$$

donde

T : Umbral de enmascaramiento

EP : Espectro particionado de energía

SF : Función de propagación definida por la ecuación (2.6)

z_i : Frecuencia bark, ecuación (2.4), a la mitad de la partición i

$\Delta(b)$: Offset de enmascaramiento según la ecuación (4.3)

$Rn(a)$: Factor de re normalización en la partición a , dado por [31, 33]

$$Rn(a) = \frac{1}{\sum_{i=1}^{N_p} SF(z_i - z_a)}, \quad a = 1, 2, \dots, N_p$$

$\Delta_{prom}(a)$: Offset promedio en la partición a , dado por [33]

$$\Delta_{prom}(a) = \frac{1}{\sum_{b=1}^{N_p} \lambda(SF(z_a - z_b))} \sum_{b=1}^{N_p} \Delta(b) \times \lambda(SF(z_a - z_b)), \quad a = 1, \dots, N_p$$

con

$$\lambda(SF(z_a - z_b)) = \begin{cases} 1, & SF(z_a - z_b) \neq 0 \\ 0, & SF(z_a - z_b) = 0 \end{cases}$$

El proceso anterior es equivalente a propagar en frecuencia, la energía de cada enmascarador y combinar las energías relativas.

En la ecuación (4.4), la forma de la función de propagación provoca en cada partición a , una sobrestimación del incremento de energía de enmascaramiento debido a los efectos de la propagación. El factor de re normalización $Rn(a)$, toma en cuenta lo anterior multiplicando en cada partición a , por el inverso de la ganancia de energía, asumiendo una energía uniforme de 1 en cada partición a .

Los valores de la función de propagación empleados en la ecuación (4.4), se almacenan como look-up table en un arreglo matricial \mathbf{SF} , con elementos:

$$\mathbf{SF}(a, b) = SF(z_a - z_b), \quad a, b = 1, 2, \dots, 63 \quad (4.5)$$

Donde $SF(dz)$ es la función de propagación definida por la ecuación (2.6) y, z_a y z_b son, respectivamente, las frecuencias bark centrales de las particiones a y b , cuyos valores se encuentran en la tabla 4.1.

De la ecuación (4.5), se puede apreciar que la columna i -ésima de la matriz \mathbf{SF} , contiene la función de propagación para a un componente enmascarador ubicado en la partición i -ésima. Lo anterior se muestra en la figura 4.6a, para las particiones $i = 2, 14, 26, 38, 50, 62$.

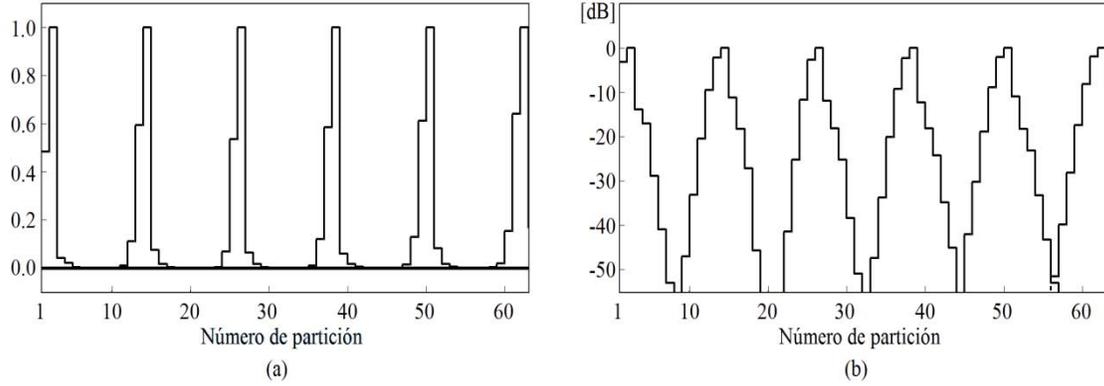


Figura 4.6. Funciones de propagación centradas en las particiones 2, 14, 26, 38, 50 y 62, respectivamente, representadas en a) la escala lineal, b) la escala decibel.

4.3.4. Cálculo del umbral global de enmascaramiento

Finalmente, se obtiene el umbral global de enmascaramiento para la partición i -ésima mediante la expresión [6, 31, 32, 33]:

$$T_g(i) = \text{MAX}(\text{ATHmin}(i), \text{MIN}(T(i), 2 \times T_{t-1}(i), 16 \times T_{t-2}(i))) \quad (4.6)$$

Donde $T(i)$, $T(i)_{t-1}$ y $T(i)_{t-2}$ son, respectivamente, los umbrales de enmascaramiento para el bloque actual y los dos previos, según la ecuación (4.4), y

$\text{ATHmin}(i)$: Umbral auditivo mínimo para la partición i -ésima

siendo

$$\text{ATHmin}(i) = \text{MIN}\left(10^{0.1 \times (T_q(f_k) - 18)}; k_{baja}(i) \leq k \leq k_{alta}(i)\right)$$

Donde $T_q(f_k)$, dado por la ecuación (2.3), es el valor del umbral auditivo a la frecuencia $f_k = 22050(2k + 1)/1024$ [Hz].

En la ecuación (4.6), el utilizar el mínimo de entre el umbral de enmascaramiento actual y versiones ponderadas de los dos umbrales anteriores, se conoce como control de pre-eco y se utiliza para incrementar la precisión de codificación del bloque actual, en caso de que éste sea el primero que cubra una porción transitoria de la señal a codificar [6, 8].

Finalmente, el umbral global de enmascaramiento obtenido mediante la ecuación (4.6), se mapea hacia las bandas de factores de escala, definidas en la tabla 4.2, con el fin de ser utilizado en la etapa de cuantización y codificación del códec implementado, como se describirá en la sección 4.4.

SFB No.	Líneas MDCT	Frecuencia [Hz]	Zmed [Bark]	Zancho [Bark]
1	0 - 003	21.5 - 00150.7	00.85	1.7
2	4 - 007	193.8 - 00323.0	02.52	1.6
3	8 - 011	366.1 - 00495.3	04.12	1.5
4	12 - 015	538.3 - 00667.5	05.60	1.4
5	16 - 019	710.6 - 00839.8	06.95	1.3
6	20 - 023	882.9 - 01012.1	08.16	1.1
7	24 - 027	1055.1 - 01184.3	09.24	1.0
8	28 - 031	1227.4 - 01356.6	10.19	0.9
9	32 - 037	1399.7 - 01615.0	11.21	1.2
10	38 - 045	1658.1 - 01959.5	12.42	1.3
11	46 - 055	2002.6 - 02390.2	13.67	1.2
12	56 - 067	2433.3 - 02907.0	14.88	1.2
13	68 - 079	2950.1 - 03423.8	15.94	0.9
14	80 - 097	3466.9 - 04199.0	16.99	1.2
15	98 - 117	4242.1 - 05060.3	18.10	1.1
16	118 - 143	5103.4 - 06180.1	19.22	1.2
17	144 - 173	6223.1 - 07472.1	20.36	1.1
18	174 - 211	7515.1 - 09108.6	21.45	1.1
19	212 - 255	9151.7 - 11003.5	22.42	0.9
20	256 - 303	11046.6 - 13070.7	23.18	0.6
21	304 - 371	13113.8 - 15999.3	23.80	0.6
22	372 - 511	16042.3 - 22028.6	24.42	0.6

Tabla 4.2. Bandas de factores de escala, considerando una frecuencia de muestreo de 44.1kHz y 512 coeficientes MDCT por bloque de análisis espectral.

Específicamente, el umbral global de enmascaramiento en cada banda de factor de escala está dado por:

$$T_{SFB}(sb) = (1 - \alpha[sb - 1]) \times T_g(b_u[sb - 1]) + \sum_{b=b_u[sb-1]+1}^{b_u[sb]-1} T_g(b) + \alpha[sb] \times T_g(b_u[sb]) \quad (4.7)$$

donde $T_g(b)$, para $1 \leq b \leq 63$, esta dado por la ecuación (4.6), $T_g(0) \equiv 0$, $T_g(64) \equiv 0$ y, $\alpha[sb]$ y $b_u[sb]$ son parámetros dados en la tabla 4.3.

sb	bu[sb]	$\alpha[sb]$	sb	bu[sb]	$\alpha[sb]$
0	0	0.0000			
1	5	0.0000	12	37	0.2500
2	9	0.0000	13	39	0.8000
3	13	0.0000	14	43	0.0000
4	17	0.0000	15	46	0.0000
5	19	0.0000	16	49	0.3333
6	21	0.0000	17	52	0.4545
7	23	0.0000	18	55	0.5000
8	25	0.0000	19	58	0.0909
9	28	0.0000	20	60	0.0286
10	31	0.0000	21	61	0.7391
11	34	0.2500	22	64	1.0000

Tabla 4.3. Parámetros para convertir particiones a bandas de factores de escala

4.3.5. Entropía perceptual

De acuerdo con Johnston [31], el concepto de entropía perceptual (PE) define el número mínimo de bits promedio por muestra espectral, necesarios para codificar una señal, sin introducir en ésta una diferencia perceptual con respecto a la señal original. Una estimación útil de la entropía perceptual está dada por [33]:

$$PE = - \sum_{i=1}^{63} N_L(i) \times \log \left(\frac{T_g(i) + 1}{EP(i) + 1} \right) \quad (4.8)$$

Donde

$N_L(i)$: Número de coeficientes MDCT en la partición i -ésima.

$T_g(i)$: Umbral global de enmascaramiento, ecuación (4.6)

$EP(i)$: Espectro particionado de energía, ecuación (4.2)

La entropía perceptual forma parte de los cálculos realizados para determinar el número de bits disponibles para codificar un bloque dado de coeficientes espectrales (ver sección 4.6.4), y supone una medida de llanura espectral del bloque en análisis, donde un valor alto de PE implica un espectro plano (correspondiente a un bloque de señal transitoria) [6]. En el apéndice D se muestra un ejemplo gráfico de los cálculos realizados en el PAM.

4.4. Etapa de cuantización y codificación

La función de la etapa de cuantización y codificación utilizada en el códec implementado, es llevar a cabo un esquema de colocación de ruido [2, 3, 6], el cual consiste de dos ciclos anidados que controlan, tanto la energía del ruido de cuantización inyectado en cada banda de factores de escala, como el número total de bits utilizados para la codificación de los coeficientes espectrales MDCT. En este caso y en contraste con un esquema de colocación de bits, utilizado en códecs de audio sin codificación entrópica tales como el Dolby AC3 y el MPEG-1 capas 1 y 2 [6], el consumo individual de bits en cada banda de factores de escala no se controla de manera explícita, sino que resulta implícitamente del proceso. Considerando la ecuación (3.12), al aumentar el factor de escala global se reduce la precisión general de cuantización y se disminuye la demanda total de bits de codificación. Al multiplicar los coeficientes espectrales en una banda de factores de escala dada, por el factor local de escala correspondiente, se incrementa la precisión local de cuantización y se incrementa la demanda total de bits de codificación. Para cumplir con la tasa de bits disponible e intentar cumplir con la calidad perceptual deseada, la precisión general del cuantizador se va reduciendo en favor de aumentar la precisión de las bandas de factores de escala con mayor distorsión de la permitida.

En la figura 4.7 se muestra el diagrama de flujo de la función `II_ciclo_iteracion ()`. Esta función se encarga, por cada canal en cada gránulo del frame en codificación, de calcular el número de bits disponibles para la codificación del espectro MDCT, inicializar las variables de cuantización y codificación utilizadas en el doble ciclo anidado, calcular el factor de escala inicial (primer $2^{glsf/4}$), ejecutar el ciclo de iteración interno y, reajustar el tamaño del reservorio de bits.

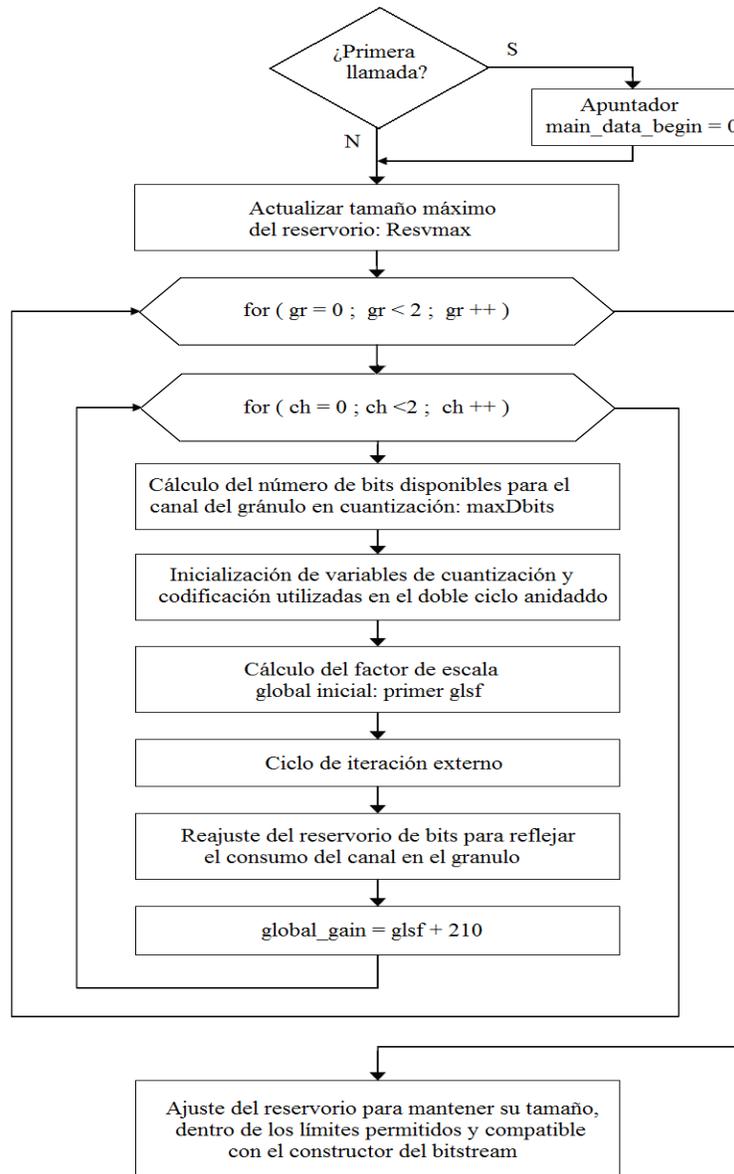


Figura 4.7. Diagrama de flujo de la función `II_ciclo_iteracion ()`.

4.4.1. Ciclo externo

El ciclo de iteración externo controla por bandas de factores de escala, el ruido producido por la cuantización de los coeficientes MDCT, la cual es llevada a cabo en el ciclo de iteración interno. Los factores de escala en la ecuación (3.12) son utilizados para moldear el ruido de cuantización, con el fin de lograr que éste pueda ser enmascarado completamente. El moldeado espectral del ruido de cuantización se lleva a cabo multiplicando, antes de la cuantización, a los coeficientes dentro de cada banda de factores de escala, por el factor de escala correspondiente. Los factores de escala son cuantizados logarítmicamente, base $\sqrt{2}$, utilizando el parámetro $scf(i)$. La primera iteración del ciclo externo comienza con $scf(i) = 0$, para las 22 bandas de factores de escala. En cualquier iteración, si la distorsión en cualquiera de las bandas de factores de escala, excede la distorsión permisible determinada en el PAM, se incrementa en 1 el parámetro $scf(i)$, con el fin de aumentar la resolución de cuantización en aquellas bandas de factores de escala. La figura 4.8 muestra el diagrama de bloques que describe la implementación del ciclo anidado externo [32].

