

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO



INSTITUTO DE CIENCIAS BÁSICAS E  
INGENIERÍA



PROCESAMIENTO DIGITAL DE SEÑALES DE AUDIO EN  
TIEMPO REAL

TESIS

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y  
TELECOMUNICACIONES

PRESENTA:

HERNÁNDEZ PÉREZ MIGUEL ÁNGEL

ASESOR:

DR. ENRIQUE ESCAMILLA HERNÁNDEZ

*Pachuca, Hidalgo., Febrero 2009.*

*Agradezco primeramente a Dios por ser mi  
compañero... mi fortaleza, por darme todo lo  
que tengo y no dejarme caer nunca.*

*A mi familia, a mi mamá y a mi papá por ser los  
mejores y estar conmigo incondicionalmente,  
gracias porque sin ellos y sus enseñanzas no  
estaría aquí ni sería quien soy ahora, a ellos les  
dedico esta tesis. En todo momento los llevo  
conmigo.*

*Al Doctor Enrique Escamilla Hernández por  
asesorarme a lo largo de la tesis y  
acompañarme en este camino que hoy culmina  
en el presente proyecto, por compartirme su  
conocimiento e inspirar mi admiración.*

*Y en especial a Karina Ruby por su  
incondicional apoyo.*

# RESUMEN

Los avances en el Procesamiento Digital de Señales de Audio están en constante cambio, ya que es una de las ramas de la electrónica que en la actualidad tienen mayor rentabilidad, debido a que se posiciona dentro de las características de la cultura mundial, ahora traer consigo un MP3 o cualquier tipo de reproductor es parte de la vida cotidiana.

Aquí se realiza una de las etapas del procesamiento digital de una señal de audio que utiliza y aprueba el MPEG 1 (Moving Picture Expert Group) en sus versiones Layer 3 (MP3) y Advanced Audio Coding (AAC) enfocados a un DSP TMS320C6727 con características de investigación básica, bajo la plataforma de programación de Matlab, donde se desarrolla un algoritmo, el mismo que se puede instalar en kits tales como el PADK.

Este trabajo consiste en la investigación y captura de ventanas para analizarlas con funciones atómicas, y así obtener la función adecuada que se complementó con la siguiente etapa, en la cual se implementa la Transformada Discreta Coseno Modificada (MDCT) y su Inversa (IMDCT). Ambas etapas fueron desarrolladas en un algoritmo que trabaja en la plataforma Matlab 7.0.

Tras pruebas realizadas se obtuvieron buenos resultados, los cuales nos muestran la etapa de la transformación que es la más importante en el estándar de MPEG.

Finalmente se realiza un estudio al Professional Audio Development Kit (PADK) de Lyrtech, Texas Instruments para entender la implementación del algoritmo creado en Matlab, a un DSP.

# Índice General

Resumen	I
Índice General	II
Índice de Figuras	IV
Índice de Tablas	VI
Planteamiento del Problema	VII
Objetivos de la Tesis	VIII
Justificación	IX

## Capítulo 1 Introducción

1.1 Introducción	1
1.2 Problemas de Almacenamiento	4

## Capítulo 2 Antecedentes

2.1 Del Audio Analógico al Digital	7
2.2 Moving Picture Layer III	12
2.3 Advanced Audio Coding	14
2.4 Estándares de Compresión	17
2.4.1 Estándares de Compresión sin Pérdida	18
2.4.2 Estándares de compresión con Pérdida	19

## Capítulo 3 Herramientas de Procesamiento Digital de Señales

3.1 Filtros Digitales	20
3.1.1 Filtros FIR	22
3.1.2 Filtros IIR	23
3.2 Muestreo	25
3.3 Modelo Psicoacústico	28
3.3.1 Umbral Absoluto de Audición	30
3.3.2 Intensidad Sonora, Tono y Timbre	31
3.3.3 Bandas Críticas	32
3.3.4 Enmascaramiento	34
3.4 Codificación	37
3.4.1 Codificación Perceptual	40
3.5 Algoritmo para la Transformada Rápida de Fourier (FFT)	41
3.6 Transformada Discreta de Coseno Modificada	46
3.7 Enventanado	48
3.8 Encabezado de Tramas	52

## Capítulo 4 Resultado de Procesamiento Digital de Señales

Resultado de Procesamiento Digital de Señales	57
4.1 Fase de Enventanado	59
4.2 Fase de Compresión con la MDCT	62
4.2.1 Análisis y Síntesis utilizando la MDCT	62
Conclusiones	72
Apéndice A: Professional Audio Development Kit	73
Apéndice B: Código Fuente	96
Referencias Bibliográficas	
Acrónimos	

# Índice de Figuras

Figura 3.1 Umbral absoluto de audición	30
Figura 3.2 Distribución de frecuencias dentro de la cóclea	34
Figura 3.3 Efecto de enmascaramiento de dos tonos	35
Figura 3.4 Ejemplo del efecto pre-enmascarador y post-enmascarador	36
Figura 3.5 Función $up(x)$	52
Figura 3.6 Espectro de ventana $up(x)$	52
Figura 3.6 Encabezado de trama	52
Figura 4.1 Ventana Hanning	59
Figura 4.2 Espectro de ventana Hanning	59
Figura 4.3 Ventana de Hamming	60
Figura 4.4 Espectro de ventana de Hamming	60
Figura 4.5 Ventana Kaiser-Bessel, $\alpha = 3$	60
Figura 4.6 Espectro de ventana Kaiser-Bessel	60
Figura 4.7 Función $up(x)$	61
Figura 4.8 Espectro de ventana $up(x)$	61
Figura 4.9 Ventana $up(x)$ y su inversa	61
Figura 4.10 Señal de prueba original	62
Figura 4.11 Bloque de 512 de la señal original	63
Figura 4.12 Bloque de la señal original multiplicado por la ventana $up(x)$	63
Figura 4.13 MDCT (de un bloque)	63
Figura 4.14 IMDCT recuperada	64
Figura 4.15 Reconstrucción de la señal con cancelación de aliasing en el dominio del tiempo	65
Figura 4.16 Bloque de 512 muestras	65
Figura 4.17 Multiplicación del bloque por la ventana $up(x)$	66
Figura 4.18 MDCT del bloque multiplicado por la ventana	66
Figura 4.19 IMDCT sin traslape del 50% ni suma	67
Figura 4.20 Reconstrucción de la señal original de prueba 2	67
Figura 4.21 Bloque de 512 muestras	68
Figura 4.22 Bloque multiplicado por la ventana $up(x)$	68
Figura 4.23 MDCT de un bloque de 512 muestras	69
Figura 4.24 IMDCT del bloque de 512 muestras	69
Figura 4.25 Señal recuperada	70
Figura 4.26 Señal original	70
Figura A.1 Setup CCStudio	78



Figura A.2 Create Board	79
Figura A.3 Tarjeta DSP modelo TMS320C6727	79
Figura A.4 Propiedades del procesador	80
Figura A.5 Entradas y las salidas del PADK TMS320C6727	83
Figura A.6 Memorias en el DSP	84
Figura A.7 Parte frontal de PADK	85
Figura A.8 Parte frontal	85
Figura A.9 Interior del PADK	86
Figura A.10 Placa física del PADK	89
Figura A.11 Componentes de entradas salidas del PADK	89
Figura A.12 Puertos de entrada salida	91
Figura A.13 MP3/Reverb	95

# Índice de Tablas

Tabla 3.1 Descripción de capa	53
Tabla 3.2 Descripción de capa <b>bitrate_index</b>	53
Tabla 3.3 Tasa de muestreo	54
Tabla 3.4 Modo de canal	54
Tabla 3.5 Código para capa III	55
Tabla 3.6 Codificación de énfasis	56

## Planteamiento del Problema

Cómo comprimir un archivo de audio, es la interrogante que, durante los últimos años ha tenido ocupado a una gran parte de la industria de la música, pues en general el acceso a la información no es un lujo, sino una necesidad, la portabilidad de los archivos es algo cotidiano y la transferencia de los mismo es una herramienta tanto de trabajo como de entretenimiento. Las limitaciones tecnológicas y económicas han impulsado la búsqueda de formas más eficientes de comprimir una señal.

La transferencia de archivos de audio y video es cada día más común, sin embargo, la información multimedia demanda grandes cantidades de almacenamiento y procesamiento. Por lo tanto es necesario mejorar la forma en la que este contenido se almacena para así realizar transferencias de audio cada vez más rápidas y de mejor calidad.

## Solución

Como solución a este problema se creó e implemento en MATLAB un código de transformación basado en la Transformada Discreta Coseno Modificada (MDCT), una transformada ampliamente usada en codificadores perceptuales. La MDCT se aplica a bloques de 512 muestras. Posteriormente se efectuó la implementación en el mismo programa de una ventana  $w_p(x)$  con funciones atómicas, para realizar un traslape en un 50%, y suma de los bloques adyacentes, que al momento de realizar la Transformada Discreta Coseno Modificada Inversa (IMDCT) también creada y ejecutada en MATLAB se obtenga la reconstrucción exacta de la señal original. Ya que la MDCT es una herramienta de compresión que resuelve en gran manera este problema debido a que presenta una máxima tasa de compresión y permite una reconstrucción total de la señal de audio. Además la MDCT es utilizada y aprobada por el MPEG 1 en su capa 3, para así estructurar al formato MP3.

# Objetivos de la Tesis

## Objetivo General

Procesar digitalmente una señal de audio mediante un algoritmo que proporcione una reconstrucción exacta, diseñado en MATLAB, el cual podrá ser implementado en un DSP para realizar el proceso en tiempo real, con las características necesarias del formato MP3 y AAC.