El parámetro $scf(i)$ de cada una de las 22 bandas de factores de escala, y el número de bits necesarios para representarlos, se guardan en cada iteración del ciclo externo. Si la ejecución del ciclo externo se interrumpe sin haber alcanzado el resultado final, estos valores junto con el parámetro $glsf$ (que determina el factor de escala global), y el espectro cuantizado; pueden ser utilizados para la construcción del bitstream. Por cada iteración del ciclo externo, se ejecuta el ciclo de iteración interno para llevar a cabo la cuantización en sí. Los parámetros requeridos para llevar a cabo el ciclo interno son: el espectro MDCT con los factores de escala, $2^{scf(i)/2}$, aplicados a los coeficientes dentro de las bandas de factores de escala correspondientes, y el número total de bits disponibles para el canal en el gránulo del frame en codificación.

El umbral global de energía de enmascaramiento calculado en el PAM, supone la máxima distorsión permitida en cada banda de factores de escala. De forma análoga, para cada banda de factores de escala, la distorsión introducida por la cuantización llevada a cabo en el ciclo de iteración interno, se cuantifica mediante la energía del error de cuantización en la banda, la cual está dada por la ecuación:

$$enerQ(i) = \sum_{k=kba(ja(i))}^{k=kalta(i)} \left(|X[k]| - Xqi[k]^{4/3} \times 2^{(glsf-2scf(i))/4} \right)^2 \quad (4.9)$$

Donde $X[k]$ y $Xqi[k]$ son, respectivamente, el k -ésimo coeficiente MDCT y su correspondiente valor escalado y cuantizado, dentro de la banda de factores de escala i -ésima.

El ciclo de iteración externo continúa hasta que ninguna banda de factores de escala exceda la distorsión permitida, se hayan amplificado todas las bandas de factores de escala, o se requieran más de 4 bits para representar cualquiera de los 22 factores de escala.

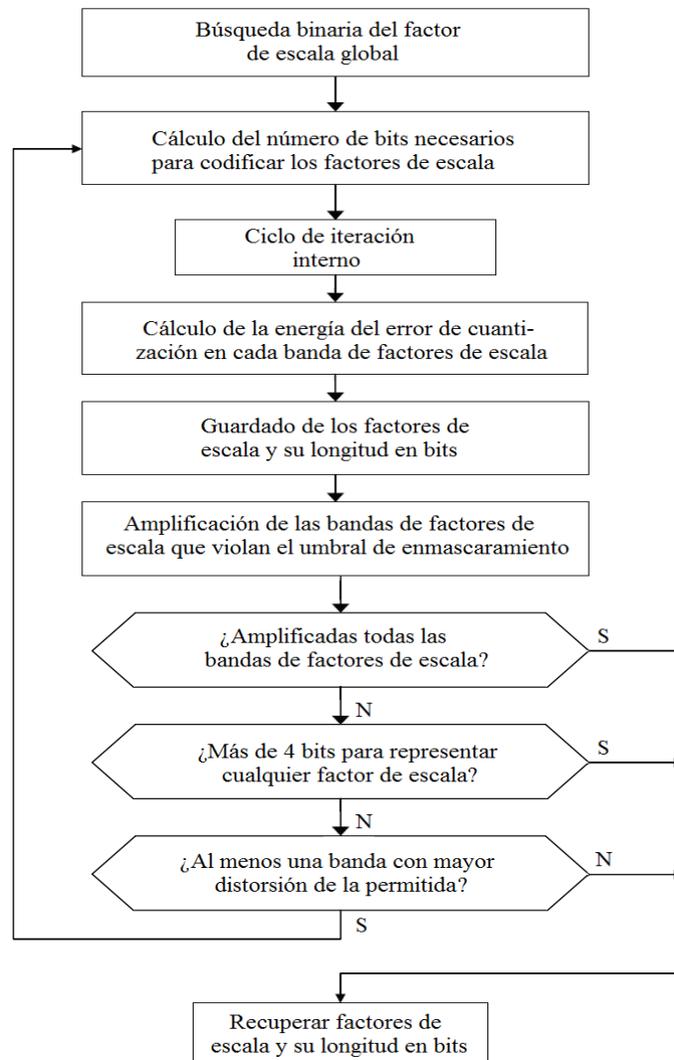


Figura 4.8. Diagrama de flujo de la implementación del ciclo de iteración externo.

4.4.2. Ciclo interno

La meta del ciclo de iteración interno es obtener, vía la ecuación (3.12), una versión cuantizada de los coeficientes MDCT, y asegurar que no se consuman más de los bits disponibles para la codificación de la misma. Esto se logra al iterar entre diferentes cuantizadores y selección de tablas de código de Huffman. Los coeficientes MDCT escalados en el ciclo externo, son primero cuantizados con valores no mayores a 8206, y posteriormente organizados en 5 regiones. A cada región diferente de cero, se le asigna una tabla de código Huffman de acuerdo a sus características estadísticas. Si el resultado de utilizar esta codificación de longitud variable excede la cantidad de bits disponibles, el cuantizador se ajusta al incrementar en uno el parámetro $glsf$, con el fin de reducir la precisión global del cuantizador y el consumo de bits. Este proceso se repite hasta que la tasa de bits objetivo sea satisfecha. Una vez que la tasa objetivo se satisface, el control regresa al ciclo de iteración externo. Los pasos involucrados en el ciclo de iteración interno se muestran en el diagrama de bloques en la figura 4.9 [32].

4.4.3. Codificación Huffman

El códec implementado utiliza para la codificación entrópica de los coeficientes MDCT cuantizados, una adaptación de la codificación Huffman empleada en el estándar del MP3 [33]. El codificador delimita los 512 coeficientes MDCT, en tres distintas regiones de longitud variable. Lo anterior posibilita que cada región sea codificada con un conjunto diferente de tablas de Huffman, cada tabla previamente ajustada para la estadística de una región en particular.

A altas frecuencias, el codificador identifica una región llamada “zero” [26, 33], en donde todos los coeficientes espectrales fueron cuantizados a cero. El tamaño de esta región se puede deducir a partir del tamaño de las otras dos regiones y no necesita ser codificado, la única restricción es que debe contener un número par de ceros, dado que las otras dos regiones agrupan sus valores, cada una, en secuencias con un número par de elementos.

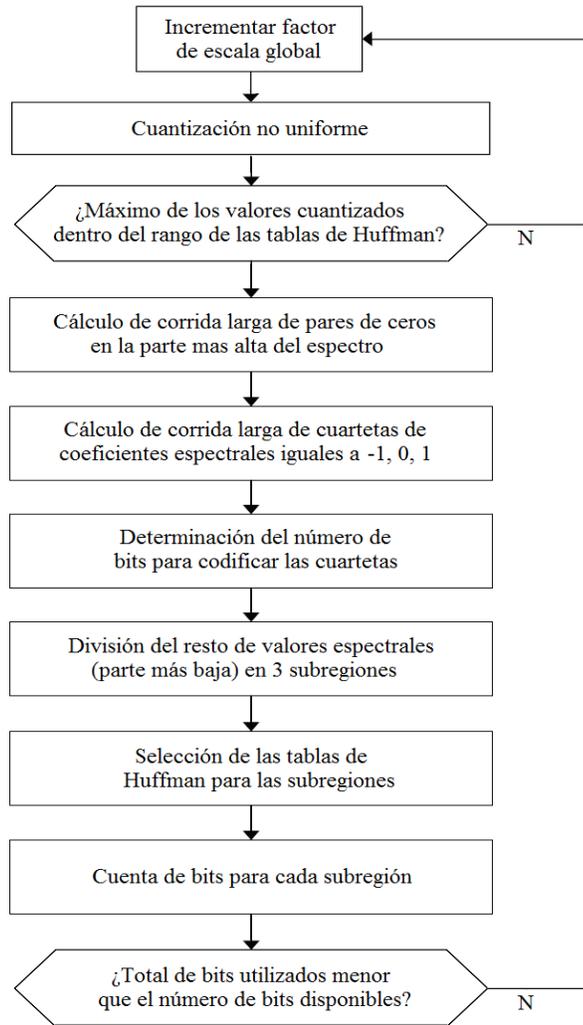


Figura 4.9. Diagrama de flujo de la implementación del ciclo de iteración interno.

La segunda región, llamada “count1”, contiene una serie de valores contiguos de solo -1, 0 o +1 [26, 33]. En esta región, los coeficientes espectrales se agrupan, para su codificación Huffman, en cuartetas sin traslape de la forma $(|X_q[k+3]|, |X_q[k+2]|, |X_q[k+1]|, |X_q[k]|)$. El código Huffman para cada cuarteta es seguido por hasta 4 bits adicionales, ya que para cada valor diferente de cero en la cuarteta, se tiene un bit de signo. Si éste es 0, el valor es positivo, si es 1, el valor es negativo. En esta región, el codificador puede escoger entre dos diferentes tablas para codificar las cuartetas [33], ver tabla 4.4.

Finalmente, la tercera y última región, llamada “bigvalues”, cubre los valores espectrales restantes, los cuales son codificados en pares [26]. Esta región se subdivide en tres particiones, cada una cubierta por una tabla de Huffman independiente elegida de un conjunto de 15 posibles tablas [33]. Para cada partición, la tabla de Huffman que mejor se adecúe a la estadística de la señal, se selecciona durante el ciclo de iteración interno, en la etapa de cuantización y codificación. Esta búsqueda dinámica permite tanto un incremento en la eficiencia de codificación como una reducción en la sensibilidad al error [2].

En la región “bigvalues”, los valores cuantizados pueden llegar a ser tan grandes como $2^{13} - 1 + 15 = 8206$ [33], y cuando se debe codificar un gran número de símbolos, la construcción de una sola tabla de código Huffman óptima se vuelve muy difícil [3]. Para utilizar los recursos de manera más eficiente, el esquema de codificación Huffman sólo se emplea para pares de valores cuantizados en el intervalo de 0 a 15 [33].

Para codificar los pares $(|x = X_q[k]|, |y = X_q[k + 1]|)$, en una subregión de bigvalues que no contenga ningún valor mayor a 15, se utiliza una de 13 posibles tablas [33], ver tabla 4.5. El código Huffman obtenido para cada par, es seguido en el bitstream por hasta 2 bits de signo, para indicar con 0 o 1, si un valor diferente de 0 es positivo o negativo, respectivamente.

Para codificar los pares $(|x|, |y|)$, en una subregión de “bigvalues” que contenga por lo menos un valor mayor a 15, se utiliza una de 2 posibles tablas [33]. Si $|x|$ y $|y|$ son menores a 15, el código Huffman obtenido para cada par, es seguido en el bitstream por hasta 2 bits de signo, para indicar con 0 o 1, si un valor diferente de 0 es positivo o negativo, respectivamente. Si $|x|$ y/o $|y|$ es mayor a 15, entonces es sustituido en el par a codificar por 15, y el código Huffman obtenido es seguido, en el bitstream, por $|x|-15$ y/o $|y|-15$, cada uno representado con “linbits” bits y seguido por el bit de signo correspondiente.

Como se muestra en la tabla 4.6, las 15 tablas de Huffman se complementan con el parámetro “linbits” para obtener un total de 29 “tablas” diferentes para codificar las tres regiones de “bigvalues” [33]. La información complementaria contiene información de cuál de las 29 “tablas” hay que utilizar. Para las primeras 13 tablas se considera que linbits = 0, para la tabla 16 considera que linbits = 1, 2, 3, 4, 6, 8, 10 o 13, y para la tabla 24 considera linbits = 4, 5, 6, 7, 8, 9, 11 o 13.

Tabla 32	
Cuarteta	Código
0	1
1	0101
2	0100
3	00101
4	0110
5	000101
6	00100
7	000100
8	0111
9	00011
A	00110
B	000000
C	00111
D	000010
E	000011
F	000001

Tabla 33	
Cuarteta	Código
0	1111
1	1110
2	1101
3	1100
4	1011
5	1010
6	1001
7	1000
8	0111
9	0110
A	0101
B	0100
C	0011
D	0010
E	0001
F	0000

Tabla 4.4. Par de tablas utilizadas para la codificación de la región count1 [33].

Tabla 1	
($ x , y $)	Código
(0,0)	1
(0,1)	001
(1,0)	01
(1,1)	000

Tabla 2	
($ x , y $)	Código
(0,0)	1
(0,1)	010
(0,2)	000001
(1,0)	011
(1,1)	001
(1,2)	00001
(2,0)	00011
(2,1)	00010
(2,2)	000000

Tabla 3	
($ x , y $)	Código
(0,0)	11
(0,1)	10
(0,2)	000001
(1,0)	001
(1,1)	01
(1,2)	00001
(2,0)	00011
(2,1)	00010
(2,2)	000000

Tabla 5	
($ x , y $)	Código
(0,0)	1
(0,1)	010
(0,2)	000110
(0,3)	0000101
(1,0)	011
(1,1)	001
(1,2)	000100
(1,3)	0000100
(2,0)	000111
(2,1)	000101
(2,2)	0000111
(2,3)	00000001
(3,0)	0000110
(3,1)	000001
(3,2)	0000001
(3,3)	00000000

Tabla 4.5. Ejemplo de tablas de código Huffman para la región bigvalues [33].

Tabla núm.	Tamaño	Valores para x e y	Linbits
0	0×0		
1	2×2	0,1	0
2	3×3	0,1,2	0
3	3×3	0,1,2	0
4	No se utiliza		
5	4×4	0,1,2,3	0
6	4×4	0,1,2,3	0
7	6×6	0,1,2,3,4,5	0
8	6×6	0,1,2,3,4,5	0
9	6×6	0,1,2,3,4,5	0
10	8×8	0,1,2,3,4,5,6,7	0
11	8×8	0,1,2,3,4,5,6,7	0
12	8×8	0,1,2,3,4,5,6,7	0
13	16×16	0,1,2,...,14,15	0
14	No se utiliza		
15	16×16	0,1,2,...,14,15	0
16	16×16	0,1,2,...,14,15	1
17,18,19, 20,21,22,23			2, 3, 4, 6, 8, 10, 13
24	16×16	0,1,2,...,14,15	4
25,26,27, 28,29,30,31			5, 6, 7, 8, 9, 11, 13

Tabla 4.6. Posibles "tablas", para la región bigvalues [33].

4.5. Etapas de análisis y síntesis TNS

Para la implementación del bloque de TNS se utiliza un modelo de predicción lineal todo-polos o autorregresivo, en el que una secuencia x_k se representa mediante la ecuación [37]:

$$x_k = - \sum_{r=1}^p a_r x_{k-r} + e_k \quad (4.10)$$

Donde p es el orden del predictor, a_r con $1 \leq r \leq p$, son los coeficientes del predictor y, e_k es el error de predicción o residual.

Suponiendo que la secuencia x_k es una muestra de un proceso aleatorio estacionario, en la ecuación (4.10), el conjunto de coeficientes a_r , que minimizan la energía de la secuencia e_k

en un sentido de mínimos cuadrados, se obtiene al resolver el sistema de ecuaciones normales [37]:

$$\sum_{r=1}^p R(|i-r|)a_r = -R(i), \quad 1 \leq i \leq p \quad (4.11)$$

donde

$$R(j) = \sum_{k=0}^{Lx-1-j} x_k x_{k+j}, \quad j \geq 0 \quad (4.11a)$$

con $x_k \neq 0$ para $k = 0, 1, \dots, Lx - 1$.