## Objetivos Particulares

- ◆ Obtener una ventana de síntesis válida para realizar adecuadamente el procesamiento digital.
- ◆ Desarrollar un algoritmo que realice la compresión de la señal.
- ◆ Desarrollar tanto la ventana como la transformada adecuada para lograr la reconstrucción total de la señal.
- ◆ Aplicar dichas funciones a un programa creado en Matlab para el procesamiento de una señal.
- ◆ Utilizar un DSP que cumpla con las características necesarias para audio.

## Justificación

La información multimedia, por su propia naturaleza, demanda grandes cantidades de almacenamiento y de procesamiento. Por lo tanto, es necesario perfeccionar la forma en la que este contenido se almacena para así minimizar el ancho de banda necesario para completar dicha transferencia. Es necesario entonces, encontrar esquemas para reducir la cantidad de información necesaria para generar un contenido con la calidad que demanda la aplicación.

Esta tesis así como todo avance tecnológico se fundamenta en varias etapas: investigación básica, aplicada, de desarrollo y producción. Y aquí se combinan las principales características de la investigación básica como de la investigación aplicada, que desarrollan estrategias de modelización y compresión de la señal de audio, las cuales podrían ser útiles en muchas aplicaciones.

En consecuencia, se busca un modelo de codificación aritmética más apropiado para audio, pues el actual comprime los coeficientes de transformación con un modelo que empieza en una distribución uniforme. Incluso, es posible que un modelo de orden mayor sea una buena opción, por lo que se estudia el transformado coseno, para que así se minimice la velocidad de transferencia de archivos de audio conforme al procesamiento digital que almacena grandes cantidades de audio en espacios reducidos.

# Capítulo 1

## Introducción

### 1.1 Introducción

A partir del inicio de la era computacional, las exigencias han sido cada vez mayores, de manera tal que todas las cosas de la vida cotidiana se han ido implementando en la computadora. En estos momentos es posible mediante las tecnologías de la información computacional, escuchar música, ver videos, comunicarnos vía telefónica, entre otras.

Una de las cosas que más preocupa a los usuarios de computadoras, es la capacidad de los dispositivos de almacenamiento, es decir, el espacio que ocupa cada archivo en el computadora, y esto con la finalidad de conservar la integridad de la calidad que se maneja en cualquier organización a nivel de computadoras, ya sean personales o de trabajo, tomando en cuenta la capacidad de almacenamiento de los programas de aplicaciones que cada día ocupan mayor espacio [1].

En la actualidad se han hecho varios intentos por resolver el mito de cómo comprimir estos archivos al máximo, en este caso la compresión de sonido digital y mantener una alta fidelidad al momento de reproducirlo, sin embargo esa calidad todavía no había sido logrado hasta la aparición de nuevos formatos de audio.

De aquí el origen de este trabajo que se basa en la música digital y los formatos que la comprimen. Uno de ellos es el formato MP3, con este archivo podemos conseguir una calidad de reproducción igual a la obtenida con un disco compacto. ¿Y cuál es el problema? la cantidad de espacio que ocupan estos archivos. Por medio de compresores, se consigue reducir el tamaño de estos archivos de una forma exponencial y así, con el formato comprimido MP3, podemos tener en un poco más de tres Megabytes en una canción de tres minutos de nuestro cantante preferido y con calidad de un CD [2]. Ya que el MP3 es un formato de compresión de audio que consigue una razón de compresión de 1:10 o hasta 1:15 (dependiendo el tipo de algoritmo a utilizar), sin pérdida de calidad apreciable, significa que en un disco compacto se pueden grabar unos 11 CD's de audio (aprox. unas 150 canciones).

Este nuevo nivel de compresión de audio es en realidad moderno, es decir se ha venido desarrollando en los últimos años; pero la historia comienza en el corazón de Europa durante los años 80 cuando Karlheinz Branderburg era un joven aspirante a doctorado en la Universidad alemana de Earlengen-Nuremberg y trataba de desarrollar un nuevo algoritmo de compresión de sonido. Un profesor de dicha Universidad, Prof. Dieter Seitzer, impulsó al joven Branderburg a resolver el problema de transmitir música de alta fidelidad por una línea telefónica utilizando su nuevo algoritmo de compresión como trabajo de investigación para su tesis doctoral. No era un problema de computación, ni siquiera de tecnología (desde finales de los 70 había llegado la fibra óptica y con ella las primeras líneas ISDN para transmisión digital), era más un problema científico: ¿cómo percibe el oído humano la frecuencia sonora?

El MP3 es uno de los tantos formatos MPEG, de hecho; su primer nombre era MPEG Layer 3. ¿Y qué es MPEG? MPEG es la sigla de Moving Picture Experts Group (Grupo de expertos en imágenes en movimiento). En sus orígenes, el MPEG era un pequeño grupo encargado del desarrollo de normas de codificación para audio y vídeo, formado en el Comité Técnico para la Tecnología de la Información ISO [3]. Técnicamente hablando es un

sistema de compresión aplicado al audio, cuya invención se basa en un estudio relacionado con el oído humano. Para crear MP3, los científicos analizaron cómo el oído humano y el cerebro perciben el sonido. Esta técnica de codificación, de alguna manera, “engaña” al oído eliminando las partes menos esenciales de un archivo de música.

Por ejemplo, si dos notas son muy similares, o si un tono alto y bajo ocurre exactamente el mismo tiempo, el cerebro percibe sólo a uno de ellos; entonces el algoritmo MP3 selecciona la señal más importante y desecha el otro. Los investigadores recortaron, además, las frecuencias muy altas y muy bajas que no son percibidas por el “oído consciente”. De esta manera, la pérdida es inaudible y el tamaño del archivo se reduce 15 veces en relación a su original. Es por ello que año con año surgen nuevas tecnologías que buscan innovar la calidad del audio y reducir el tamaño del archivo, para poner más en menos espacio.

La portabilidad es lo que ha hecho del procesamiento digital de señales, un negocio muy rentable en los últimos años, y de la investigación todo un arte. Sin embargo el avance de esta ciencia actualmente sólo se limita a investigadores y grandes compañías que mejoran sus tecnologías, pero en las universidades aún no está al alcance de todos, por este motivo, es necesaria la creación de software que además de cumplir con la tarea de compresión, describan de forma didáctica y comprensible dicho proceso.

Todas las personas buscan productos de calidad que brinden servicios de compresión en el menor tiempo posible, lo cual sólo es posible con la ayuda de grandes procesadores que realizan estos procesos en unos cuantos segundos.

Debido a las nuevas y crecientes demandas que durante las últimas décadas se han desarrollado por parte de los usuarios de la tecnología, el internet, la computación, y los sistemas de multimedia ha sido necesario reemplazar el audio analógico por el audio digital [4].

## 1.2 Problemas de Almacenamiento

En particular los formatos de grabación digital de audio con calidad CD requieren mucho espacio de almacenamiento, por ejemplo, la señal de audio digital contenida en un CD de audio convencional consiste en muestras de 16 bits grabadas a una tasa de muestreo de 44100Hz. Es como si se tratara de una película, donde cada segundo de la misma está compuesto por 44100 fotos que ocupan 16 bits de memoria cada una. Esto implica que cada segundo de audio digital estéreo (2 canales con señal independiente) en un CD ocupa  $16\text{bits} \times 44100\text{Hz} \times 2\text{canales} = 1411200$  bits, sabiendo que  $1\text{byte} = 8\text{bits}$  concluimos que para guardar un segundo de audio digital estéreo calidad CD necesitamos  $1411200 / 8 = 176400\text{bytes}$ , lo que es equivalente a 172.27Kbytes. Una típica pista de 4 minutos de puro audio digital estéreo ocupa entonces unos  $172.27 \times 4 \times 60 = 41,343.75$  Kbyte, que es equivalente a 40.37 Mbyte [5].

Suponiendo que se descarga una pista de 4 minutos a través de Internet, mediante un MODEM de 56kbps. En la práctica nunca se llega a 56kbps sino que las velocidades de transmisión típicas son de 48kbps para una línea en muy buen estado, incluso pueden ser mucho menores si la red es precaria o está congestionada. Entonces, para descargar los 40.37Mbytes a 48kbps tardaríamos aproximadamente unos 117.6 minutos, es decir: con una conexión de 56k en condiciones óptimas habría que estar cerca de 2 horas conectado a internet para bajar una pista de audio de 4 minutos. Por lo tanto es totalmente impráctico bajar audio digital sin compresión con una conexión de 56k. Si intentamos reducir el tamaño de la señal, por ejemplo, cambiando la tasa de muestreo de 44100KHz a 8000Hz, el audio resulta de menor tamaño pero también de mucha menor calidad. El estándar de codificación MP3 permitió reducir drásticamente la cantidad de información necesaria para representar a una señal de audio de alta calidad. El típico archivo MP3 de calidad cercana a la de un CD ocupa unos 128kbits por segundo, por lo que nuestra pista de audio de 4 minutos comprimida en el formato MP3 ocuparía unos 3750Kbytes. Si

estamos en las mismas condiciones mencionadas anteriormente, recibir este archivo MP3 a través de nuestro modem tomaría unos 10.67 minutos [6].

El formato MP3 se hizo tan popular por este motivo, y es que permite obtener en poco más de 10 minutos lo que antes nos hubiese tomado cerca de 2 horas conectadas a Internet, sin tener una pérdida significativa de calidad. El estándar MP3 logró transformar la transmisión de audio digital de alta calidad en algo práctico, razonable y convincente para el usuario promedio de Internet.

A pesar de que el MP3 resultó muy novedoso, dadas las nuevas demandas de los usuarios así como la reducción del ancho de banda disponible para el envío de audio o la necesidad de transportar señales de audio por internet en un tiempo considerablemente pequeño, fue necesaria la creación de nuevos formatos que tuvieran calidad similar o mayor a la del MP3 y que requieran menor capacidad de almacenamiento. En respuesta a esta necesidad, la investigación se ha dedicado considerablemente al desarrollo de algoritmos para codificación de alta fidelidad de audio digital. Como resultado, muchos algoritmos se han propuesto, y varios se han convertido en exitosas normas de productos comerciales.

Es común asociar a MP3 con internet, y entendemos que son archivos pequeños de audio, pero no alcanzamos a visualizar que este novedoso invento ya casi tiene 10 años de su aparición, y sin embargo a pesar de su creación, se ha transmitido mundialmente hasta nuestras fechas.

Sin embargo en el MP3, debido a que solo posee 2 canales, es común que se pierda esa sensación envolvente característica de los formatos sin compresión, por lo que aun quedaban algunas mejoras por realizar. Como fruto de muchas investigaciones surge AAC (Advanced Audio Coding) que con sus 48 canales independientes, proveía de una excelente calidad de audio con muy poco espacio de almacenamiento, por ejemplo, una

señal de audio con formato MP3 que normalmente ocupa 5.71MB, con el formato AAC es de solamente 2.66MB.

AAC es un estándar para audio digital que usa un esquema de compresión con pérdida y se anunció como el sucesor de MP3 por sus propios creadores. Está especificado como Parte 7 del estándar MPEG-2, y Parte 3 del estándar MPEG-4, sin embargo de cualquier forma es común referirse a él como AAC. Su popularidad creció al establecerse como el formato por defecto de famosas y prestigiosas marcas de reproductores portátiles y algunas tiendas de música por internet, así como de video juegos de moda.

El AAC utiliza una codificación variable de la frecuencia de bits (abreviado VBR), un método de codificación que adapta el número de bits utilizados por segundo para codificar datos de audio, en función de la complejidad de la transmisión de audio en un momento determinado.

El algoritmo que se utiliza tiene un rendimiento superior al del MP3, que produce una mejor calidad en archivos pequeños y requiere menos recursos del sistema para codificar y decodificar.

A diferencia de los dos canales (estéreo) que, como máximo, puede admitir el formato MP3, AAC permite sonidos polifónicos con un máximo de 48 canales. El formato AAC también ofrece frecuencias de muestreo que varían de 8 Hz a 96,0 kHz, en contraposición con las frecuencias de mp3 que varían de 16 a 48 kHz. Los archivos AAC, que tiene la extensión .mp4 (para MPEG-4), .m4a (para MPEG-4 Audio) o .m4p (para MPEG-4 protegido), son, al final, más pequeños que los archivos MP3 [7].

# Capítulo 2

## Antecedentes

### 2.1 Del Audio Analógico al Digital

Se conoce de la existencia de la música desde tiempos remotos, incluso antes de la aparición de Cristo, y es "El arte de combinar los sonidos y los silencios, a lo largo de un tiempo, produciendo una secuencia sonora que transmite sensaciones agradables al oído, mediante las cuales se pretende expresar o comunicar un estado del espíritu". Estos sonidos eran producidos inicialmente mediante la percusión de objetos como piedras, el piso, cueros de animales, o cualquier cosa que sonara placentero al oído.

A través del tiempo se han creado nuevos instrumentos incrementando el número de los estilos musicales. Debido al incremento de estilos musicales, la expansión de éstos fue inevitable creando la necesidad de inventar equipos que fuesen capaces de grabar y reproducir música. Esto sucedió ya que no todo el mundo era capaz de crear música.

Estos equipos resultaron fundamentales para dar paso a la innovación de equipos ya existentes. Desde aquel entonces los equipos fueron evolucionando pasando por el fonógrafo, gramófono, el casete, el CD y hasta llegar al el mp3, entre otros, que son los que conocemos hoy en día [8].

El audio analógico en condiciones óptimas representa la más alta calidad que al audio se refiere, ya que no presenta ningún tipo de pérdida durante el proceso de grabación ni de reproducción del sonido.

La era del audio digital, comienza tras un largo proceso, producto de la aparición y desaparición de formas de grabar y reproducir los sonidos.

- ◆ Para el año 1877 fue inventado el primer tocadiscos llamado fonógrafo por el inventor Tomas A. Edison, en el cual consistía en la grabación de un sonido en un cilindro que luego hacía girar contra una aguja. Ésta subía y bajaba en los surcos del cilindro, produciendo vibraciones que se amplificaban en una bocina cónica.
- ◆ En 1888, un ciudadano alemán radicado en Washington llamado Emilio Berliner, en el cual registraba y patentaba una maquina muy similar al fonógrafo pero con la diferencia de que ésta no usaba el cilindro como soporte de la grabación sino, un disco plano y además la impresión se efectuaba en el surco por amplitud lateral y no como en el cilindro que se hacía en forma vertical. A esa máquina parlante Emilio Berliner la bautizó con el nombre de gramófono.

La principal ventaja que trajo el gramófono sobre el fonógrafo es que al solo una vez se podrían reproducir miles de copias, en cambio el fonógrafo si se quería reproducir miles de copias, se tenía que insertar el sonido varias veces en diferentes fonógrafos.

- ◆ En 1948 surge el disco LP (de la expresión inglesa long play) o disco de larga duración es un disco de vinilo de tamaño grande, 30 cm, y en los cuales se puede registrar, en formato analógico, un máximo de unos 20-25 minutos de sonido por cara. Un disco LP suele constar de unas diez canciones, aunque depende de su longitud. También hay LP fabricados en plástico, aluminio u otros materiales. Normalmente están grabados a una velocidad de 33 y 1/3 o 16 rpm [6].
- ◆ En 1963 Desarrollada por Philips, la Casete Compacto (Compact Cassette) es una cinta magnética pequeña, encapsulada en una caja almacén que la protege y evita su manipulación directa. Con este sistema se pueden efectuar grabaciones domésticas con una calidad de sonido aceptable, y su reducido tamaño posibilitó la

- ◆ construcción de aparatos grabadores y reproductores portátiles, alimentados con pilas.

Sin embargo, el sistema presenta algunos inconvenientes. Como toda cinta, su acceso es secuencial, y localizar puntos en la grabación es una labor que consume tiempo (y requiere cierta habilidad). El ruido de fondo generado por la naturaleza granular del recubrimiento magnético es notorio. Y la desventaja más grande es la calidad de sonido, que además de ser limitada, depende de la calidad y el estado físico del equipo grabador y del equipo reproductor (alineación de cabezales, suciedad, magnetismo acumulado, desgaste natural), y del fabricante de la casete [9].

- ◆ En 1982 el Disco Compacto (Compact Disc, o CD), fue el resultado de un trabajo conjunto entre las casas Philips y Sony, que unieron fuerzas para desarrollar un nuevo y revolucionario formato de audio.

Fue el primero que aprovechaba las ventajas del audio digital, donde es posible separar y eliminar ruidos e imperfecciones ajenas al sonido original, y también el primero donde se emplearon métodos ópticos para la lectura de la información de audio, realizada mediante un haz láser que explora la superficie del disco. De este modo, al no existir contacto físico entre el medio y la máquina, los discos no se desgastan, y la calidad del sonido es la mejor conseguida hasta el momento por un formato de audio doméstico.

- ◆ En 1987 la vuelta a los formatos secuenciales vino con la introducción al mercado la llamada Cinta de Audio Digital (Digital Audio Tape, o DAT). Basada en el registro helicoidal de las grabadoras de video VHS, este sistema permite hacer registros digitales sin compresión de datos, y se pueden efectuar copias perfectas de discos compactos.

Fue esta perfección la que preocupó a las compañías disqueras por el temor a la proliferación de copias ilegales de CDs en DAT (el problema real para la industria musical aparecería años después, con la entrada en escena de los grabadores de CD-ROMs). La presión ejercida por las disqueras para impedir la masificación del sistema surtió efecto,

pues las máquinas DAT no bajaron de precio, y su uso quedó relegado al ámbito profesional, donde se usa actualmente. Después de alcanzar el éxito con el CD, Sony y Philips siguieron caminos separados en el desarrollo de un medio regrabable para el consumidor no profesional.

- ◆ En 1992 fue la presentación al público de la Casete Compacto Digital (Digital Compact Cassette o DCC). La aparición del DCC obliga a Sony a lanzar al mercado ese mismo año el sistema Minidisco. Sin embargo, el DCC no logra captar al público y los precios de las máquinas y las cintas se mantienen muy altos. En 1996, Philips decide discontinuar su fabricación.
- ◆ En 1992 el mini disk fue fabricado por la compañía Sony en el año 1992 y es un formato doméstico de audio digital de alta calidad, que consiste en un pequeño disco y una máquina capaz de reproducir o grabar sonido en él.

Éste fue fabricado con el objetivo de reemplazar a los casete compacto. Su uso puede ser además de domestico también profesional. Está fabricado por un material llamado makrolon recubierto y encapsulado [10].

- ◆ A partir de 1990 ya en plena era digital, se comenzó a comercializar los lectores láser para discos de vinilo. Aunque el disco de vinilo era bueno en calidad el CD predomino en el mercado por su buena calidad de grabación, tamaño y durabilidad.

La introducción del CD y su método de grabado digital marcó un significativo avance frente a las tecnologías de grabación analógicas. Estos avances estaban relacionados fundamentalmente con la fidelidad y la forma de reproducción. Así mismo es posible percibir que en un CD, el sonido permanece igual sin importar el número de veces que se reproduzca, además cuenta con la ventaja de poder eliminar tanto ruido como señales que distorsionen a la señal original. Por otra parte el CD también supuso un medio de almacenamiento y marcó la pauta para el desarrollo de todas las tecnologías que

actualmente conocemos en cuanto a la codificación y compresión de alta fidelidad de audio se refiere [11].