Este método para estimar los coeficientes a_r en la ecuación (4.10), se conoce como el método de autocorrelación y, por la estructura particular de la ecuación (4.11) es posible utilizar para la resolución de la misma, el algoritmo recursivo Levinson-Durbin descrito mediante [37]:

$$E_0 = R(0) \quad (4.12a)$$

$$k_i = - \left[R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j) \right] / E_{i-1} \quad (4.12b)$$

$$a_i^{(i)} = k_i \quad (4.12c)$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1 \quad (4.12d)$$

$$E_i = (1 - k_i^2) E_{i-1} \quad (4.12e)$$

En donde las ecuaciones (4.12b) a (4.12.e) se resuelven de forma recursiva para $i = 1, \dots, p$. La solución final está dada por

$$a_j = a_j^{(p)}, \quad 1 \leq j \leq p \quad (4.12f)$$

Las cantidades intermedias k_i , $1 \leq i \leq p$, en la ecuación (4.12), se conocen como coeficientes de reflexión o coeficientes PARCOR y a partir de ellos, una vez concluida la

recursión de Levinson-Durbin, es posible calcular los coeficientes del predictor mediante las ecuaciones (4.12.c) y (4.12d) para $i = 1, 2, \dots, p$. Lo anterior resulta muy útil dado que para representar un predictor lineal todo-polos, es mejor utilizar los coeficientes de reflexión [37], ya que la cuantización directa de los coeficientes a_r , puede conducir a problemas de estabilidad en el filtro IIR que supone la ecuación (4.10).

4.5.1. Etapa de análisis TNS

En primer lugar se lleva a cabo el análisis LPC de orden $p = 12$, descrito por las ecuaciones (4.11) y (4.12), sobre los coeficientes espectrales MDCT $X[k]$, $k = k_{TNS}, \dots, M1-1$, del bloque de audio a codificar. Como resultado de este análisis, se obtienen los coeficientes a_r del predictor, los correspondientes coeficientes de reflexión k_r y, la ganancia de predicción esperada, definida por la ecuación (4.13) [8].

$$PG = \frac{R(0)}{E_p} \quad (4.13)$$

Si la ganancia de predicción esperada excede el límite de 1.5, entonces se activa el procesamiento TNS. En este caso, los coeficientes MDCT a cuantizar y codificar, correspondientes a $k = k_{TNS}, \dots, M1-1$, son reemplazados por la secuencia residual e_k , la cual es obtenida a partir de la ecuación (4.10), considerando que $x_k = X[k + k_{TNS}]$, $k = 0, 1, \dots, M1 - k_{TNS} - 1$. Nótese que k_{TNS} define a partir de que frecuencia se comienza a aplicar el TNS (e.g. $k_{TNS} = 46$ implica 2002 Hz).

En la información transmitida al decodificador, se incluye una bandera indicando si se emplea o no el procesamiento TNS y, en caso afirmativo, también se incluyen los valores de los coeficientes de reflexión del filtro empleado para la obtención de e_k , estos últimos cuantizados, utilizando 5 bits, mediante la ecuación [37, 38]:

$$kq_i = \text{round} \left[\frac{(2^4 - 1)}{\pi/2} \times \text{sen}^{-1}(ki) \right] \quad (4.14)$$

4.5.2. Etapa de síntesis TNS

En el decodificador, se recupera la información relativa al TNS, la cual incluye la bandera que indica si se utiliza o no el procesamiento TNS y, en caso afirmativo, también incluye los valores cuantizados de los coeficientes de reflexión del predictor empleado durante la codificación.

Los coeficientes de reflexión se decuantizan utilizando la ecuación

$$k_i = \text{sen} \left(\frac{\pi/2}{(2^4 - 1)} \times kq_i \right) \quad (4.15)$$

Y, a partir de éstos, empleando las ecuaciones (4.12c) y (4.12d), se obtienen los coeficientes del predictor en la ecuación (4.10). Posteriormente, mediante este predictor se estima el espectro MDCT, $\hat{X}[k + k_{TNS}] = x_k$, $k = 0, \dots, M1 - k_{TNS} - 1$, a partir de la secuencia residual \hat{e}_k recuperada del bitstream.

4.6. El bitstream

El archivo de audio codificado se denomina bitstream y contiene tanto los datos de audio codificados, como las instrucciones requeridas para que el decodificador recupere la señal de audio reconstruida [6]. Al igual que el bitstream utilizado en el MP3 [6, 26, 32], el bitstream del códec implementado se divide en bloques de datos llamados frames, cuya estructura se muestra en la figura 4.10.

El tamaño de un frame puede ser de N bytes o N+1 bytes dependiendo del bit de "padding" [33]. El parámetro N está dado por la ecuación:

$$N = \text{int} \left(\frac{1}{8} \times \frac{\text{bitrate}}{44100} \times 2 \times 512 \right) \quad [\text{bytes}] \quad (4.16)$$

Donde bitrate es la tasa de bits promedio elegida para la codificación de la señal de audio.

El tamaño en bits de un frame se obtiene mediante la ecuación [33]:

$$\text{bitsXframe} = 8 \times (\text{N} + \text{padding}) \quad [\text{bits}] \quad (4.17)$$

Donde N está dado por la ecuación (4.16), y padding puede valer 0 o 1, para ajustar el bitrate medio a la frecuencia de muestreo de 44.1kHz. A continuación se procede a describir cada uno de los campos que componen un frame de audio estereofónico.

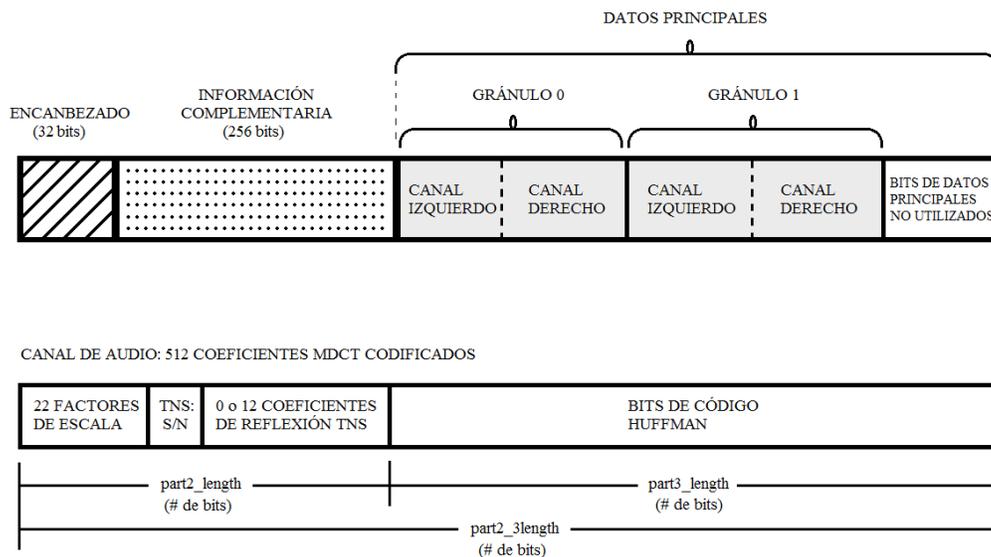


Figura 4.10. Estructura de un frame de audio estereofónico codificado.

4.6.1. Encabezado de frame

Los primeros 32 bits del frame constituyen el encabezado, el cual está formado por:

- 12 bits para la palabra de sincronización 0xFFF.
- 4 bits de índice de tabla de bitrate. Donde la tabla de bitrate está dada por:

$$\text{bitrate}[] = \{ 64, 96, 128, 160, 192, 256, 320 \};$$

- 1 bit de padding. Si éste es igual a 1, el frame contiene 1 byte adicional para ajustar el bitrate medio a la frecuencia de muestreo de 44.1kHz.
- 15 bits no utilizados en el códec implementado.

4.6.2. Información complementaria

El campo de información complementaria de cada frame de audio estéreo tiene un tamaño de 256 bits [6, 32, 33]. Esta sección contiene la información necesaria para decodificar los datos principales de un frame. En la figura 4.11 se muestra la estructura en lenguaje C, que agrupa las variables utilizadas para conformar la información complementaria de un frame dado. Esta estructura es una adaptación de la estructura utilizada en el codificador LAME [33]. A continuación se procede a describir cada uno de los elementos que conforman la estructura `side_info_t`.

```

/* Información complementaria ACC. */
typedef struct {
    int    main_data_begin;
    struct {
        struct gr_infoA {
            unsigned part2_3_length;
            unsigned big_values;
            unsigned global_gain;
            unsigned idx_ArrBits2sfbs;
            unsigned table_select[3];
            unsigned region0_count;
            unsigned region1_count;
            unsigned count1table_select;

            unsigned part2_length;
            unsigned count1;
            unsigned address1;
            unsigned address2;
            unsigned address3;
            float    glsf;

        } grn[2];
    } ch[2];
} side_info_t;

```

Figura 4.11. Estructura de datos complementarios

Variables cuyos valores conforman la sección de información complementaria de cualquier frame [32, 33]:

- `main_data_begin`
Este dato de 9 bits proporciona la ubicación, en el bitstream, del comienzo de los datos principales de un frame.

Por cada canal, en cada gránulo que conforma un frame se tiene:

- `part2_3_length`
Este dato de 12 bits contiene el número de bits de datos principales utilizados por: los 22 factores de escala, la bandera de uso del TNS, los 12 coeficientes TNS y los datos de código Huffman. Dado que la longitud de la información complementaria es siempre la misma, este valor puede ser utilizado para determinar el inicio de la información principal de cada canal en cada gránulo.
- `big_values`
Variable de 9 bits que indica el número de pares de coeficientes MDCT, pertenecientes a la región bigvalues de codificación Huffman.
- `global_gain`
Variable de 8 bits con el valor $\text{global_gain} = \text{glsf} + 210$, el cual representa el factor de escala global en el bitstream.
- `idx_ArrBits2sfbs`
Índice representado con 4 bits, utilizado para seleccionar por cada canal, el número de bits empleados para la transmisión de cada uno de los 22 parámetros $scf(i)$, ecuaciones (3.12) y (3.13), de acuerdo a la siguientes dos tablas:

$$\text{bits2sfbs1_Arr}[] = \{ 0, 0, 0, 0, 3, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4 \};$$

$$\text{bits2sfbs2_Arr}[] = \{ 0, 1, 2, 3, 0, 1, 2, 3, 1, 2, 3, 1, 2, 3, 2, 3 \};$$

Cada elemento en la primera tabla indica el número de bits utilizados para codificar los factores de escala de las bandas de la 1 a la 11. Cada elemento en la segunda

tabla indica el número de bits utilizados para codificar los factores de escala de las bandas de la 12 a la 22.

- `table_select[]`
Arreglo con tres valores, cada uno con tamaño de 5 bits, que indica la tabla de Huffman utilizada para codificar cada una de las tres particiones en que se divide la región `bigvalues`.
- `region0_count`
Representa con 4 bits, el número de bandas de factores de escala en la primera partición de la región `bigvalues`.
- `region1_count`
Representa con 3 bits, el número de bandas de factores de escala en la segunda partición de la región `bigvalues`. El número de bandas en la tercera partición se calcula con la ayuda del parámetro `big_values`.
- `count1table_select`
Este valor de 1 bit, indica la selección de una de las dos posibles tablas de código Huffman, para la región de cuartetos con magnitud no mayor que 1.

Variables auxiliares no incluidas en la información complementaria del frame [32, 33].

- `part2_length`
Este valor contiene el número de bits de datos principales utilizados para los factores de escala, la bandera de uso del TNS, y los 12 coeficientes TNS.
- `count1`
Número de cuartetos contenidas en la región `count1`.
- `address1`, `address2` y `address3`
Estos valores, respectivamente, se utilizan para definir el límite superior, en términos del número de coeficiente MDCT correspondiente, de cada una de las tres particiones de la región `bigvalues`. Los límites de las particiones se encuentran alineados con la división del espectro en bandas de factores de escala.

- $glsf$
Flotante utilizado durante el doble ciclo anidado de iteración, para representar el factor global de escala, $2^{-glsf/4}$, ecuación (3.12).

4.6.3. Datos principales

En el códec implementado, como se observa en la figura 4.10, la sección de datos principales de un frame está compuesta de dos gránulos, cada uno con $512 + 512$ coeficientes MDCT codificados, correspondientes a un par de bloques simultáneos de audio, cada bloque con longitud de 1024 muestras en el dominio del tiempo.

La sección de “datos principales” de un frame contiene, por cada canal en cada gránulo: 22 factores de escala, la bandera de empleo del TNS, 12 coeficientes de reflexión si se utiliza el TNS, y los datos en código Huffman.

El número promedio disponible de bits de datos principales, por cada canal en cada gránulo de un frame estéreo, está dado por:

$$\text{mediaDbitsXgrCh} = \frac{\text{bitsXframe} - (32 + 256)}{2 \times 2} \text{ [bits]} \quad (4.18)$$

donde bitsXframe está dado por la ecuación (4.17)

4.6.4 El reservorio de bits

Durante un periodo de señal transitoria, el control de pre-eco en el PAM, ecuación (4.6), puede conducir a un alto valor de entropía perceptual, ecuación (4.8), lo cual indica un gran incremento en el requerimiento de bits de codificación [6, 32]. Por otro lado, después del proceso de cuantización y codificación, durante periodos de señales de audio cuasi-estacionarias, habrá frames en los cuales el número de bits utilizados por el codificador Huffman es menor que la cantidad promedio, definida por la ecuación (4.18). Entonces, para tratar de satisfacer la necesidad adicional de bits, durante señales transitorias, y aprovechar los bits no utilizados por la codificación Huffman, durante periodos de señales

cuasi-estacionarias, el códec implementado utiliza el “reservorio de bits”, una técnica de buffer a corto plazo empleada por los codificadores de audio en formato MP3 [32]. En el ejemplo ilustrado en la figura 4.12, no todos los bits en la sección “datos principales” del frame 0 son utilizados. Estos bits se donan al reservorio. Los datos principales del frame 1 utilizan los bits del frame 0 en el reservorio, y los propios bits del frame 1. El frame 2 utiliza los bits del frame 1 en el reservorio, y la totalidad de los bits en la sección “datos principales” del frame 2, son donados al reservorio. El siguiente frame, el frame 3, requiere un gran número de bits y, en consecuencia, además de los bits en su sección “datos principales”, el frame utiliza todos los bits disponibles en el reservorio, correspondientes a los frames 1, 2. El reservorio ha quedado vacío. Finalmente, el frame 4 no utiliza bits del reservorio ya que solo utiliza los bits de su propia sección “datos principales”. Vale la pena, recalcar que dentro de los bits del reservorio disponibles para un frame dado, no se incluyen los bits en la sección “datos principales”, del frame en cuestión.