Todo este proceso generaba una gran cantidad de material y de espacio al portar un reproductor de estos, espacio en cuanto a tamaño físico y de archivos, llegando a la aparición del Reproductor MP3: El reproductor MP3 es el equipo de música más reciente y mejor conocido por la juventud de hoy en día.

- ◆ En julio de 1995, llegó el segundo gran boom en la revolución del audio digital: el MP3, el cual es capaz de alojar en su formato a más de 150 canciones en un CD de audio normal, y posee una gran flexibilidad para ajustar su calidad de compresión desde 32Kbps con calidad similar a la de AM, hasta 320Kbps con una fidelidad cercana a la del CD. Siendo la calidad de bitrate estándar para éste de 128Kbps [4]. MP3 es un algoritmo de codificación perceptual desarrollado por el consorcio MPEG y que finalmente, se ha estandarizado como norma MPEG Audio Layer 3.

El sistema de codificación perceptual es un sistema de compresión con pérdida, esto quiere decir que el sonido original y el comprimido no son exactamente iguales. Estas pérdidas responden al funcionamiento del oído humano, así aunque los sonidos no son iguales si los percibimos como si lo fuesen.

- ◆ Poco después surge la tecnología AAC (Advanced Audio Coding, Codificación de Audio Avanzada), la cual marcó un paso adelante en cuanto a la calidad de sonido pues resulta casi igual al de un CD, permitiéndoles a los usuarios experimentar las transmisiones tal como fueron concebidas. El estándar AAC es sin duda el nuevo estándar de la industria creado por el influyente grupo Moving Picture Experts Group (MPEG). La adopción de esta tecnología coloca a STREAMTHEWORLD a la cabeza de la elite de desarrolladores y proveedores de aplicaciones de difusión continua en todo el mundo. Posterior a éste y con la popularización del audio digital continuo existiendo una creciente demanda de técnicas de compresión y transmisión efectivas [12].

## 2.2 Moving Picture Layer III

MP3 es un formato de datos que debe su nombre a un algoritmo de codificación llamado MPEG 1 Layer 3, el cual, a su vez, es un sistema de compresión de audio que permite almacenar sonido con una calidad similar a la de un CD y con un índice de compresión muy elevado.

El MP3 es un formato de audio digital comprimido con pérdida en el cual se puede grabar o introducir información de audio la cual es comprimida en una memoria, la compresión con pérdida es aquella que una vez comprimida la información ésta no se puede recuperar en su forma original. El formato MP3 se convirtió en el estándar de audio y compresión de audio de alta calidad [13]. Desde su primera reunión en 1988, el grupo que desarrollo el MPEG ha crecido hasta incluir 350 miembros de distintas industrias y universidades.

MPEG ha normalizado los siguientes formatos de compresión y normas auxiliares:

- ◆ MPEG-1: estándar inicial de compresión de audio y vídeo. Usado después como la norma para CD de vídeo, incluye formato popular de compresión de audio Capa 3 (más conocido como MP3).
- ◆ MPEG-2: normas para audio y vídeo para difusión de calidad de televisión. Utilizado para servicios de TV por satélite como Direct TV (Cadena estadounidense de televisión vía satélite de difusión directa), señales de televisión digital por cable y (con ligeras modificaciones) para los discos de vídeo DVD.
- ◆ MPEG-3: diseñado originalmente para HDTV (Televisión de Alta Definición), pero abandonado posteriormente en favor de MPEG-2.
- ◆ MPEG-4: expande MPEG-1 para soportar "objetos" audio/vídeo, contenido 3D, codificación de baja velocidad binaria y soporte para gestión de derechos digitales (protección de copyright).

- ◆ MPEG-7: sistema formal para la descripción de contenido multimedia.
- ◆ MPEG-21: MPEG describe esta norma futura como un marco multimedia.

El formato MP3 fue desarrollado principalmente por Karlheinz Brandenburg, director de tecnologías de medios electrónicos del Instituto Fraunhofer IIS, perteneciente al Fraunhofer-Gesellschaft, red de centros de investigación alemanes, que junto con Thomson Multimedia controla el grueso de las patentes relacionadas con el MP3 [14].

El formato MP3 se convirtió en el estándar gracias a la posibilidad de ajustar la calidad de la compresión, proporcional el número de bits que se transmiten por unidad de tiempo (bitrate), y por tanto el tamaño final del archivo, que podía llegar a ocupar 12 e incluso 15 veces menos que el archivo original sin comprimir. Fue el primer formato de compresión de audio popularizado gracias a Internet, ya que hizo posible el intercambio de ficheros musicales. Los procesos judiciales contra empresas como Napster y AudioGalaxy son resultado de la facilidad con que se comparten este tipo de ficheros.

Unas de las ventajas del MP3 son:

- ◆ La posibilidad de ajustar la calidad de la compresión, proporcional al tamaño por segundo (bitrate), y por tanto el tamaño final del archivo.
- ◆ Así como también los archivos que ocupaban gran espacio ahora se pueden reducir a 12 e incluso 15 veces menos que el archivo original sin comprimir.

Tras el desarrollo de reproductores autónomos, portátiles o integrados en cadenas musicales (estéreos), el formato MP3 llega más allá del mundo de la informática.

A principios de 2002 otros formatos de audio comprimido como Windows Media Audio, empiezan a ser masivamente incluidos en programas, sistemas operativos y reproductores autónomos, lo que hizo prever que el MP3 fuera paulatinamente cayendo en desuso, en favor de otros formatos, como los mencionados, de mucha mejor calidad. Uno de los

factores que influye en el declive del MP3 es que tiene patente. Técnicamente no significa que su calidad sea inferior ni superior, pero impide que la comunidad pueda seguir mejorándolo y puede obligar a pagar por la utilización de algún códec, esto es lo que ocurre con los reproductores de MP3. Aun así, a inicios del 2008, el formato MP3 continua siendo el más usado y el que goza de más éxito.

A raíz del descubrimiento del MP3, empresas como Apple Computers, Real Networks o Sony han desarrollado, cada uno, tecnologías propias de reproducción. Esto hace, por ejemplo, que canciones compradas en la tienda de música iTunes de Apple sólo pueden escucharse en los iPods de Apple. Últimamente las casas discográficas están empezando a preocuparse por la futura industria legal de MP3, y prueba de ello son las llamadas arañas que se están desarrollando. Estas arañas son sistemas de búsquedas de canciones que localizan posibles canciones y comprueban si son legales. En muchos sitios encontrarás canciones MP3 pero con extensión cambiada, por ejemplo a Zip o cualquier otra forma de compresión [15]

Por tanto en este trabajo se enfoca a presentar los resultados obtenidos de la compresión de audio en MP3 con la programación en MATLAB.

### **2.3 Advanced Audio Coding**

Aunque la historia de la tecnología MP3 es reciente, en poco tiempo se convirtió en el estándar para la música digital. Éste formato, es una abreviatura de MPEG Layer 3, apareció como un sistema para comprimir archivos de audio y propiciar que cientos de canciones se puedan almacenar en un CD, y que éstas puedan transmitirse fácilmente por internet.

Con esta tecnología coincidió el surgimiento de Napster, un programa de internet que aprovechaba lo ligero de los archivos MP3 para que los usuarios pudieran compartir sus bibliotecas musicales a través de la red. A pesar de que Napster y otros programas

similares para intercambio de canciones fueron obligados a desaparecer u operar con protección de derechos de autor, cientos de canciones quedaron en los discos duros de los usuarios, los que impulsaron la venta de reproductores portátiles de música en formato MP3.

El primer reproductor de MP3 fue lanzado a finales de 1998 por la compañía Diamond Multimedia (el modelo Rio PMP300), y desde entonces surgieron múltiples modelos en el mercado. Y fue hace de cinco a seis años que Apple lanzó el iPod, que ahora domina esta industria.

Es precisamente el iPod el que propició el cambio en el uso del tradicional formato MP3 a un formato AAC (Advanced Audio Coding), protegido (M4A), que puede preservar los derechos de autor, pero que es incompatible con la mayor parte de bibliotecas musicales que tienen los usuarios.

"MPEG-4 (AAC) es el nuevo estándar mundial para la creación de multimedia interactiva, el envío y la reproducción para internet. Lo que MPEG-1 significó para la industria de CD-ROM y MPEG-2 para el desarrollo del DVD, MPEG-4 significará para internet", explican ejecutivos de Apple en un comunicado [16].

Agregan que la adopción del estándar MPEG-4 no es crítica sólo para computadoras de escritorio, sino que es muy importante a medida que los medios digitales se expanden en nuevas áreas como codificadores de TV, dispositivos inalámbricos y consolas de video. Por cierto, las compañías del MPEG-4 Industry Forum (M4IF) como Philips y Sony han lanzado muchos dispositivos de este tipo.

La decisión de Apple para adoptar el formato AAC fue la protección de los derechos de autor (dado que esta compañía también vende música a través de su portal en forma

legal), y el futuro de la multimedia a través de múltiples dispositivos, porque iPod también reproduce videos y películas.

Otra razón es que AAC es un formato más evolucionado que MP3. "En el soporte en audio QuickTime incluye codificadores ACC, que ofrecen por lo menos el doble de claridad de audio MP3, a una misma tasa de bits con el mismo o menor tamaño de archivo", explican ejecutivos de Apple [17]. No obstante todo lo anterior, es posible convertir la música de un formato MP3 a M4A, con programas como el mismo iTunes. En contrasentido, este programa tiene una opción avanzada para exportar archivos en formato AAC, el cual es compatible con la mayor parte de dispositivos de reproducción de música.

Sin embargo, pueden existir archivos imposibles de convertirse, dado que el sistema puede detectar la falta de los derechos de autor.

- ◆ .m4a Un estándar de audio AAC, que se puede reproducir en cualquier dispositivo que soporte este formato.
- ◆ .mp4 Un estándar de audio AAC; éste se perfila como la siguiente generación después de MP3 y se puede reproducir en cualquier reproductor que soporta AAC.
- ◆ .m4p Actualmente sólo se pueden encontrar este tipo de archivos en la tienda iTunes de música de Apple en internet. Estos archivos contienen el sistema de administración de derechos de autor adoptado por Apple. Se pueden reproducir solamente en el programa iTunes, en un iPod o en el teléfono ROKR de Motorola.
- ◆ .m4b Éste es un formato de audio libros, que permite marcar el momento en que el usuario lo dejó para que en la siguiente ocasión lo pueda leer desde ese punto [18].

## 2.4 Estándares de Compresión de Audio

A partir de que comenzó la revolución del audio digital se comenzaron a crear diversos sistemas de compresión, como respuesta a la problemática de almacenamiento, y que posteriormente se convertirían en los estándares que hoy en día usamos cotidianamente. Los estándares de compresión de audio cumplen fundamentalmente la función de reducir la cantidad de datos digitales necesarios para reproducir una señal auditiva. Lo que comúnmente se denomina "compresión de datos", pero aplicado a un fin muy concreto.

Para poder entender los estándares que existen en la compresión de audio se necesita saber que es la "compresión" y se dice que está en el lenguaje digital significa que la tasa de bits se reduce, idealmente dejando sin cambios la dinámica de la señal [19].

Este proceso tiene que ser eficaz, esto es: la técnica empleada para la reducción debe permitir que el terminal receptor o reproductor presente la información de modo que, luego de expandirla, los observadores no se encuentren en capacidad de notar diferencia alguna si tuvieran la oportunidad de comparar la señal original con la procesada.

Además, el proceso tiene que ser eficiente, es decir: ha de efectuarse en tiempo real y con un costo que justifique los beneficios pretendidos. El dominio elegido para este proceso es el digital, por las innumerables ventajas que posee sobre el analógico. Las técnicas de compresión de audio, en general, se pueden clasificar en dos categorías básicas: sin pérdidas y con pérdidas. Aunque hay muchas diferentes técnicas de compresión, todas ellas caen en una u otra de estas categorías.

### 2.4.1 Estándares de Compresión de Audio sin Pérdida

Los estándares de compresión de audio sin pérdida son aquellos en los que no existe pérdida de información desde la fuente de audio, lo que lo convierte en un formato ideal para colecciones de muy alta calidad ya que permite reconstruir el audio original en su totalidad, también es conocida como compresión reversible, sin embargo, permite solo una modesta reducción de tasa de bit, por lo que un codificador de este tipo no puede garantizar un factor de compresión en particular [20].

En la compresión sin pérdidas, los datos del descompresor son idénticos bit por bit a los datos originales de la fuente. La señal de audio comprimido puede ser descomprimida sin omitir ninguna muestra de información. Es importante apreciar que un codificador sin pérdidas no puede garantizar un factor de compresión en particular y por lo tanto los vínculos de comunicaciones o los procedimientos de almacenamiento utilizados deberán ser capaces de funcionar con una tasa de datos variable.

Los datos de fuente, que en un códec determinado, causan factores pobres de compresión son conocidos como dificultad del material. Se debe indicar que la dificultad es frecuentemente una función del códec. En otras palabras, los datos que un codificador encuentra difícil puede que otro no lo encuentre difícil.

Los codificadores sin pérdida pueden incluirse en los planes de prueba de la tasa de bits erróneos. Es también posible poner en serie o en tándem codificadores sin pérdida sin precauciones especiales. Las canciones comprimidas en este tipo de formatos pesan alrededor de 70% a 50% del original, por lo que resulta una buena opción para tener alta fidelidad [19].

### 2.4.2 Estándares de Compresión de Audio con Pérdida

La compresión con pérdida, permite una más alta reducción de la tasa de bits. En general, una mayor compresión crea una mayor distorsión pero con técnicas cuidadosamente seleccionadas, esta distorsión puede hacerse bastante imperceptible al oído humano. Los codificadores con pérdida no son apropiados para los datos de computadora, pero son habituales en aplicaciones de audio puesto que ellos permiten un mayor factor de compresión que los codificadores sin pérdida.

Para reducir el volumen de datos generado, es necesario utilizar estándares de compresión de audio con pérdida, que permitan ofrecer una excelente calidad de sonido en un tamaño que es varias veces inferior al original. El principio básico en el que se asientan estos algoritmos de compresión es, de nuevo, el de la eliminación de información inútil no apreciada por el oído.

MP3 fue pionero en este campo y permitió, con la eliminación de rangos de frecuencias inaudibles para el ser humano, lograr relaciones de compresión de 12 a 1. Así, cada 12 bytes del fichero original podía ser representado con sólo 1, manteniendo una calidad notable. Aunque depende de la calidad del sonido (estéreo o mono) y la frecuencia de muestreo se obtendrán calidades y tamaños de fichero distintos [19].

# Capítulo 3

## Herramientas de Procesamiento Digital de Señales

### 3.1 Filtros Digitales

Como regla general, cualquier algoritmo o sistema de tratamiento puede interpretarse como un filtro. Aquí se entiende como filtro aquel sistema lineal e invariante que permite el paso de las componentes de la señal existentes en un determinado intervalo frecuencial, y elimina las demás.

De forma ideal, el módulo de la respuesta frecuencial del filtro toma un valor constante en el margen de frecuencias que queremos conservar, que se denomina banda de paso. El intervalo de frecuencias complementario al anterior en que la respuesta en magnitud es nula se denomina banda de rechazo o atenuada. La banda de transición es aquella que se sitúa entre dos bandas cuyas atenuaciones están especificada, por tanto, se caracteriza porque no imponemos al filtro ningún requisito en dicho intervalo frecuencial dando libertad de esa forma al diseño del filtro siempre y cuando se cumplan los requisitos impuestos en la banda de paso y de rechazo. Los cuatro filtros básicos, desde el punto de vista ideal del comportamiento del módulo de la respuesta frecuencial, según sea la posición relativa de bandas de paso y bandas atenuadas, reciben el nombre de paso bajo, paso alto, paso banda y elimina banda, dependiendo de la parte del espectro de frecuencias en la que se centra la banda de paso.

En este se va a ocupar filtro digitales y un filtro digital es un sistema discreto utilizado para extraer características desde el dominio de la frecuencia sobre señales muestreadas. La operación de filtrado se realiza por medio de cálculos directos con las señales muestreadas. Las ventajas que presentan los filtros digitales frente a los analógicos son las siguientes:

- ◆ Respuesta dinámica: El ancho de banda del filtro digital está limitado por la frecuencia de muestreo, mientras que en los filtros analógicos con componentes activos suelen estar restringidos por los amplificadores operacionales.
- ◆ Intervalo dinámico: En filtros analógicos aparecen derivas que limitan por abajo el rango y se saturan con la alimentación. En cambio en los filtros digitales es fijado por el número de bits que representa la secuencia, y el límite inferior por el ruido de cuantificación y los errores de redondeo
- ◆ Conmutabilidad: Si los parámetros de un filtro se conservan en registros, los contenidos de dichos registros pueden ser modificados a voluntad. Además, estos filtros se pueden conmutar, pudiéndose multiplexar en el tiempo para procesar varias entradas a la vez.
- ◆ Adaptabilidad: Un filtro digital puede ser implementado en soporte físico (hardware) o mediante un programa de ordenador (software).
- ◆ Ausencia de problemas de componentes: Los parámetros de los filtros se representan por medio de números binarios y no derivan con el tiempo. Al no haber componentes, no hay problemas de tolerancia o deriva de componentes, y ningún otro problema asociado con un comportamiento no ideal de resistencias, condensadores, bobinas o amplificadores. Tampoco existen problemas de impedancia de entrada ni salida, ni efectos de adaptación de impedancias entre etapas.

Una distinción fundamental en los sistemas discretos dinámicos lineales e invariantes, y en particular en los filtros digitales, es la duración de la respuesta ante el impulso. Se habla

de sistemas de respuesta de pulso finito o no recursivo (FIR, finite impulse response) y de sistemas de respuesta infinita o recursivo (IIR, infinite impulse response) [21].

### **3.1.1 Filtros FIR**

Los filtros no recursivos FIR tienen ventajas muy interesantes que les hacen ser ampliamente utilizados en múltiples aplicaciones. La característica más destacable es su facilidad de diseño para conseguir una respuesta en frecuencias de fase lineal, esto es, la señal que pase a través de él no será distorsionada. Los FIR son por su propia constitución estables, no habiendo problemas en su diseño o fase de implementación.

Aunque el diseño de los FIR requiera de una gran cantidad de operaciones de sumas y multiplicaciones, tanto su estructura de programación como su implementación en soporte físico resultan fáciles y escalables. Un parámetro importante es la respuesta de fase del filtro, que interesa sea lo más lineal posible, de ahí que usualmente se suele representar el retardo de grupo. El hecho de que sea lo más lineal posible el retardo de fase implica que el retardo de grupo sea constante. Esto implica que el filtro no influya ni distorsione la salida deseada del filtro en la banda de paso, objetivo primordial de un filtro.

Existen diversos métodos para el diseño de filtros FIR, entre ellos destacan tres. El más sencillo es el de enventanado de la respuesta impulsional.

Durante mucho tiempo se ha trabajado en el diseño de filtros analógicos obteniendo para ello implementaciones caracterizadas porque al llevarlas al campo digital tenían una respuesta de tipo IIR. Parece más o menos intuitivo el pensar que si tomamos la secuencia infinita de la respuesta impulsional y nos quedamos con una parte de ella, el resultado desde el punto de vista de la función de filtrado del sistema sería el mismo. Aunque a grosso modo eso es así, desde el punto de vista frecuencial se producen una serie de deformaciones en el espectro del filtro obtenido que nos llevarían a considerables errores a no ser por el uso de ventanas pensadas para este uso.