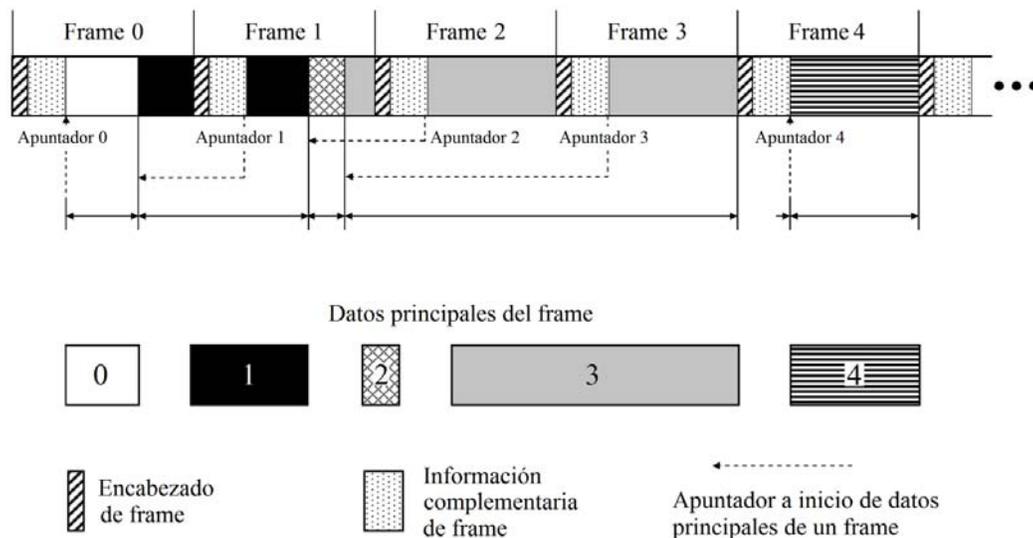


Figura 4.12. Reservorio de bits

Entonces, aunque cada frame codificado tiene el mismo número de bits, los datos de audio correspondientes a un frame en particular, pueden comenzar antes o inmediatamente después del encabezado correspondiente. La posición relativa al encabezado puede variar. En el caso de que el reservorio de bits esté vacío, los datos codificados comenzarán

directamente después del encabezado. El apuntador `main_data_begin`, de 9 bits, contiene la posición respecto al encabezado, del byte de inicio de la señal de audio codificada. El apuntador `main_data_begin` se utiliza para encontrar la información principal, en el buffer de entrada del decodificador. El codificador solo puede tomar bits donados de frames pasados, no puede tomar bits de frames futuros [32, 33]. A continuación, se procede a describir el funcionamiento del reservorio de bits utilizado en el códec implementado:

Al comenzar la codificación de una señal de audio, el tamaño del reservorio, `ResvSize`, es de 0 bits. Al inicio del procesamiento de un frame dado de audio, ver figura 4.7, el tamaño máximo permisible para el reservorio, `ResvMax`, se determina de la siguiente forma [32], [33]:

$$\text{bitsXframeMax} = \frac{320}{48} \times 2 \times 576 = 7680$$

Si (`bitsXframe > bitsXframeMax`)

$$\text{ResvMax} = 0$$

Sino

$$\text{ResvMax} = \text{MIN}(\text{bitsXframeMax} - \text{bitsXframe}, (2^9 - 1) \times 8)$$

Después, como se puede apreciar en la figura 4.7, antes de la ejecución del ciclo de iteración externo, para cada canal en cada gránulo del frame en codificación, se calcula el número máximo de bits disponibles para los datos de audio codificados, `maxDbits` [32]:

$$\text{moreBits} = 3.1 \times \text{PE}[\text{gr}][\text{ch}] - \text{mediaDbitsXgrCh}$$

$$\text{addBits} = 0$$

Si (`moreBits > 100`)

$$\text{addBits} = \text{MAX}\left(\text{ResvSize} - \frac{8}{10}\text{ResvMax}, \text{MIN}\left(\text{moreBits}, \frac{8}{10}\text{ReservSize}\right)\right)$$

$$\text{maxDbits} = \text{MIN}(2^9 \times 8 - 1, \text{mediaDbitsXgrCh} + \text{addBits})$$

Una vez concluidos los cálculos en el ciclo externo para un canal dado, el número de bits no utilizados por los datos de audio codificados del canal, se agrega al reservorio para reflejar el consumo de bits del canal.

$$\text{ResvSize} = \text{ResvSize} + (\text{mediaDbitsXgrCh} - \text{part2_3_length}[\text{gr}][\text{ch}])$$

4.7. Decodificación

El decodificador ACC opera sobre el bitstream codificado para reconstruir la señal de audio PCM. En la figura 4.13 se muestra el diagrama de flujo de la implementación de la etapa de decodificación del códec ACC.

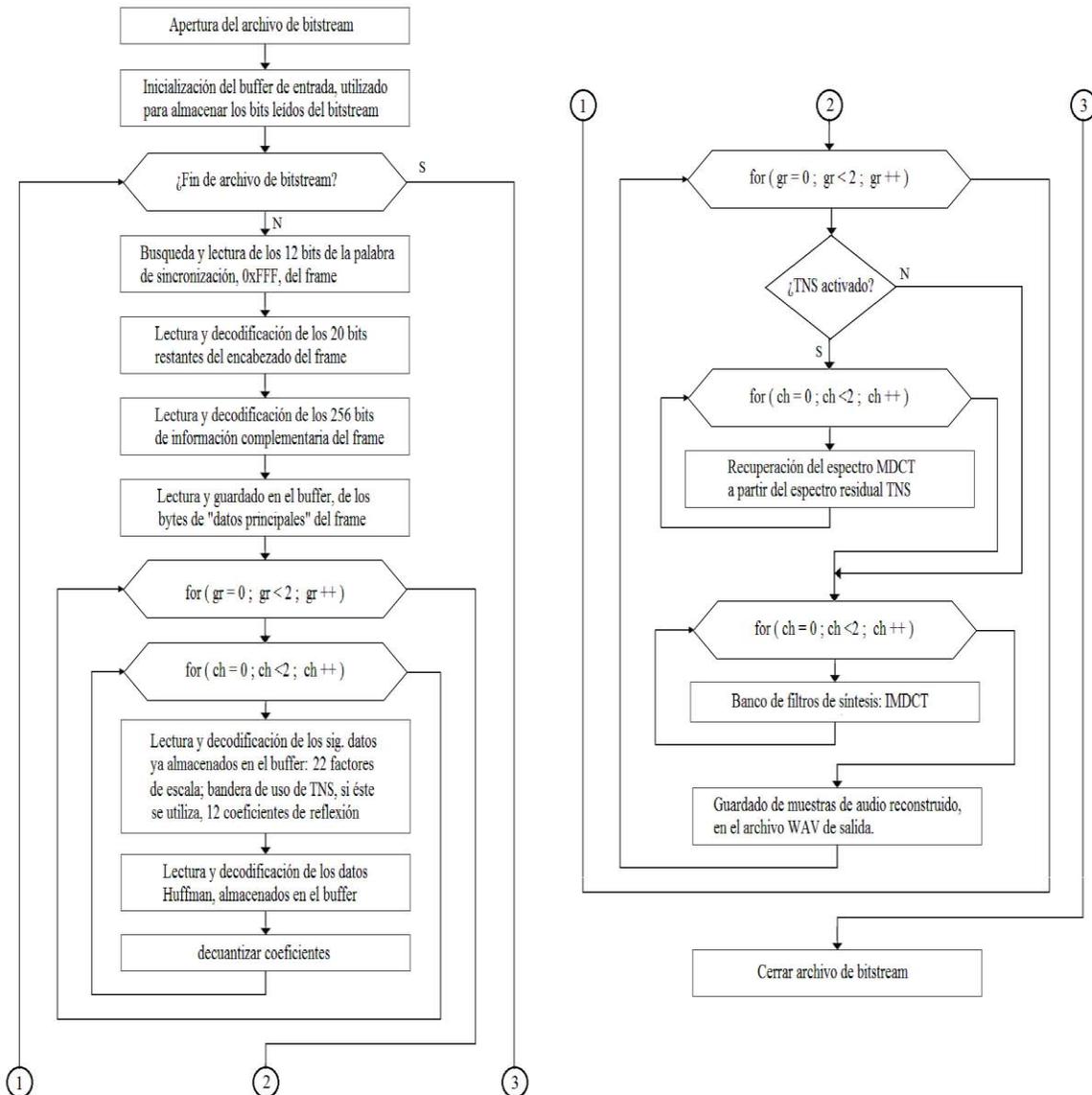


Figura 4.13. Diagrama de flujo de la implementación del decodificador

4.7.1 Buffer de lectura de bitstream

El empleo del reservorio de bits es equivalente a una variación local de la tasa de bits del codificador. Para permitir esta desviación, el decodificador utiliza un buffer de entrada, cuyo tamaño es igual al número máximo de bits que un frame puede utilizar del reservorio de bits [32, 33]. Para las distancias de los datos principales dentro del reservorio de bits, los tamaños del encabezado y de la información secundaria no se toman en cuenta, ya que éstos son de longitud fija y ocurren a intervalos regulares en el bitstream [32, 33].

4.7.2 Sincronización y lectura del encabezado

Para comenzar la decodificación de un frame dado, el decodificador debe detectar el comienzo del mismo en el bitstream. Esto se efectúa al buscar y leer en el bitstream, la palabra de sincronización 0xFFF. Este proceso se facilita ya que la distancia entre el comienzo de dos palabras de sincronización consecutivas está totalmente determinada y es de N o $N+1$ bytes, dependiendo del bit de padding. El valor de N está dado por la ecuación (4.16).

Una vez detectado el inicio de un frame, se procede a leer los 20 bits restantes del encabezado, los cuales indican la tasa de bits empleada para la codificación, y si se ha utilizado el byte de padding.

4.7.3 Lectura de información complementaria

La sección de información complementaria de un frame, contiene los datos necesarios para decodificar los datos principales. Específicamente, contiene la información perteneciente a las regiones y tablas de Huffman a utilizar durante el proceso de decodificación, la información para la recuperación de los factores locales de escala y, el factor de escala global codificado. La información complementaria también contiene el `main_data_begin`, un apuntador al comienzo de los datos principales, los cuales no necesariamente se localizan adyacentes a su información complementaria. Para una señal estéreo la información complementaria es de 256 bits.

4.7.4 Lectura de los datos principales

La lectura de los bloques de “datos principales”, se lleva a cabo de tal forma que todos los bloques correspondientes a un frame dado, se encuentran en el buffer de entrada cuando éste recibe el encabezado del frame siguiente. El decodificador omite cualquier encabezado y la información complementaria correspondiente, al decodificar los datos principales.

4.7.5 Decodificación de los datos principales

Los bits representando cada valor $scf(i)$, se leen de acuerdo a un tamaño en la tabla `bits2sfbs1_Arr` o `bits2sfbs1_Arr`, especificado por el índice `idx_ArrBits2sfbs`. Los valores decodificados, $scf(i)$, se utilizan para calcular los factores de cada banda de factores de escala. La bandera de uso del TNS utiliza 2 bits, 0 indica no TNS y 1 indica el uso del TNS. Cada coeficiente de reflexión cuantizado está representado con 5 bits. Los valores recibidos se decuantizan y utilizan en la etapa de síntesis TNS.

La decodificación Huffman se lleva a cabo utilizando una máquina de estados [33]. Esta máquina trabaja con una tabla, donde cada entrada es la información para un nodo en el árbol decodificador. La información necesaria para implementar los árboles de código Huffman, se obtiene de las tablas en el archivo `tables.c` obtenido en [33].

Las palabras de código individuales no se encuentran separadas en los bits de código Huffman entrante. Por tanto, una sola palabra en medio de los bits de código, no puede ser identificada sin comenzar a decodificar desde el principio de las palabras de código. Todos los bits de información necesaria para decodificar los bits de código Huffman se obtienen de la información complementaria. En primer lugar, se decodifican las tres particiones de la región “`bigvalues`”, utilizando las tablas indicadas por `table_select[]`. Las líneas de frecuencia en las regiones `region0`, `region1` y `region2` se decodifican en pares hasta que el número de pares alcanza el valor especificado por `big_values`. Los bits de código Huffman restantes, pertenecientes a la región “`count1`”, se decodifican utilizando la tabla especificada por `count1table_select`. La decodificación se lleva a cabo hasta que todos los bits han sido decodificados o los valores cuantizados para 512 líneas de frecuencia han sido decodificados. Si hay más bits de código Huffman que los necesarios para decodificar 512

coeficientes, estos son descartados como bits de relleno. El decodificador Huffman debe detectar la presencia de valores “-4nbits” y los bits de signo también, con el fin de reconstruir las líneas de frecuencia cuantizadas.

4.7.6 Recuperación de los coeficientes MDCT

Una vez recuperados del bitstream, el factor global de escala, los factores de escala locales, los coeficientes espectrales cuantizados, y si aplica, los coeficientes de reflexión TNS; se procede a recuperar los coeficientes espectrales decuantizados utilizando la ecuación (3.13), obteniendo:

$$Xdqi[k], \quad k = 0, 1, 2, \dots, k_{TNS}, \dots, 511 \quad (4.19)$$

Donde k_{TNS} indica a partir de que coeficiente MDCT se aplicó el análisis TNS.

En la ecuación anterior, si no se utiliza el TNS, entonces $k_{TNS} = 511$ y los 512 valores espectrales corresponden directamente a los coeficientes MDCT decuantizados. Si se utiliza el TNS, entonces los valores para $k \geq k_{TNS}$, corresponden a la secuencia residual MDCT decuantizada, ver sección 4.5.2. En este caso la etapa de síntesis TNS se ocupa de recuperar los coeficientes MDCT correspondientes a $k = k_{TNS}, \dots, 510, 511$.

4.7.7 Reconstrucción de señal de audio PCM lineal

Finalmente a partir de los coeficientes MDCT recuperados, utilizando el banco de filtros de síntesis descrito en la sección 4.2, se obtiene la señal de audio en código PCM lineal. Esta señal se guarda en un archivo de salida en formato WAV.

Resumen

En este capítulo se describió la implementación de los elementos que conforman el códec de audio ACC. En primer lugar, se presentó la implementación del banco de filtros MDCT/IMDCT. Luego, se describió detalladamente la implementación del modelo psicoacústico utilizado en el ACC. Después, se discutió la implementación del proceso de cuantización y codificación, el cual incluye el doble ciclo anidado de cuantización y la codificación Huffman de los coeficientes MDCT cuantizados. Posteriormente, se detalló la implementación de las etapas de análisis y síntesis TNS. Además, se describió el formato de bitstream utilizado por el ACC. Por último, en este capítulo se describió el proceso de decodificación llevado a cabo por el ACC, para reconstruir la señal de audio de salida, en formato WAV.

Capítulo 5

Evaluación y resultados del códec ACC

Las medidas de calidad convencionales, tales como el SNR o el MSE por bloques, proporcionan medidas simples y objetivas de la calidad de audio, pero no toman en cuenta los fenómenos psicoacústicos que pueden conducir a grandes diferencias en la calidad percibida. Por otro lado, en los esquemas objetivos de medición perceptual, basados en modelos de la audición, se tiene que los resultados predichos no se correlacionan bien con la calidad de audio percibida. Teniendo en cuenta lo anterior, en la codificación perceptual de audio se ha confiado, tradicionalmente, en pruebas de escucha formales para valorar la calidad de un sistema de audio, cuando se requiere de una valoración altamente confiable [6].

La ineficiencia de las medidas de calidad objetivas simples, para medir la distorsión percibida, fue puesta de manifiesto dramáticamente a finales de los años 80, cuando J. Johnston and K. Brandenburg presentaron el llamado “milagro de los 13 dB”. En ese ejemplo, dos señales procesadas con un SNR medido de 13 dB fueron presentadas a una audiencia. En el primer caso, a la señal original se le agregó ruido blanco, mientras que en el segundo caso, a la señal original se le agregó ruido perceptualmente moldeado. En el caso con ruido blanco, la distorsión se percibió como un siseo de fondo bastante molesto. En contraste, en el caso con ruido perceptualmente moldeado, la percepción de la distorsión varió entre ser apenas notable y ser inaudible. Aunque la medida de SNR fue la misma para ambas señales, la calidad percibida fue muy diferente, la segunda señal siendo juzgada como una señal de muy buena calidad [6].

Por lo tanto, para la evaluación de la calidad perceptual del códec implementado, se utilizan los métodos para la evaluación subjetiva de pequeños deterioros en sistemas de audio, presentados en la recomendación ITU-R BS.1116-3 [39]. No obstante, en la sección 5.3 se utiliza el criterio de SNR definido en [3], para comparar el desempeño del códec implementado, con el desempeño del codificador de MP3 LAME [33], y con el desempeño del codificador AAC-LC, incluido en el software iTunes 12.4.3.1.