Esas ventanas no son más que secuencias de longitud finita que tienen una respuesta frecuencial que permite que al ser multiplicadas por la función de transferencia utilizada el error no sea muy grande. Esto es una descripción intuitiva y nada rigurosa del sentido del enventanado.

Otra metodología simple para el diseño de filtros la ofrece el muestreo en frecuencia de la respuesta ideal. El procedimiento asegura un error nulo para la aproximación en un conjunto finito de frecuencias equiespaciadas, aquéllas en las que se muestrea la respuesta frecuencial ideal. El diseño por muestreo en frecuencia es muy popular dada su sencillez. Presenta, sin embargo, importantes deficiencias. No es posible controlar directamente la amplitud del error. Tampoco se conoce un criterio estimativo del orden del filtro. Para conseguir un comportamiento ajustado a una plantilla debe acudir a una estrategia de ensayo y error tediosa, que en la mayoría de los casos proporciona un filtro de orden excesivo y que, incluso, no garantiza la existencia de solución.

La tercera metodología empleada es la del uso de filtros óptimos, considerados así aquellos con rizado de amplitud constante. La respuesta frecuencial que ofrecen los filtros diseñados mediante la manipulación directa del comportamiento ideal (el enventanado de la respuesta impulsional o el muestreo de la respuesta frecuencial) presenta un error en las bandas de paso y atenuadas cuya amplitud crece en las proximidades de las bandas de transición. La solución a ese problema que aporta esta metodología es la de repartir el error por las diversas bandas usando una función que lo permita [22].

Los filtros FIR tienen un mejor comportamiento desde el punto de vista del retardo de grupo y de la distorsión de fase que provoca el filtro que los filtros IIR.

### **3.1.2 Filtros IIR**

La ventaja de los filtros IIR respecto a los FIR es la de tener un menor orden del filtro para iguales especificaciones de diseño. Aunque la desventaja es la falta de desfase lineal

introducido por el filtro, así como la necesidad de realizar estudios de estabilidad, pues ésta no está garantizada en el diseño.

Los filtros recursivos pueden ser diseñados por varios métodos, siendo el más común el basado en las transformaciones bilineales. Este procedimiento requiere del conocimiento de la función de transferencia en el dominio  $s$  del filtro a diseñar. Los coeficientes del filtro en el dominio  $s$  son transformados a uno equivalente en el dominio  $z$ , los coeficientes de la discretización formarán el filtro IIR.

El origen de este proceder viene dado por la cantidad de experiencia acumulada en el diseño de filtros analógicos. Por tanto, todos los polinomios, tablas, métodos analíticos y gráficos para definir el filtro analógico, empleados en el anterior capítulo, serán usados en el diseño de los filtros recursivos. Si bien hay varios métodos de discretización, la mayoría de ellos tienen problemas de solapamiento en frecuencias, por realizar una relación entre el plano  $s$  a  $z$  de varias regiones del dominio  $s$  a una sola  $z$ . Sin embargo, la transformación bilineal realiza una transformación unívoca entre el dominio  $s$  a  $z$ .

En el caso de los filtros con respuesta al impulso de longitud infinita, la expresión de la función de transferencia en el dominio  $Z$  es en forma de cociente de polinomios. Por eso, la forma de obtener en general la salida en este tipo de filtros es mediante fórmulas recursivas.

Una de las particularidades de estos filtros respecto al tipo FIR es el hecho de que su comportamiento respecto a la fase es peor. Además, estos filtros proceden directamente de la aplicación de métodos que tradicionalmente se han aplicado en el desarrollo de filtros analógicos tales como eran las aproximaciones de Butterworth, Chebyshev o Elíptica. Como ya se ha mencionado, la aproximación elíptica es, entre todas, la que requiere menor orden, por lo que es comúnmente utilizada cuando el principal interés se centra en minimizar el orden del filtro; sin embargo, su fase es la que más se aleja del comportamiento lineal.

También comprobaremos que la aproximación inversa de Chebyshev proporciona filtros con menor distorsión de fase que la aproximación elíptica a costa de aumentar ligeramente el orden. La aproximación de Chebyshev precisa igual orden que la inversa de Chebyshev, pero su fase se comporta considerablemente peor.

Finalmente la aproximación de Butterworth es la que presenta una fase más próxima al ideal para un orden dado, pero el orden que necesita para cumplir las especificaciones suele ser notablemente mayor al que requieren las demás [23].

### 3.2 Muestreo

La idea de muestras referidas a un sistema basado en un computador digital, se concreta en la toma de valores de una señal continua en sucesivos instantes de tiempo. El muestreo de señales consiste en la construcción de secuencias a partir de señales continuas. Un problema que salta a la vista es la pérdida de información.

En este punto se analizará fundamentalmente este problema, dándose las condiciones en las que el proceso de muestreo/reconstrucción, (CAD/CDA), no signifique pérdida de información.

El caso más interesante de muestreo es el periodo, que se caracteriza por que los instantes de toma de muestras están regularmente espaciados en el tiempo y a éste se le llama periodo de muestreo,  $T$  y a su inversa, frecuencia de muestreo,  $f_s$ .

Intuitivamente se puede apreciar que el periodo de muestreo va a jugar un papel fundamental en este proceso. La elección de la frecuencia de muestro es un problema fundamental en el sistema de datos muestreados. Conviene muestrear los datos con la menor frecuencia posible, pero que sea lo suficientemente grande como para poder evitar la pérdida de información y sea posible reconstruir la función continua a través de las

muestras. Justamente, el teorema del muestreo es el que puede ayudar a determinar la frecuencia óptima del convertidor analógico/digital.

Para poder establecer las conclusiones del teorema del muestreo, se va a recurrir al estudio frecuencial. Si una señal continua,  $x(t)$  con transformada de Fourier continua,  $x_c(w)$ . Entonces las relaciones señal/transformada serán:

$$x_c(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (1)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x_c(w)e^{j\omega t} dw \quad (2)$$

La relación existente entre la secuencia procedente del muestreo de esta señal y ella misma será como se aprecia:

$$x_k = x(kT) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x_c(w)e^{j\omega kT} dw \quad (3)$$

Si la secuencia  $x_k$  tiene asimismo transformada discreta de Fourier  $x_d(w)$ , entonces estará relacionada con su transformada por medio de la siguiente expresión:

$$x_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} x_d(w)e^{j\omega k} dw \quad (4)$$

El problema es estudiar si existe alguna relación entre estas dos transformadas, la de la señal continua  $x_c(w)$  y la de la secuencia  $x_d(w)$ . Se puede demostrar que la relación existente es como se muestra en la siguiente ecuación:

$$x_d(w) = \sum_{r=-\infty}^{\infty} x_c\left(w + \frac{2\pi r}{T}\right) \quad r = 0, \pm 1, \pm 2, \pm 3, \dots \quad (5)$$

Por tanto, si la señal continua tiene una transformada de Fourier, cuya curva de módulos, es decir es igual para ambos lados del eje Y. Entonces la curva de módulos de la transformada de la secuencia obtenida por muestreo, será periódica  $2P/T$ .

Llegando así a las condiciones que se deben cumplir en el proceso de muestreo para que se pueda realizar sin pérdida de información. Suponiendo que la señal continua  $x(t)$  con transformada de Fourier  $x_c(w)$  sea nula a partir de una determinada frecuencia  $w(0)$  como se indica en la siguiente condición:

$$x_c(w) = 0 \quad |w| \geq w_0$$

Las señales que cumplen esta condición se denominan de banda limitada. Si una señal de este tipo se muestrea con un periodo que verifique:

$$\frac{2\pi}{T} \geq 2w_0$$

Es decir;

$$T \leq \frac{\pi}{w_0}$$

La transformada de Fourier será aun periódica con  $P/T$  se la llama frecuencia angular de Nyquist.

Si por el contrario, el periodo de muestreo es superior a este valor,  $T \geq P/w(0)$  la transformada de la secuencia, apreciara entonces que  $x_d(w)$  y  $x_c(w)$  no coinciden, y por lo tanto el conocimiento de la transformada de la secuencia no permite obtener la transformada de la señal continua. A este fenómeno se llama solapamiento en frecuencias o aliasing. El diseño de filtros antialiasing en la propagación de la señal continua al convertidor A/D, con el objeto de tener señales de banda limitada y que cumplan con las ecuaciones donde dice que la frecuencia de muestreo debe de ser por lo menos el doble de la frecuencia original

Hay que citar que además del problema del muestreo, habría de estudiar el efecto tanto de la cuantificación como el de reconstrucción. El primero analiza cómo afecta el problema del convertidor A/D en la transformación de una señal analógica de infinitos valores a otra de valores finitos; mientras el segundo estudia los problemas de la reconstrucción debido a los convertidores D/A [24].

### 3.3 Modelo Psicoacústico

El sistema auditivo humano es el complejo del cual parte el análisis de cualquier codificador de audio, es decir, es la base de la cadena de actuaciones que determinarán las frecuencias relevantes para la percepción auditiva y que aunadas las limitaciones de éste con la forma de procesar el sonido por el cerebro hacen que la codificación sea cada vez más eficiente. Por tal motivo, es de vital importancia tener un completo conocimiento de cómo funciona este sistema a la hora de diseñar un codificador de audio.

La función principal del modelo psicoacústico en un sistema de codificación perceptual de audio es proporcionar estimaciones precisas del máximo ruido permitido (umbrales de enmascaramiento), de acuerdo con la resolución tiempo/frecuencia del sistema de codificación. En muchas ocasiones, como por ejemplo en el estándar MPEG para audio, simplemente se define el formato de transmisión, permitiendo cambios y mejoras en el modelo perceptual [25].