5.1. Método de prueba perceptual

El método de prueba utilizado en [39], se conoce como método “doble-ciego, estímulo-triple con referencia escondida”. Este método consiste en presentar a un escucha 3 señales (estímulos): la señal de referencia, R, y las señales de prueba A y B. Una de las señales de prueba es nuevamente la señal de referencia y la otra es la señal codificada. Se denomina “doble ciego” porque tanto el conductor de la prueba como el escucha, desconocen la correspondencia de las señales de prueba A y B.

Al escucha se le permite repetir y cambiar entre R, A y B. Posteriormente, se le pide que califique las señales A y R, y B y R de acuerdo con la escala de calificaciones mostrada en la tabla 5.1, puesto que una de las señales de prueba es la misma señal de referencia, una de las dos deberá ser calificada con cinco, mientras que la otra describirá la calidad subjetiva percibida por el sujeto. Para las pruebas se utilizaron 10 muestras de sonido enumeradas y descritas brevemente en la tabla 5.2

Deterioro	Grado
Imperceptible	5.0
Perceptible, pero no molesto	4.0
Ligeramente molesto	3.0
Molesto	2.0
Muy molesto	1.0

Tabla 5.1. Escala de 5 grados de deterioro ITU-R [39]

Abrev.	Nombre de muestra	Procedencia
BAS	Vocal Bass	http://soundexpert.org/sound-samples
CST	Castanets	http://soundexpert.org/sound-samples
FMS	French Male Speech	http://soundexpert.org/sound-samples
GLK	Glockenspiel	http://soundexpert.org/sound-samples
HRP	Harpsichord	http://soundexpert.org/sound-samples
V4	Cuatro estaciones	CD: Las cuatro estaciones, H. Von Karajan
MEC	Las curvas de esa chica	CD: Entre el cielo y el suelo, Mecano
HG	Te quiero	CD: La cagaste Burt Lancaster, Hombres
FS	Something stupid	CD: Nothing but the best, Frank Sinatra

Tabla 5.2. Muestras de audio utilizadas en las pruebas de escucha.

Las pruebas se efectuaron a 10 personas y los resultados se analizaron mediante el método ANOVA de un factor (análisis de varianza), con intervalo de confianza de 95%. Los artefactos documentados en la sección de discusión de resultados, corresponden a los detectados por el autor.

5.2 Resultados perceptuales

A continuación se exponen los resultados de las pruebas perceptuales. Los 3 códec a comparar, a las tasas de bits de 64, 96, 128, 160, 192, 256 y 320 kbps, son:

1. ACC: El códec diseñado e implementado.
2. MP3: Codificador LAME v3.99.5 [33].
3. AAC-LC: Codificador incluido en el software iTunes 12.4.3.1.

En las tablas, de la 5.3 a la 5.9, se muestran los grupos de resultados de las pruebas, para las diferentes muestras de audio descritas en la tabla 5.2. Para cada una de las tablas, en la figura 5.10 se grafican los promedios generales y sus intervalos de confianza de 95%. El eje

vertical de las gráficas corresponde a la escala de la tabla 5.1, y en el eje horizontal se encuentran cada uno de los códec comparados.

Códec	BAS	CST	FMS	GLK	HRP	V4	MEC	HG	FS	General
ACC	3.4	3.1	3.2	3.2	3.3	2.9	3.2	3.1	2.9	3.14
MP3	3.4	3.2	3.3	3.3	3.4	3	3.2	3.1	3.2	3.23
AAC-LC	3.7	3.5	3.7	3.6	3.7	3.6	3.7	3.6	3.3	3.60

Tabla 5.3. Resultados de escucha a 64 kbps

Códec	BAS	CST	FMS	GLK	HRP	V4	MEC	HG	FS	General
ACC	4.1	3.9	4	3.9	4	4.1	4.1	3.7	4	3.97
MP3	4.23	4.17	4.18	4.17	4.16	4.26	4.26	4.19	4.25	4.21
AAC-LC	4.3	4.25	4.23	4.22	4.22	4.44	4.41	4.35	4.4	4.31

Tabla 5.4. Resultados de escucha a 96 kbps

Códec	BAS	CST	FMS	GLK	HRP	V4	MEC	HG	FS	General
ACC	4.4	4.1	4.1	4.3	4.4	4.45	4.46	4.35	4.44	4.33
MP3	4.5	4.41	4.48	4.46	4.47	4.49	4.47	4.44	4.47	4.46
AAC-LC	4.6	4.58	4.59	4.57	4.56	4.6	4.59	4.56	4.58	4.58

Tabla 5.5. Resultados de escucha a 128 kbps

Códec	BAS	CST	FMS	GLK	HRP	V4	MEC	HG	FS	General
ACC	4.66	4.54	4.57	4.58	4.56	4.64	4.61	4.57	4.6	4.59
MP3	4.61	4.56	4.58	4.57	4.57	4.65	4.63	4.59	4.62	4.60
AAC-LC	4.7	4.66	4.69	4.67	4.66	4.7	4.69	4.66	4.67	4.67

Tabla 5.6. Resultados de escucha a 160 kbps

Códec	BAS	CST	FMS	GLK	HRP	V4	MEC	HG	FS	General
ACC	4.7	4.65	4.66	4.67	4.66	4.7	4.68	4.65	4.67	4.67
MP3	4.73	4.66	4.67	4.68	4.68	4.71	4.69	4.66	4.68	4.68
AAC-LC	4.75	4.71	4.7	4.71	4.72	4.74	4.75	4.72	4.74	4.72

Tabla 5.7. Resultados de escucha a 192 kbps

Códec	BAS	CST	FMS	GLK	HRP	V4	MEC	HG	FS	General
ACC	4.83	4.8	4.78	4.76	4.74	4.8	4.77	4.73	4.75	4.77
MP3	4.82	4.8	4.79	4.76	4.75	4.79	4.76	4.72	4.74	4.77
AAC-LC	4.85	4.83	4.84	4.8	4.78	4.81	4.78	4.75	4.76	4.8

Tabla 5.8. Resultados de escucha a 256 kbps

Códec	BAS	CST	FMS	GLK	HRP	V4	MEC	HG	FS	General
ACC	4.89	4.87	4.88	4.89	4.87	4.93	4.92	4.88	4.89	4.89
MP3	4.89	4.88	4.89	4.86	4.88	4.92	4.91	4.85	4.87	4.88
AAC-LC	4.91	4.9	4.92	4.89	4.92	4.95	4.94	4.88	4.91	4.91

Tabla 5.9. Resultados de escucha a 320 kbps

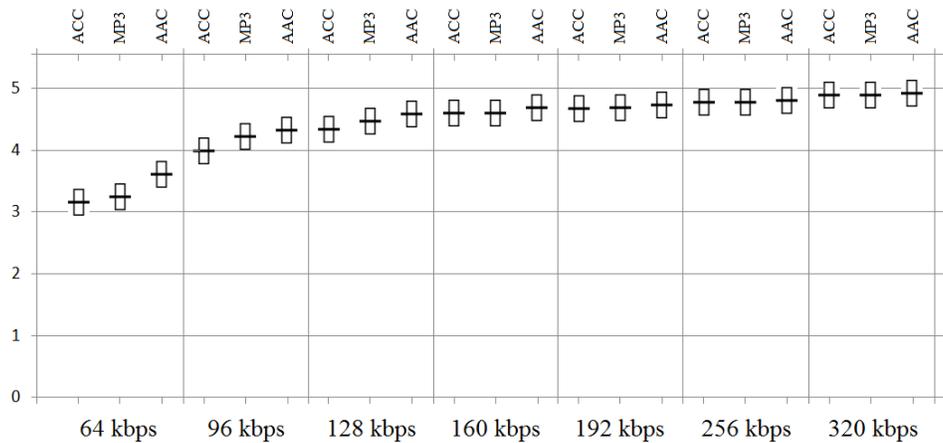


Figura 5.1 Resultados de escucha generales

5.3. Resultados objetivos

La medida de distorsión objetiva utilizada para comparar el desempeño de los códec ACC, MP3 y AAC-LC, es la razón señal a ruido definida mediante la ecuación [3]:

$$\text{SNR(dB)} = 10 \log_{10} \left(\frac{\sum_{n=1}^N (x_n)^2}{\sum_{n=1}^N (x_n - y_n)^2} \right) \quad (5.1)$$

donde x_n es un bloque de N muestras de la señal original, e y_n es el correspondiente bloque de N muestras reconstruidas. En la figura 5.2 se muestra una gráfica de los datos contenidos en la tabla 5.10, correspondientes a los resultados obtenidos mediante la ecuación (5.1), utilizando la pista instrumental “Por la cara”, del CD Descanso dominical del grupo Mecano. En la figura 5.3a, se presenta un bloque de 1024 muestras del canal izquierdo de la pista “Por la cara”.

Códec \ Tasa	AAC-LC (SNR dB)	MP3 (SNR dB)	ACC (SNR dB)
64 kbps	17.1377	14.4069	13.1665
96 kbps	20.8588	17.3451	15.4007
128 kbps	23.4296	18.8771	18.3859
160 kbps	25.4674	21.0839	21.7999
192 kbps	30.1759	24.5445	24.5447
256 kbps	33.0724	30.1725	29.1471
320 kbps	37.8658	35.6428	33.3891

Tabla 5.10. Comparación de SNR entre el ACC, el MP3 y el AAC-LC

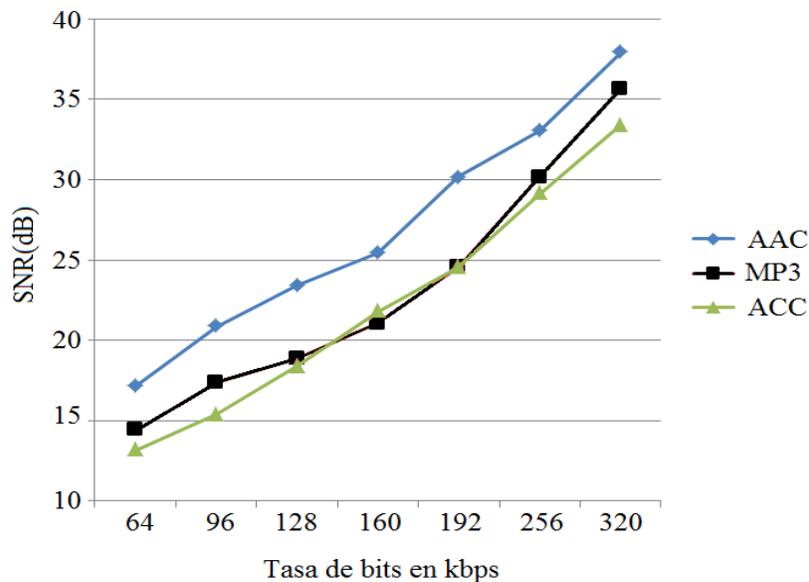


Figura 5.2. Comparación de SNR entre el ACC, el MP3 y el AAC-LC

En cada una de las figuras 5.3b, 5.3c, 5.3d, 5.3e, 5.3f, 5.3g y 5.3h se muestra, respectivamente, para las tasas de bits 64, 96, 128, 160, 256 y 320 kbps, un bloque de 1024 muestras de audio codificado/decodificado mediante el códec ACC, correspondiente al bloque de 1024 muestras de audio mostrado en la figura 5.3a. Nótese que en las figuras 5.3 se muestran superpuestas, tanto la señal decodificada, como la señal de ruido causada por la cuantización de los coeficientes espectrales MDCT. En este caso, puede apreciarse como se va reduciendo el ruido de cuantización, al incrementarse la tasa de bits utilizada.

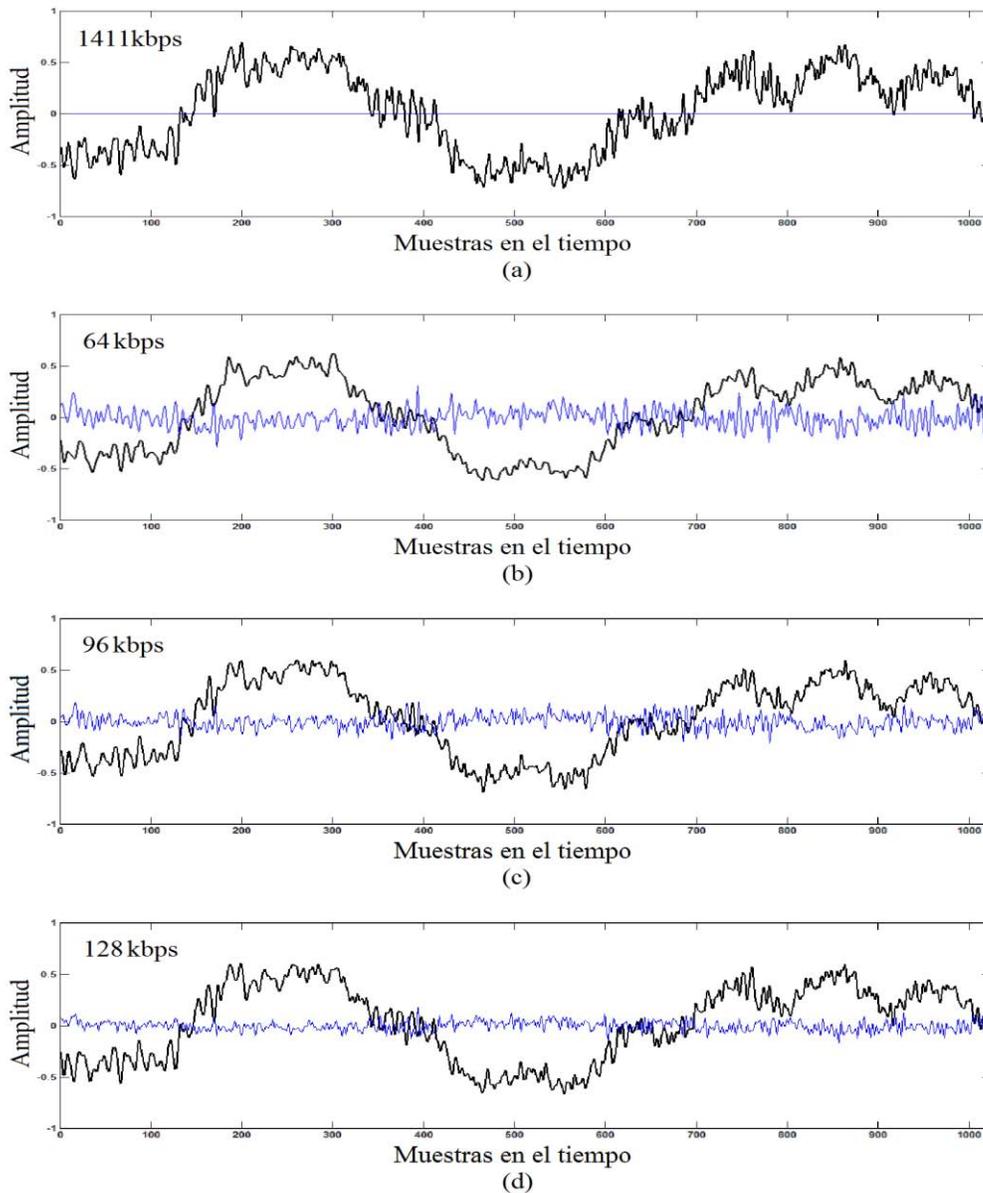


Figura 5.3. Efectos de la compresión utilizando el ACC.

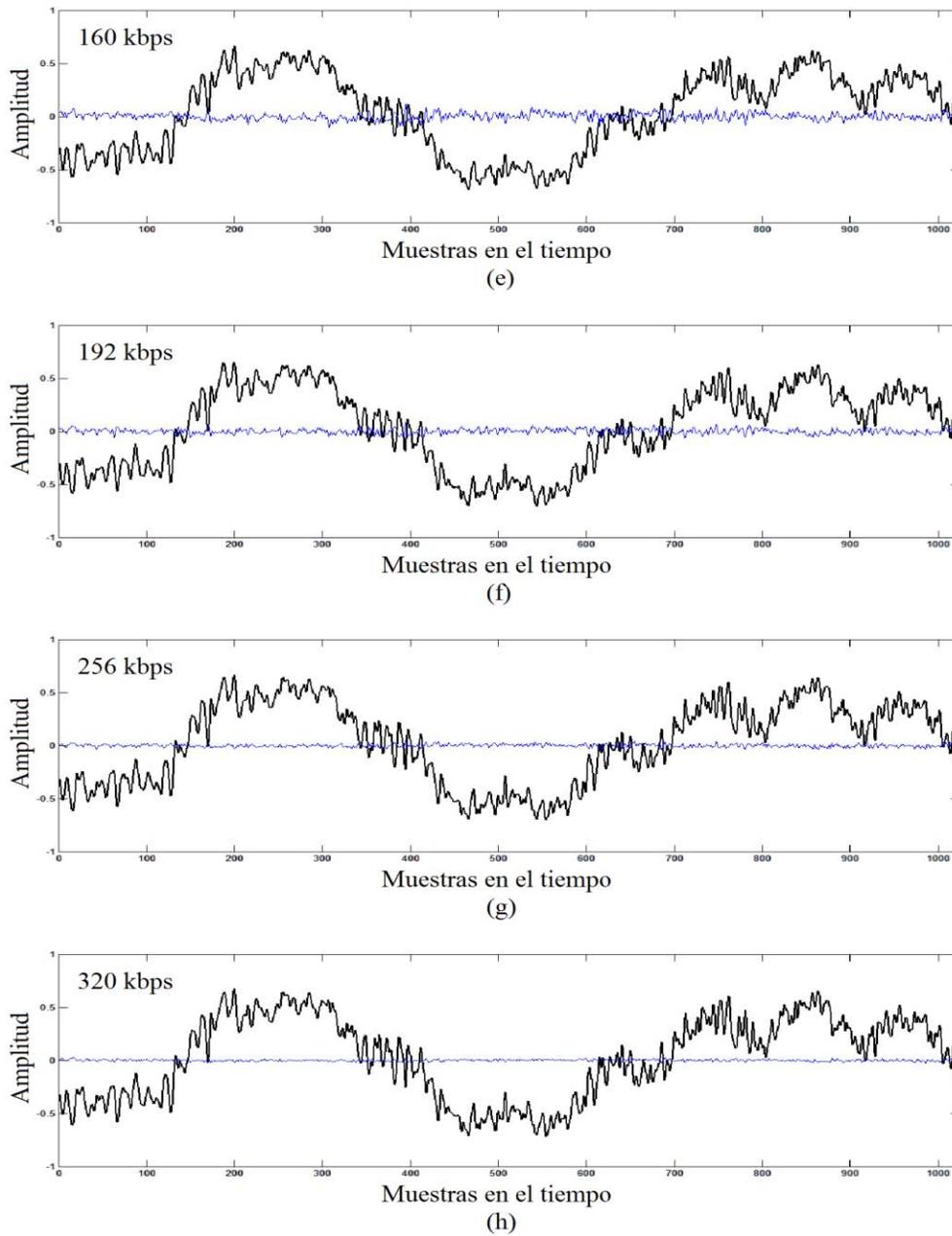
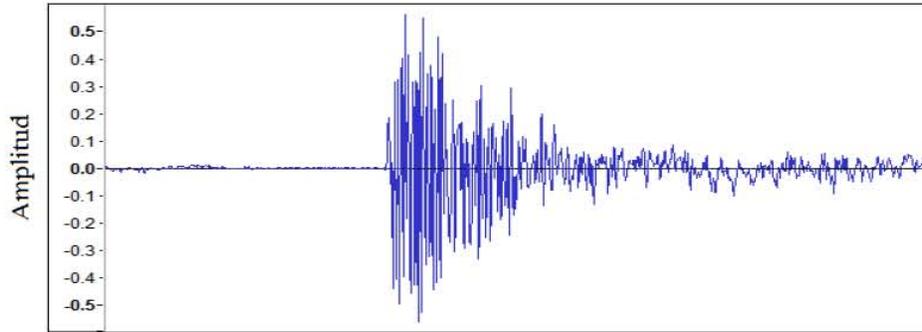


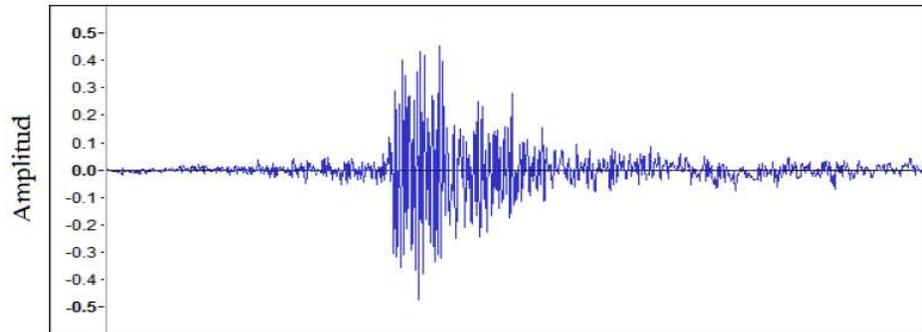
Figura 5.3. Efectos de la compresión utilizando el ACC.

En la figura 5.4, se presenta un ejemplo del comportamiento de la etapa de TNS del códec ACC. En la figura 5.4a se muestra un bloque de 1024 muestras de audio de la señal de prueba CST, descrita en la tabla 5.2. Considerando un proceso de codificación y

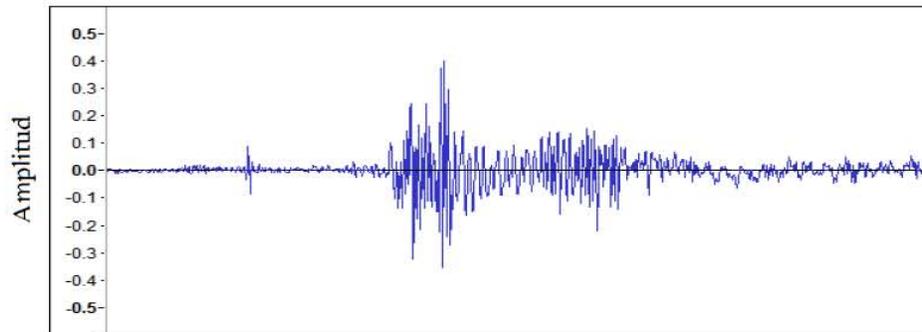
decodificación utilizando el códec ACC a 128kbps, en las figuras 5.4b y 5.4c se muestra, respectivamente, la señal reconstruida sin y con el bloque de TNS activo en el códec ACC.



Muestras en el tiempo
(a)



Muestras en el tiempo
(b)



Muestras en el tiempo
(c)

Figura 5.4.Efecto del bloque TNS. a) Señal original. b) Señal codificada sin el TNS activo. c) Señal codificada con el TNS.

5.4. Discusión de resultados

El códec implementado presenta prácticamente la misma calidad perceptual que el AAC y el MP3, a las tasas de bits de 192 kbps, 256 kbps y 320 kbps. A los 160 kbps el ACC se desempeña casi como el MP3, siendo el AAC el mejor de los tres. Para tasas de bits menores o iguales a 128 kbps el peor desempeño perceptual lo tuvo el códec implementado. A los 128 kbps el ACC sufre de pequeños artefactos debidos principalmente al fenómeno de pre-eco. A los 96 kbps, en el códec ACC se hacen más notorios los artefactos por pre-eco y se comienzan a notar artefactos tipo “birdie” [9]. A los 64 kbps en el ACC se hacen mucho más notorios los artefactos tipo “birdie”, tanto que los artefactos de pre-eco se notan mucho menos.

El principal artefacto en el códec implementado a los 128 kbps es el pre-eco. Éste fue reducido dramáticamente con la inclusión de la técnica conjunta de control de pre-eco y de reservorio de bits. El moldeado del ruido temporal contribuye moderadamente a la reducción del pre-eco para frecuencias mayores a 5500Hz. Si se utiliza el TNS desde el principio del espectro MDCT, se notan distorsiones debidas al TDA inherente a la MDCT.

Capítulo 6

Conclusiones

Gracias a la realización del presente trabajo se adquirió un profundo conocimiento de los principios básicos utilizados en la codificación perceptual de audio. Se adquirió una buena comprensión de la arquitectura empleada por los codificadores de audio MP3 y AAC-LC.

En general, teniendo en cuenta la relativa “baja complejidad” del códec implementado, se puede considerar que éste ofrece una calidad que va de buena a excelente para tasas de bits entre los 160 kbps y 320 kbps. A los 160 kbps se obtuvo la mejor relación entre la razón de compresión y la calidad perceptual alcanzada. Por supuesto, para 192 kbps, 256 kbps y 320 kbps se obtuvo una calidad perceptual muy aproximada a la del codificador LAME, que a su vez tiene un desempeño comparable a aquel del AAC-LC. A los 128 kbps no se logró alcanzar una calidad comparable a la del MP3 ni a la del AAC-LC. Lo anterior es debido, principalmente, a que para mantener baja la complejidad de códec ACC, no se contempló la posibilidad de utilizar ventanas cortas durante segmentos transitorios de la señal de audio. Esto provoca que se siga percibiendo ligeramente el artefacto de pre-eco, a pesar de haber utilizado el control de pre-eco, el reservorio de bits y el TNS. Nótese, sin embargo, que sin la aplicación de dichos métodos para la reducción del pre-eco, este artefacto se torna muy perceptible rayando en lo molesto.

En cuanto al modelo psicoacústico basado en los coeficientes MDCT, se puede considerar que tiene un desempeño comparable al obtenido utilizando el PAM basado en la DFT, y resulta en una notable reducción en la complejidad del códec ACC.

Para construir sobre las bases que sienta el presente trabajo, se propone a futuro:

- La implementación de un mecanismo de conmutación de ventana para que, durante segmentos de señal transitoria, se utilice una ventana de 128 muestras, aumentando así, la resolución temporal del códec durante la ocurrencia de señales transitorias.
- La traducción a aritmética de punto fijo, de las partes del códec programadas en aritmética de punto flotante.
- La implementación del códec ACC en algún DSP, por ejemplo el TMS320C6416.
- La programación de una App para Android, que implemente la etapa de decodificación del ACC.

Apéndice A

Abreviaturas

AAC-LC:	Codificación Avanzada de Audio – Baja Complejidad
ACC:	Audio Códec Conceptual
AC-2:	Códec de Audio Dolby 2
AMD:	Distorsión de amplitud
DCT-IV:	Transformada Discreta de Coseno, tipo IV
DFT:	Transformada Discreta de Fourier
DSP:	Procesador Digital de Señales
ELT:	Transformada Traslapada Extendida
FFT:	Transformada Rápida de Fourier
FIR:	Respuesta al Impulso Finita
IEC:	International Electrotechnical Commission
IMDCT:	Transformada de Coseno Discreta Modificada Inversa
ISO:	International Organization for Standardization
ITU:	International Telecommunication Union
KBD:	Ventana Kaiser-Bessel Derivada
LPC:	Codificación Lineal Predictiva
LSB:	Bit Menos Significativo
LT:	Lapped Transform
MDCT:	Transformada de Coseno Discreta Modificada

MPEG:	Moving Picture Experts Group
MSE:	Error Cuadrático Medio
OLA:	Suma con Traslape
PAM:	Modelo Psicoacústico
PCM:	Modulación por Código de Pulso
PE:	Entropía Perceptual
PQMF:	Pseudo Quadrature Mirror Filter
PR:	Reconstrucción Perfecta
PSD:	Densidad de Potencia Espectral
SBR:	Replicación de Banda Espectral
SFBs:	Bandas de factores de escala
SFM:	Medida de Llanura Espectral
SMR:	Razón Señal a Máscara
SNR:	Razón Señal a Ruido
SPL:	Nivel de Presión de Sonido
TDA:	Aliasing en el Dominio del Tiempo
TDAC:	Cancelación del Aliasing en el Dominio del Tiempo
TBT:	Transformada por Bloques Traslapados
TNS:	Moldeado del Ruido Temporal
VQ:	Cuantización Vectorial

Apéndice B

Demostración de un FB-FIR como TBT

A continuación se demuestra cómo, un banco de filtros FIR (FB-FIR), uniforme máximamente decimado de M canales, puede ser descrito por las ecuaciones 2.11-2.15 en el capítulo 2, como una transformada por bloques traslapados (TBT).

De acuerdo con la figura 2.12, en el banco de análisis, la salida paso-banda de cada filtro $h_k[n]$ se puede expresar como:

$$u_k[n] = x[n] * h_k[n] = \sum_{l=-\infty}^{+\infty} x[l] h_k[n-l]$$

sí los filtros $h_k[n]$ se consideran FIR, causales y de longitud $L = QM$, entonces $h_k[n] \neq 0$ sólo para $n = 0, 1, \dots, L-1$, y la ecuación anterior es equivalente a:

$$u_k[n] = \sum_{l=n-(L-1)}^n x[l] h_k[n-l]$$

aplicando el cambio de variable $l = n - (L - 1) + m$, se obtiene que:

$$u_k[n] = \sum_{m=0}^{L-1} x[n - (L - 1) + m] h_k[(L - 1) - m] \quad (\text{B. 1})$$

sí los filtros $f_k[m]$ en el banco de síntesis se escogen tales que

$$f_k[m] = h_k[(L - 1) - m] \quad (\text{B. 2})$$

entonces la ecuación (B.1) se puede reescribir como:

$$u_k[n] = \sum_{m=0}^{L-1} x[n - (L-1) + m] f_k[m]$$

A partir de la ecuación anterior, decimando por un factor de M, $n = rM$, se obtiene:

$$X_k[r] = u_k[rM] = \sum_{m=0}^{L-1} f_k[m] x[rM - (L-1) + m], \quad k = 0, 1, \dots, M-1 \quad (\text{B.3})$$

Matricialmente B.3, para un instante r en el dominio decimado, se puede expresar como:

$$\begin{bmatrix} X_0[r] \\ X_1[r] \\ \vdots \\ X_{M-1}[r] \end{bmatrix} = \begin{bmatrix} f_0[0] & f_0[1] & f_0[2] & \cdots & f_0[L-1] \\ f_1[0] & f_1[1] & f_1[2] & \cdots & f_1[L-1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{M-1}[0] & f_{M-1}[1] & f_{M-1}[2] & \cdots & f_{M-1}[L-1] \end{bmatrix} \begin{bmatrix} x[rM - (L-1) + 0] \\ x[rM - (L-1) + 1] \\ x[rM - (L-1) + 2] \\ \vdots \\ x[rM] \end{bmatrix} \quad (\text{B.4})$$