El oído humano se puede dividir en tres partes, cada una de las cuales realiza un procesamiento diferente de los sonidos que llegan al sistema:

- ◆ **Oído externo:** Es la parte visible del sistema auditivo formado por el pabellón y el canal auditivo. La principal función se limita a la localización de las fuentes del oído en el espacio aunque también realiza otras acciones. Por ejemplo, protege al tímpano del posible daño causado por cuerpos extraños y cambios en la humedad y temperatura. Desde el punto de vista acústico, el canal auditivo (de 2 o 3 cm

aprox.) tiene una frecuencia de resonancia cercana a 4 KHz, lo que provoca una ganancia en la señal en este rango de frecuencias, siendo la causa de la alta sensibilidad del oído en esta banda y del mínimo del umbral absoluto de audición.

- ◆ **Oído medio:** Comienza en el tímpano e incluye toda la cadena de huesos del oído. Básicamente realiza una transmisión del sonido desde el tímpano, a través de los huesos del oído (martillo, yunque y estribo) hasta la entrada del caracol. Esta parte del oído tiene una respuesta adaptada a las frecuencias medias (de 500 a 4000 Hz) porque la adaptación mecánica de las ondas sonoras desde aire (tímpano) a fluido (cóclea) está fisiológicamente diseñada para estas frecuencias.
  
- ◆ **Oído interno:** Es la parte más importante del sistema auditivo desde el punto de vista psicoacústico. Incluye la cóclea o caracol del oído donde se realiza la conversión de señal mecánica a eléctrica. El ruido de la cóclea es excitado por el hueso estribo y estas ondas se propagan hasta donde se encuentran las células sensoriales. Esta propagación tiene la particularidad de que, dependiendo de la frecuencia, el pico de la respuesta de las ondas se sitúa en una parte u otra de la membrana donde están los receptores. Como consecuencia, se excitan diferentes receptores en función de la frecuencia del sonido, de forma que los receptores están sintonizados a la frecuencia de entrada gracias a la conversión frecuencia a lugar que realiza la cóclea. Desde un punto de vista de señal, la cóclea se comporta como un conjunto de filtros paso banda, con anchos de banda no uniformes que crecen con la frecuencia. El concepto de bandas críticas se relaciona con este fenómeno.

Otro fenómeno que tiene lugar en el oído interno es el enmascaramiento, el cual es producido por la presencia en la misma banda (para los mismos receptores) de una excitación suficiente para bloquear la recepción de una señal más débil.

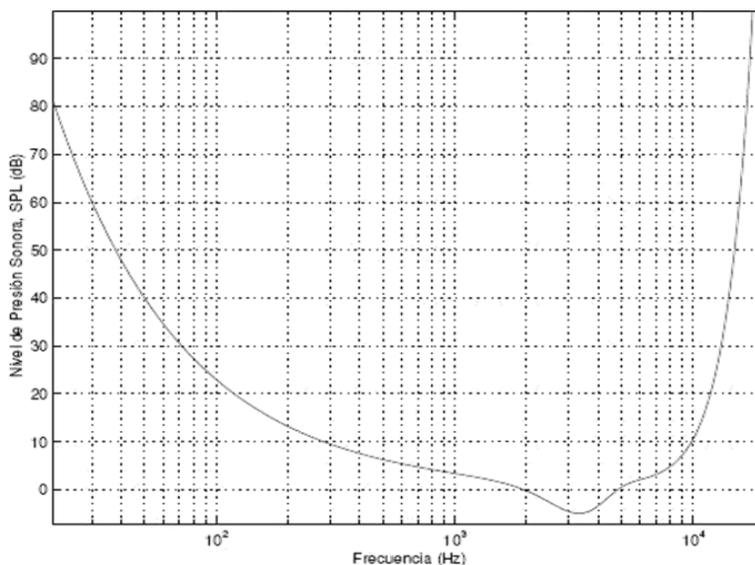
Finalmente, la percepción de un sonido se realiza en el cerebro mediante la composición de las diferentes respuestas eléctricas de las células sensoriales de cada banda enviadas por medio del nervio auditivo [26].

La ciencia que estudia las relaciones entre los estímulos acústicos y las sensaciones auditivas se conoce como psicoacústica. Y es esta la que permite analizar y desarrollar de manera óptima los factores característicos de un codificador de audio eficiente.

### 3.3.1 Umbral Absoluto de Audición

El umbral absoluto de audición o umbral de silencio, como se muestra en la figura 3.1, indica el nivel de presión sonora SPL (por sus siglas en inglés) necesario para que un tono puro en función de la frecuencia se empiece a escuchar.

En la siguiente figura se muestra éste umbral dependiendo de la frecuencia.



*Figura 3.1 Umbral absoluto de audición*

Se puede observar cómo el oído es más sensible en el rango de frecuencias de 1 a 5 KHz, principalmente debido a la acción del oído externo. El umbral crece rápidamente tanto en baja como en alta frecuencia. El umbral auditivo varía según la frecuencia debido a la sensibilidad del oído humano.

Es posible obtener una buena aproximación de la dependencia del umbral con la frecuencia mediante la siguiente ecuación que fue estudiada por Fletcher [27] y aproximada por Terhardt [28]:

$$T_q(f) = 3.64 \left( \frac{f}{1000} \right)^{-0.8} - 6.5 e^{-0.6 \left( \frac{f}{1000} - 3.3 \right)^2} + 10^{-3} \left( \frac{f}{1000} \right)^4 \quad (6)$$

Donde  $f$  es la frecuencia expresada en KHz y el resultado está dado en dB. Las componentes en frecuencia que se encuentren bajo la curva de umbral absoluto de audición no pueden ser escuchadas por lo que se pueden discriminar en el momento de realizar la codificación. Se sabe que el rango de frecuencias audible por el oído oscila entre los 20Hz y 20 KHz [27].

### 3.3.2 Intensidad Sonora, Tono y Timbre

La intensidad sonora es un atributo de los sonidos en función del cual se pueden ordenar en una escala de más bajo a más alto en intensidad. Además de la potencia de un sonido la intensidad sonora depende también de la duración y la estructura en tiempo y frecuencia del mismo. En el caso de la frecuencia del sonido se definen contornos de la misma intensidad sonora, donde se toma la frecuencia de 1 KHz como referencia. La unidad en la que se mide la intensidad sonora es el fono. El umbral de silencio es un ejemplo de contorno de igual intensidad sonora.

El tono se define como la propiedad que permite ordenar los sonidos en una escala musical. Con el tono se aprecia el patrón de repetición de un sonido, así para el caso de un tono puro se relaciona con su frecuencia, y si se trata de un complejo armónico con la frecuencia fundamental del mismo.

El timbre es otra sensación auditiva, aunque no se puedan ordenar los sonidos en función del timbre en una escala unidimensional. La definición de timbre es negativa, es la

propiedad por la cual dos sonidos se distinguen como diferentes, aunque tengan la misma intensidad sonora y el mismo tono. En otras palabras, el timbre permite distinguir entre la misma nota tocada, por ejemplo, por un piano y una flauta. Desde un punto de vista psicoacústico, el timbre se detecta en el cerebro al estudiar la composición de señales en diferentes bandas críticas [25].

### 3.3.3 Bandas Críticas

Las ondas acústicas que viajan por el oído generan picos de respuesta en diferentes posiciones en base a la frecuencia que esté siendo percibida. Como consecuencia de esta transformación, la cóclea se entiende (desde el punto de vista del procesado digital de señales) como un banco de filtros. El ancho de banda, conocido como ancho de banda crítica, no es uniforme y se incrementa con la frecuencia, es por ello que, es posible distinguir tonos con pocos Hertz de separación a bajas frecuencias, mientras que para altas frecuencias son necesarios cientos de Hertz de separación para distinguir la diferencia de un tono a otro.

El ancho de las bandas críticas permanece aproximadamente constante de los 100 hasta los 500 Hz, y se incrementa en aproximadamente en un 20% de la frecuencia central por encima de los 500 Hz. Entonces las bandas críticas tienen un ancho aproximado de 100Hz para bajas frecuencias y de 4000Hz para altas frecuencias. Y en función a la existencia de bandas críticas, el umbral de detección de ruido de cuantificación depende únicamente de la potencia de la señal en un entorno de la misma.

El modelo psicoacústico calcula el nivel en el que el ruido comienza a ser perceptible para cada banda. En general, el ancho de las bandas críticas puede aproximarse por la siguiente expresión, como lo hace Zwicker, en su documento sobre el modelo psicoacústico:

$$BW_c(f) = 25 + 75 \left[ 1 + 1.4 \left( \frac{f}{1000} \right)^2 \right]^{0.69} \quad (7)$$

A la distancia entre dos bandas críticas adyacentes se le llama, Bark (en honor a Barkhausen) [26].