En el banco de síntesis, para el k-ésimo canal se tiene que:

$$w_k[n] = \sum_{r=-\infty}^{+\infty} X_k[r] \delta[n - rM] \quad (\text{B.5})$$

De la figura 2.12 se tiene que la k-ésima señal de subbanda $v_k[n]$, interpolada y con aliasing, se puede expresar como:

$$v_k[n] = w_k[n] * f_k[n] = \sum_{l=-\infty}^{+\infty} w_k[l] f_k[n-l]$$

dado que $f_k[n] \neq 0$ solo para $n = 0, 1, \dots, L-1$, la ecuación anterior es equivalente a:

$$v_k[n] = \sum_{l=n-(L-1)}^n w_k[l] f_k[n-l]$$

aplicando el cambio de variable $l = n - (L-1) + i$, se obtiene que:

$$v_k[n] = \sum_{i=0}^{L-1} w_k[n - (L-1) + i] f_k[(L-1) - i]$$

sustituyendo la ecuación (B.5) en la ecuación anterior, resulta que:

$$v_k[n] = \sum_{i=0}^{L-1} \left[\left(\sum_{r=-\infty}^{+\infty} X_k[r] \delta[n - (L-1) + i - rM] \right) f_k[(L-1) - i] \right]$$

o bien,

$$v_k[n] = \sum_{r=-\infty}^{+\infty} \left(X_k[r] \sum_{i=0}^{L-1} \delta[n - (L-1) + i - rM] f_k[(L-1) - i] \right)$$

como $\delta[n - (L-1) + i - rM] = 1$ solo cuando $n - (L-1) + i - rM = 0$, es decir cuando $n - rM = (L-1) - i$, entonces la ecuación anterior es equivalente a:

$$v_k[n] = \sum_{r=-\infty}^{+\infty} X_k[r] f_k[n - rM] \quad (\text{B.6})$$

de la figura 2.12 se tiene que:

$$\hat{x}[n] = \sum_{k=0}^{M-1} v_k[n]$$

sustituyendo la ecuación (B.6) en la ecuación anterior, resulta que:

$$\hat{x}[n] = \sum_{k=0}^{M-1} \left(\sum_{r=-\infty}^{+\infty} X_k[r] f_k[n - rM] \right) = \sum_{r=-\infty}^{+\infty} \left(\sum_{k=0}^{M-1} f_k[n - rM] X_k[r] \right)$$

La ecuación anterior se puede reescribir como:

$$\hat{x}[n] = \sum_{r=-\infty}^{+\infty} y_r[n] \quad (\text{B.7})$$

donde

$$y_r[n] = \sum_{k=0}^{M-1} f_k[n - rM] X_k[r], \quad n = rM + 0, rM + 1, \dots, rM + (L - 1) \quad (\text{B.8})$$

Para una r dada, la ecuación anterior se puede expresar de forma matricial como:

$$\begin{bmatrix} y_r[rM + 0] \\ y_r[rM + 1] \\ y_r[rM + 2] \\ y_r[rM + 3] \\ \vdots \\ y_r[rM + L - 1] \end{bmatrix} = \begin{bmatrix} f_0[0] & f_1[0] & \cdots & f_{M-1}[0] \\ f_0[1] & f_1[1] & \cdots & f_{M-1}[1] \\ f_0[2] & f_1[2] & \cdots & f_{M-1}[2] \\ f_0[3] & f_1[3] & \cdots & f_{M-1}[3] \\ \vdots & \vdots & \ddots & \vdots \\ f_0[L-1] & f_1[L-1] & \cdots & f_{M-1}[L-1] \end{bmatrix} \begin{bmatrix} X_0[r] \\ X_1[r] \\ \vdots \\ X_{M-1}[r] \end{bmatrix} \quad (\text{B.9})$$

y es posible definir la matriz:

$$\mathbf{P} = \begin{bmatrix} f_0[0] & f_1[0] & \cdots & f_{M-1}[0] \\ f_0[1] & f_1[1] & \cdots & f_{M-1}[1] \\ f_0[2] & f_1[2] & \cdots & f_{M-1}[2] \\ f_0[3] & f_1[3] & \cdots & f_{M-1}[3] \\ \vdots & \vdots & \ddots & \vdots \\ f_0[L-1] & f_1[L-1] & \cdots & f_{M-1}[L-1] \end{bmatrix} \quad (\text{B.10})$$

De las ecuaciones B.3, B.4, B.7, B.8, B.9 y B.10 se deriva que el banco de filtros en cuestión, puede ser descrito en el dominio del tiempo mediante las ecuaciones 2.11-2.15 en el capítulo 2.

Apéndice C

Demostración DCT-IV vía DFT

De la ecuación (3.7) se tiene que la DCT-IV de una secuencia $x[n]$, de longitud M , está dada por:

$$X_k = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} x[n] \cos \left[\frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(k + \frac{1}{2} \right) \right] \quad (\text{C. 1})$$

la ecuación anterior es equivalente a:

$$X_{2k} = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} x[n] \cos \left[\frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(2k + \frac{1}{2} \right) \right] \quad (\text{C. 2})$$

$$X_{M-(2k+1)} = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} x[n] \cos \left[\frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(M - (2k + 1) + \frac{1}{2} \right) \right] \quad (\text{C. 3})$$

si se definen las secuencias

$$u[n] = \sqrt{\frac{2}{M}} x[2n]$$

$$v[n] = \sqrt{\frac{2}{M}} x[M - (2n + 1)]$$

entonces la ecuación (C.2) se puede reescribir como:

$$\begin{aligned}
 X_{2k} &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{\pi}{M} \left(2n + \frac{1}{2} \right) \left(2k + \frac{1}{2} \right) \right] \\
 &+ \sum_{n=0}^{M/2-1} v[n] \cos \left[\frac{\pi}{M} \left(M - (2n + 1) + \frac{1}{2} \right) \left(2k + \frac{1}{2} \right) \right] \\
 &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{2\pi}{8M} (4n + 1)(4k + 1) \right] \\
 &+ \sum_{n=0}^{M/2-1} v[n] \cos \left[\left(\frac{\pi}{2} - \frac{2\pi}{8M} (4n + 1) \right) (4k + 1) \right] \\
 &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{2\pi}{8M} (16kn + 4k + 4n + 1) \right] \\
 &+ \sum_{n=0}^{M/2-1} v[n] \cos \left[\frac{\pi}{2} (4k + 1) - \frac{2\pi}{8M} (16kn + 4k + 4n + 1) \right]
 \end{aligned}$$

como $\cos \left(\theta - \frac{\pi}{2} (4p + 1) \right) = \text{sen}(\theta)$, siendo p un entero, entonces

$$\begin{aligned}
 X_{2k} &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{2\pi}{8M} (16kn + 4k + 4n + 1) \right] \\
 &+ \sum_{n=0}^{M/2-1} v[n] \text{sen} \left[\frac{2\pi}{8M} (16kn + 4k + 4n + 1) \right]
 \end{aligned}$$

como $e^{j\theta} = \cos(\theta) + j\text{sen}(\theta)$, entonces

$$X_{2k} = \text{Re} \left\{ \sum_{n=0}^{M/2-1} (u[n] + jv[n]) e^{-j\frac{2\pi}{8M}(16kn+4k+4n+1)} \right\}$$

o bien

$$X_{2k} = \operatorname{Re} \left\{ e^{-j\frac{2\pi}{2M}k} \sum_{n=0}^{M/2-1} c[n] e^{-j\frac{2\pi}{M/2}kn} \right\}$$

donde

$$c[n] = \sqrt{\frac{2}{M}} (x[2n] + jx[M - (2n + 1)]) e^{-j\frac{2\pi}{8M}(4n+1)}$$

además, la ecuación (C.3) se puede reescribir como:

$$\begin{aligned} X_{M-(2k+1)} &= \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} x[n] \cos \left[\frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(M - \left(2k + \frac{1}{2} \right) \right) \right] \\ &= \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} x[n] \cos \left[\left(\pi \left(n + \frac{1}{2} \right) - \frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(2k + \frac{1}{2} \right) \right) \right] \\ &= \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} x[n] \cos \left[\left(\frac{\pi}{2} (2n + 1) - \frac{2\pi}{8M} (2n + 1)(4k + 1) \right) \right] \end{aligned}$$

$$\begin{aligned} X_{M-(2k+1)} &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{\pi}{2} (4n + 1) - \frac{2\pi}{8M} (4n + 1)(4k + 1) \right] \\ &\quad + \sum_{n=0}^{M/2-1} v[n] \cos \left[\frac{\pi}{2} (2M - (4n + 1)) - \frac{2\pi}{8M} (2M - (4n + 1))(4k + 1) \right] \\ &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{\pi}{2} (4n + 1) - \frac{2\pi}{8M} (4n + 1)(4k + 1) \right] \\ &\quad + \sum_{n=0}^{M/2-1} v[n] \cos \left[\pi M - \frac{\pi}{2} (4n + 1) - \frac{2\pi}{8M} (2M - (4n + 1))(4k + 1) \right] \end{aligned}$$

como $\cos(\theta - \pi M) = \cos(\theta)$ cuando M es par, entonces:

$$\begin{aligned}
 X_{M-(2k+1)} &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{\pi}{2} (4n+1) - \frac{2\pi}{8M} (4n+1)(4k+1) \right] \\
 &+ \sum_{n=0}^{M/2-1} v[n] \cos \left[\frac{\pi}{2} (4n+1) + \frac{2\pi}{8M} (2M - (4n+1))(4k+1) \right] \\
 &= \sum_{n=0}^{M/2-1} u[n] \cos \left[\frac{\pi}{2} (4n+1) - \frac{2\pi}{8M} (4n+1)(4k+1) \right] \\
 &+ \sum_{n=0}^{M/2-1} v[n] \cos \left[\frac{\pi}{2} (4n+1) + \frac{\pi}{2} (4k+1) - \frac{2\pi}{8M} (4n+1)(4k+1) \right]
 \end{aligned}$$

como $\cos\left(\theta - \frac{\pi}{2}(4p+1)\right) = \text{sen}(\theta)$ y $\text{sen}\left(\theta - \frac{\pi}{2}(4p+1)\right) = -\text{cos}(\theta)$, siendo p un entero, entonces

$$\begin{aligned}
 X_{M-(2k+1)} &= \sum_{n=0}^{M/2-1} u[n] \text{sen} \left[\frac{2\pi}{8M} (4n+1)(4k+1) \right] \\
 &- \sum_{n=0}^{M/2-1} v[n] \cos \left[\frac{2\pi}{8M} (4n+1)(4k+1) \right]
 \end{aligned}$$

como $e^{j\theta} = \cos(\theta) + j\text{sen}(\theta)$ entonces

$$X_{M-(2k+1)} = -\text{Im} \left\{ e^{-j\frac{2\pi}{2M}k} \sum_{n=0}^{M/2-1} c[n] e^{-j\frac{2\pi}{M/2}kn} \right\}$$

donde $c[n] = \sqrt{\frac{2}{M}} (x[2n] + jx[M - (2n+1)]) e^{-j\frac{2\pi}{8M}(4n+1)}$

Apéndice D

PAM: un ejemplo gráfico

A continuación se presentan de forma gráfica los cálculos realizados en el PAM, para el espectro MDCT mostrado en la figura D.1.

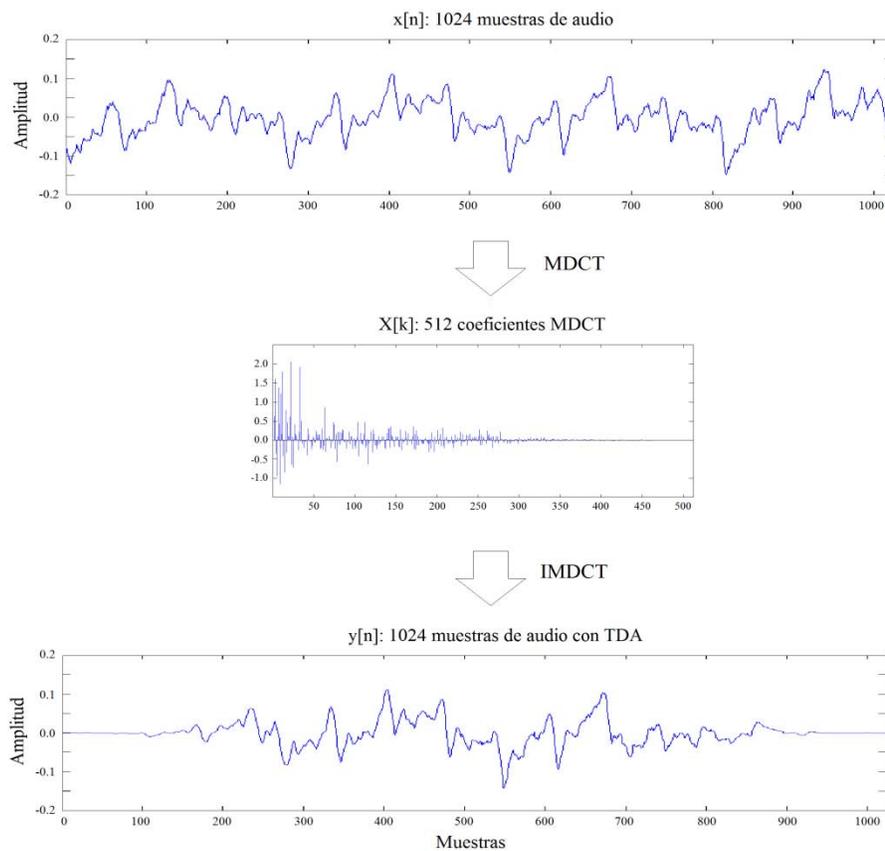


Figura D.1. Espectro MDCT de un bloque de 1024 muestras de audio.

En la figura D.2 se muestra la partición del espectro MDCT en 63 componentes enmascaradores, de acuerdo con los datos en la tabla 4.1.

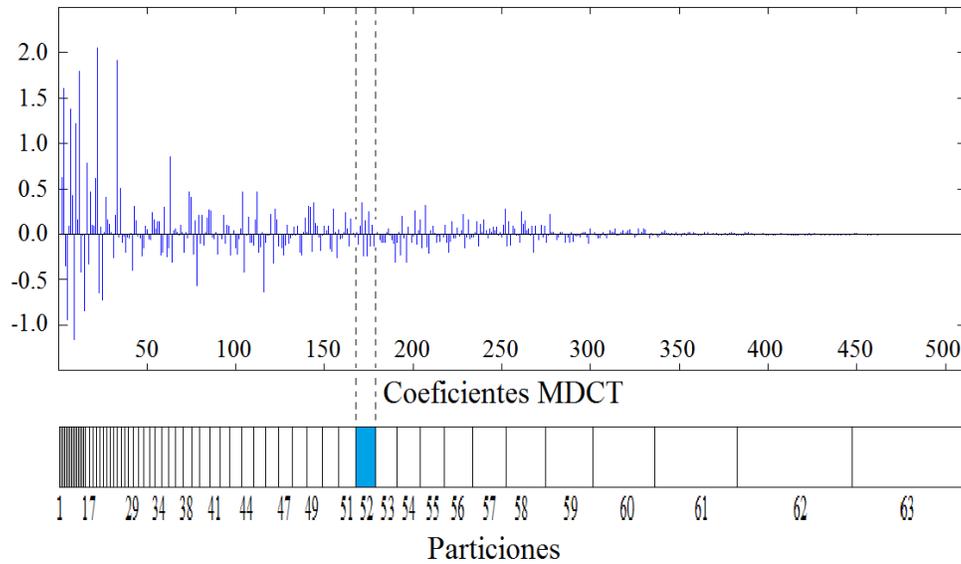


Figura D.2. 63 Particiones del espectro MDCT.

En la figura D.3 se presenta el espectro particionado de energía obtenido mediante la ecuación (4.2). En esta figura, se resalta el componente enmascarador número 52 y se indica con una línea punteada su correspondiente umbral de enmascaramiento.

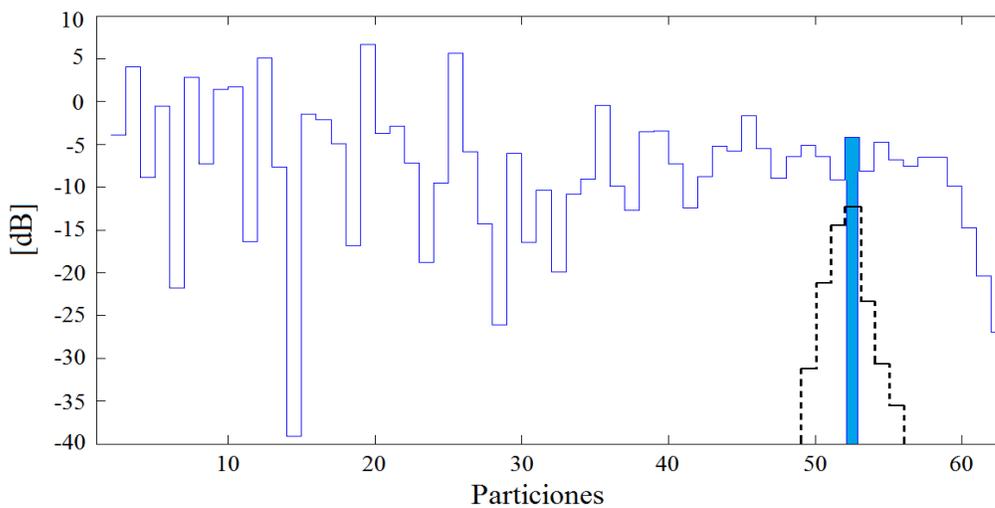


Figura D.3. Espectro particionado de energía.

En la figura D.4 se presenta el umbral global de enmascaramiento obtenido utilizando la ecuación (4.4). Este umbral es una combinación de los umbrales de enmascaramiento de los 63 componentes enmascaradores mostrados en la figura D.3.

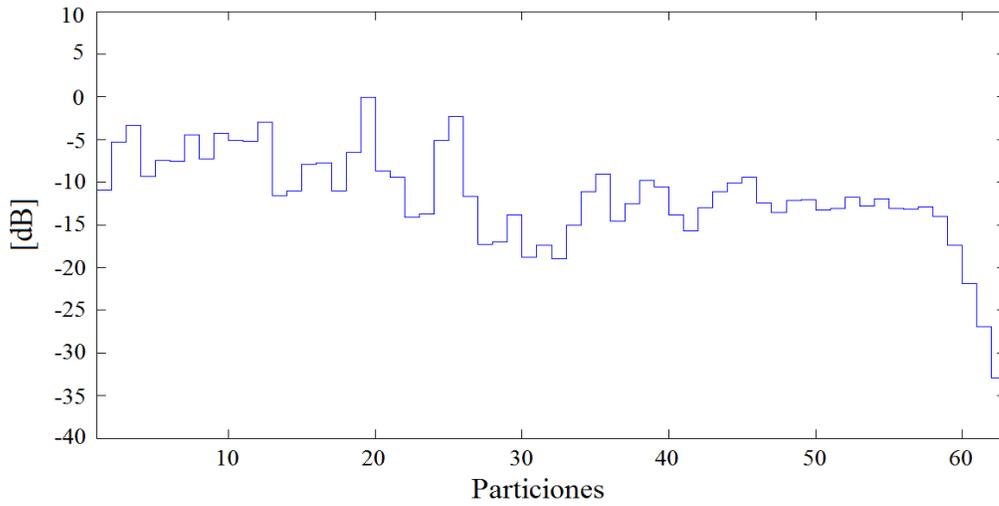


Figura D.4. Umbral de enmascaramiento.

En la figura D.5 se muestra el umbral auditivo ATHmin utilizado en la ecuación (4.6).

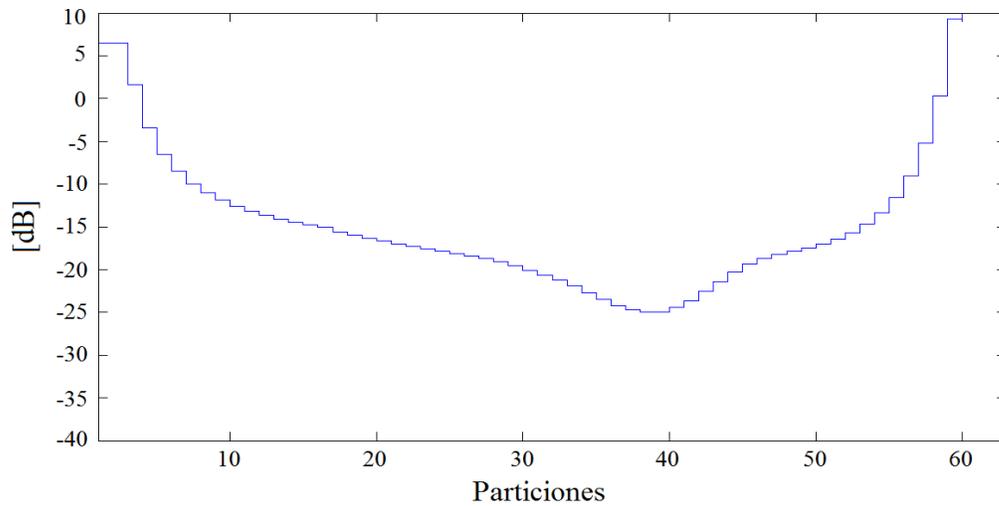


Figura D.5. Umbral auditivo ATHmin

En la figura D.6 se muestra el umbral global de enmascaramiento según la ecuación (4.6), considerando sólo el umbral de enmascaramiento actual.

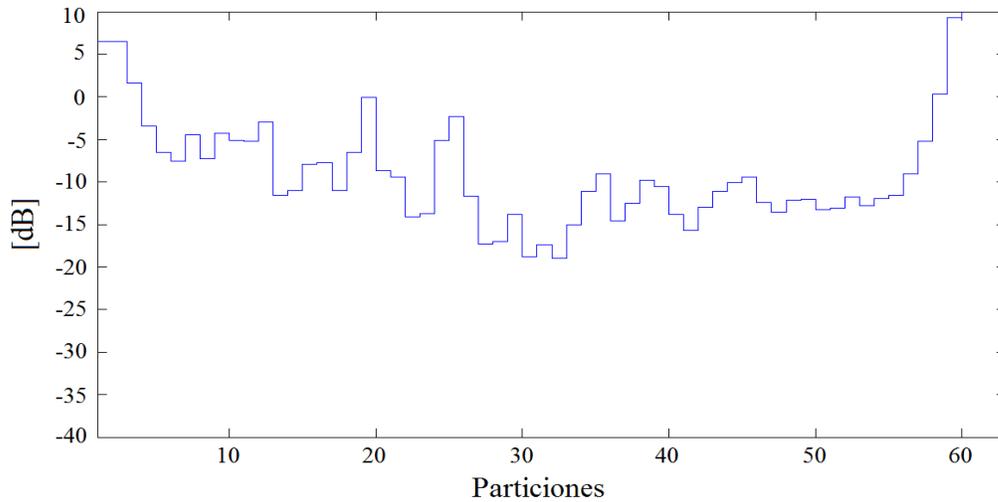


Figura D.6. Umbral global de enmascaramiento

Finalmente, en la figura D.7 se muestra el umbral global de energía de enmascaramiento obtenido mediante la ecuación (4.7).

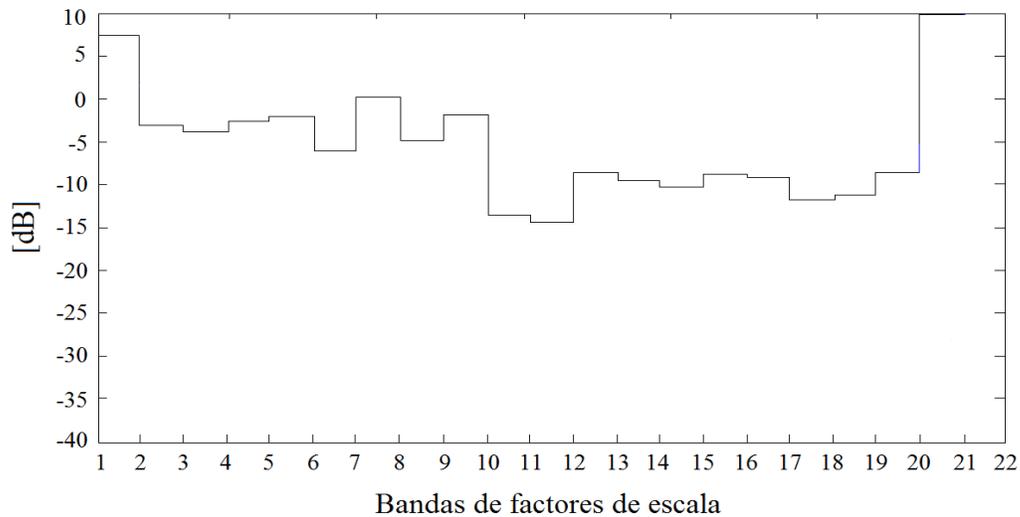


Figura D.7. Umbral global de energía de enmascaramiento.

Bibliografía

- [1] K. Brandenburg, "MP3 and AAC Explained," in Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding, August 1999.
- [2] M. Kahrs y K. Brandenburg, "Applications of Digital Signal Processing to Audio and Acoustics", Kluwer Academic Publishers, 2002.
- [3] K. Sayood, "Introduction to Data Compression", Morgan Kaufmann, 3a edición 2006.
- [4] Vetterli, Martin and Kovacevic, Jelena. "Wavelets and Subband Coding". Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [5] E. Zwicker and H. Fastl, "Psychoacoustics Facts and Models", Springer-Verlag, 3a edición 2006.
- [6] M. Bosi and R. Goldberg, "Introduction to Digital Audio Coding and Standards", Kluwer Academic Publishers, 2003.
- [7] Lin, Yiqing, Abdulla Waleed H, "Audio Watermark", Springer International Publishing, 2015.
- [8] J. Herre, J. D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)", 101st AES convention, Los Angeles 1996, Preprint 4384
- [9] Chi-Min Liu, Han-Wen Hsu, and Wen-Chieh Lee, "Compression Artifacts in Perceptual Audio Coding", IEEE Transactions On Audio, Speech, And Language Processing, Vol. 16, No. 4, May 2008.
- [10] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," Proc. IEEE, vol. 78, pp. 56-93, Jan. 1990.
- [11] Rabiner, Lawrence: "Unified approach to short-time Fourier analysis and synthesis", Proceedings of the IEEE, Vol. 65, NO. 11, November 1977.

- [12] Malvar. H., –Signal Processing with Lapped Transforms”, Artech House, Inc., 1992.
- [13] J. H. Rothweiler, –Polyphase quadrature filters, a new subband coding technique,” in Proc. /EEC Int. Conf. on ASSP, Boston, MA, Apr. 1983, pp. 1980-1983
- [14] Malvar. H., –Modulated QMF filter banks with perfect reconstruction,” Electron. Lett., vol.26, no. 13, pp. 906-907, June 1990.
- [15] Malvar. H., –Extended Lapped Transforms: Properties, Applications, and Fast Algorithms”, IEEE Transactions On Signal Processing, Vol. 40, No. 11, November 1992.
- [16] J. Princen and A. Bradley, –Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation,” IEEE Trans. ASSP, pp. 1153-1161, Oct. 1986.
- [17] J. Princen, A. Johnson and A. Bradley, –Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation,” in Proc. Int. Conf. Acous., Speech, and Sig. Process. (ICASSP-87), pp. 50.1.1-50.1.4, May 1987.
- [18] Malvar. H., –Lapped Transforms for Efficient Transform/Subband Coding,” IEEE Trans. Acous., Speech, and Sig. Process., Vol. 38, No. 6, pp. 969-978, Jun. 1990.
- [19] M. H. Cheng and Y. H. Hsu, –Fast IMDCT and MDCT algorithms—A matrix approach,” IEEE Trans. Signal Process., vol. 51, no. 1, pp. 221–229, Jan. 2003.
- [20] C. R. Johnson, –Matrix Theory and Applications”, American Mathematical Society, 1990.
- [21] Z. Wang, –Fast algorithms for the discrete W transform and for the discrete Fourier transform,” IEEE Trans. Acoust., Speech, Signal Processing 32 (4) (1984) 803–816.
- [22] R. Gluth, –Regular FFT-related transform kernels for DCT/DST-based polyphase filter banks,” in Proc. Int. Conf. Acous., Speech, and Sig. Process. (ICASSP-91), Vol. 3, pp. 2205-2208, Apr 1991.
- [23] C. Jing, H.-M. Tai, –Fast algorithm for computing modulated lapped transform,” Electronic Letters vol. 37, no. 12, pp. 796–797, June 2001.
- [24] Yin-Tsung Hwang and Shin-Chi Lai, –A novel MDCT/IMDCT computing kernel design”, IEEE Workshop on Signal Processing Systems Design and Implementation, Nov. 2005, pp. 526-531.
- [25] Xuancheng Shao, Steven G. Johnson, –Type-IV DCT, DST, and MDCT algorithms with reduced numbers of arithmetic operations,” Elsevier Signal Processing vol. 88, no. 6, pp. 1313–1326, June 2008.

- [26] D. Pan, "A Tutorial on MPEG/Audio Compression," IEEE Mult. Med., pp. 60-74, Sum. 1995.
- [27] Yuichiro Takamizawa, Toshiyuki Nomura, Masao Ikekawa, "High quality and processor-efficient implementation of an MPEG-2 AAC encoder," in Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol.2, Page(s): 985-899.
- [28] Shih-Way Huang, Tsung-Hun Tsai, Liang-Gee Chen, "A low complexity design of psycho-acoustic model for MPEG-2/4 advanced audio coding" IEEE Transactions on Consumer Electronics, Volume 50, Issue 4, Nov. 2004 Page(s): 1209-1217.
- [29] S.-W. Huang, T.-H. Tsai and L.-G. Chen, "Memory Reduction Technique of Spreading Function in MPEG AAC Encoder," 7th International Conference on Digital Audio Effects, Oct., 2004.
- [30] Shih-Way Huang; Liang-Gee Chen; Tsung-Han Tsai; "Memory and Computationally Efficient Psychoacoustic Model for MPEG AAC on 16-bit Fixed-point Processors" Circuits and Systems, 2005. ISCAS 2005. Symposium on IEEE International 23-26 May 2005 Page(s):3 155 -3158.
- [31] J. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," IEEE J. Sel. Areas in Comm., pp. 314-323, Feb. 1988.
- [32] J.J. Thiagarajan and A. Spanias, "Analysis of the MPEG-1 Layer III (MP3) Algorithm Using MATLAB", Morgan & Claypool, 2012.
- [33] LAME versión 3.99.5 de 2012-02-28, Disponible en: <http://lame.sourceforge.net/>.
- [34] E. Alexandre, A. Pena, and M. Sobreira, "Low-Complexity Bit-Allocation Algorithm for MPEG AAC Audio Coders," IEEE Signal Processing Letters, Dec. 2005, pp. 824-82.
- [35] B. Wang, J. Zhang, Y. Yao, and L. Xie, "A New Bit-allocation Algorithm for AAC Encoder Based on Linear Prediction," in Proc. of the IEEE 11th ICCT'08, pp. 726-729, Hangzhou, Nov. 2008.
- [36] Han-Wen Hsu and Chi-Min Liu, "Autoregressive modeling of temporal/spectral envelopes with finite-length discrete trigonometric transforms," IEEE Trans. Signal Process., vol. 58, no. 7, pp. 3692-3705, 2010.
- [37] J. Makhoul, "Linear prediction: A tutorial review," Proc. IEEE, vol. 63, no. 4, pp. 561-580, Apr. 1975.

[38] Y. Tohkura and F. Itakura, "Spectral sensitivity analysis of PARCOR parameters for speech data compression," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 3, pp. 273–280, June 1979.

[39] International Telecommunications Union, Radiocommunication Sector BS.1116-3, "Methods for the subjective assessment of small impairments in audio systems", Geneva 2015.