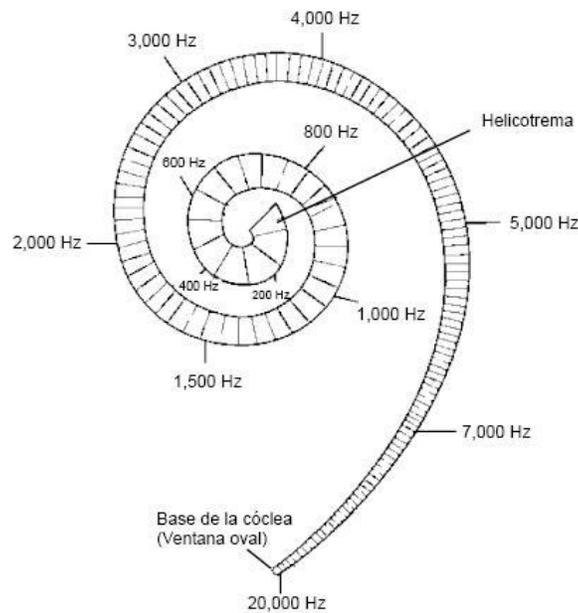
### 3.3.3.1 Escala de Bark

La escala de Bark resulta fundamental el modelado eficiente del oído. Para asemejar la forma en la que el oído trabaja, se utiliza esta escala no lineal que agrupa frecuencias en bandas, la cual describe una distancia fija a lo largo de la membrana basilar. Tomando en cuenta que el oído humano difícilmente percibe sonidos con una frecuencia superior a los 20000Hz, esto da como resultado 25 bandas críticas (Bark 0 a Bark 24) y basados en estudios clínicos que demuestran que la membrana basilar tiene una distancia de 32mm, esto indica que cada banda crítica representa aproximadamente 1.3mm de distancia a lo largo de la membrana basilar. De acuerdo con Eberhard Zwicker [29] el uso de la siguiente ecuación para convertir de kHz a Bark ha demostrado ser útil:

$$z(f)/Bark = 13 \tan^{-1} \left( \frac{0.76f}{1000} \right) + 3.5 \tan^{-1} \left( \frac{f}{75000} \right)^2 \quad (8)$$

Donde  $f$  es la frecuencia en KHz.

El oído humano percibe los sonidos mediante una descomposición en frecuencia realizada en la cóclea: oscilaciones a lo largo de la membrana basilar determinan que frecuencia es audible. La localidad en que esta oscilación tendría lugar depende de la frecuencia del sonido. En la figura 3.2 se muestra la cóclea y las localidades de esta en donde se perciben ciertas frecuencias.



*Figura 3.2 Distribución de frecuencias dentro de la cóclea*

### 3.3.4 Enmascaramiento

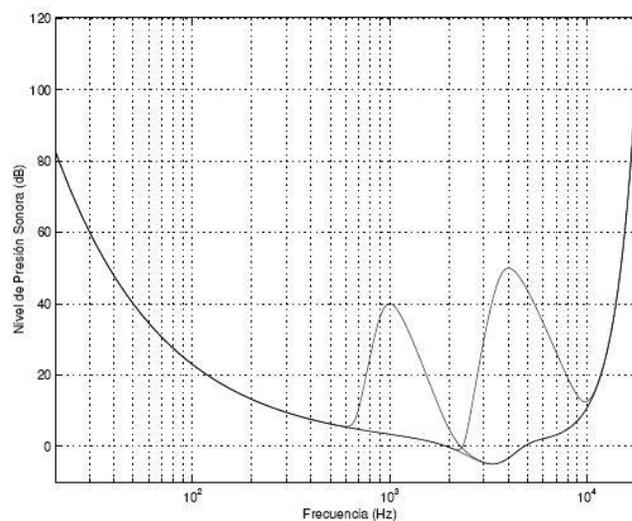
En psicoacústica el efecto más importante que debe ser tenido en cuenta en codificación de audio es el enmascaramiento. Se conoce como enmascaramiento al proceso por el cual el umbral de audición de un sonido crece por la presencia de otro sonido.

Es decir, ocurre cuando la presencia de una señal fuerte hace que una señal más débil cercana, en frecuencia o en el tiempo, no sea percibida. Hay dos tipos diferentes de enmascaramiento: el enmascaramiento simultáneo y el enmascaramiento temporal [25].

### 3.3.4.1 Enmascaramiento Simultáneo

Este fenómeno se produce cuando una señal débil pero audible (señal enmascarada) se hace inaudible ante otra señal más fuerte (señal enmascaradora), las cuales suceden de manera simultánea.

Por ejemplo, en la figura 3.3, el umbral de enmascaramiento obtenido a partir del umbral del silencio y del efecto de enmascaramiento producido por dos tonos puros localizados en 1 y 4KHz, todas las señales con nivel de presión sonora por debajo del umbral resultante y que sean simultáneas a estos dos tonos no serán audibles [26].



*Figura 3.3 Efecto de enmascaramiento de dos tonos*

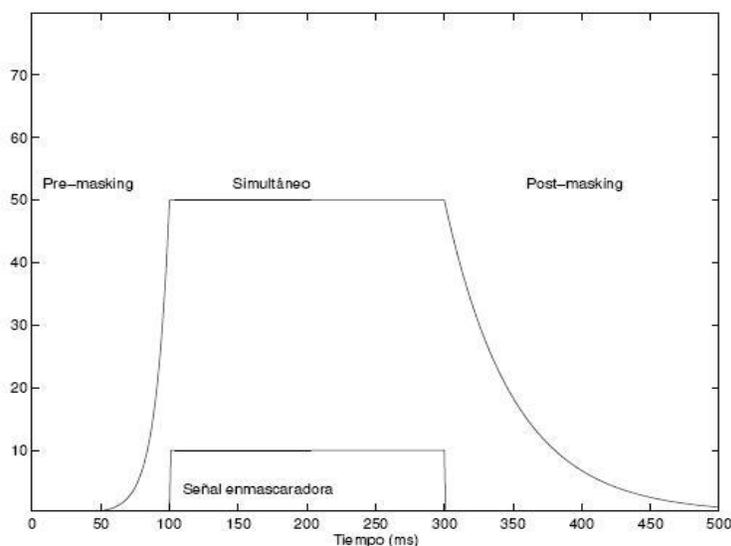
### 3.3.4.2 Enmascaramiento Temporal

Además del enmascaramiento simultáneo, el enmascaramiento puede extenderse en el tiempo y puede ocurrir antes y después de la generación del elemento enmascarador. Si el enmascaramiento sucede antes de la presencia del enmascarador se le denominará pre-enmascaramiento o enmascaramiento hacia atrás; mientras que sí el efecto se presenta

cuando éste, ya no esté presente se le conoce como post-enmascaramiento o enmascaramiento hacia adelante.

El origen del fenómeno de pre-enmascaramiento radica en el tiempo que ocupan los sentidos para ajustarse y percibir las sensaciones. El tiempo de duración de ambos efectos es diferente, pues mientras el pre-enmascaramiento dura aproximadamente 20ms, el post-enmascaramiento puede durar hasta 200ms. Este efecto da lugar al empleo de sistemas con resolución temporal limitada para lograr una codificación digital de alta calidad, tales como bancos de filtros con gran resolución en frecuencia.

A continuación se muestra una figura 3.4 que ilustra ambos efectos [27].



*Figura 3.4 Ejemplo del efecto pre-enmascarador y del post-enmascarador*

### 3.3.4.3 Umbral Global de Enmascaramiento

El umbral global de enmascaramiento es una curva que indica el nivel de presión de sonido necesario para que un componente espectral determinado sea percibido por el oído humano. Se forma a partir de las curvas de enmascaramiento de los componentes

espectrales, el enmascaramiento temporal y el umbral auditivo. El problema radica en la forma de combinarlos. Es posible que se desee sumar, sub-sumar, sobre-sumar o simplemente elegir el valor mayor de las máscaras en una frecuencia determinada. Cualquiera de estos casos puede ser descrito de acuerdo con la siguiente fórmula:

$$I_N[f] = \left( \sum_{n=0}^{N-1} I_n^\alpha[f] \right)^{\frac{1}{\alpha}} \quad (9)$$

Donde  $I_N[f]$  representa la intensidad de la curva de enmascaramiento resultante de combinar  $N$  curvas de enmascaramiento con intensidades  $I_n[f]$  en una frecuencia  $f$ . El parámetro  $\alpha$  define la forma en que las curvas se combinan. Si  $\alpha = 1$ , la ecuación equivale a sumar intensidad, mientras que si  $\alpha = +\infty$ , equivale a utilizar el máximo.

La curva resultante es el umbral global de enmascaramiento. La utilidad del umbral global de enmascaramiento es la de poder asignar ruido sin que este sea perceptible, en el caso de codificación perceptual, ruido de cuantización. Mientras mayor sea el umbral global de enmascaramiento con relación al componente espectral menor será el número de bits necesarios para cuantizarlo. En otras palabras, es deseable tener una razón señal a ruido superior a la razón señal a enmascarar. Es decir, ocurre cuando la presencia de una señal fuerte hace que una señal más débil cercana, en frecuencia o en el tiempo, no sea percibida. Hay dos tipos diferentes de enmascaramiento: el enmascaramiento simultáneo y el enmascaramiento temporal [28].

### 3.4 Codificación

La codificación es una representación de los datos y existen diferentes objetivos para los que es útil aplicarla. Un objetivo común es crear un flujo de información que sea resistente a errores, en cuyo caso es común que la tasa de datos sea incrementada. En

algunos otros casos se requieren que la codificación de la información sea ininteligible para cualquier dispositivo diferente que el receptor, como es el caso de la criptográfica.

Otro esquema de codificación, pretende reducir la cantidad de datos necesarios para reconstruir la información, en este caso, la codificación implica compresión [30].

A la hora de definir un sistema de codificación de audio, es necesario tener en cuenta los requisitos que se le piden. Dependiendo de la aplicación, algunos de ellos serán más relevantes que otros. Los principales criterios que se tienen en cuenta a la hora de diseñar un esquema de codificación perceptual son los siguientes:

- ◆ Eficiencia de compresión. En muchas aplicaciones, obtener la mayor tasa de compresión para la misma calidad de servicio se traduce directamente en ahorro de costes. Por tanto, la calidad de señal para una tasa binaria dada (o la tasa binaria necesaria para conseguir una cierta calidad de señal) es un criterio de diseño importante.
- ◆ Calidad de la señal decodificada. En algunas aplicaciones se precisa calidad transparente (no existe diferencia audible entre la señal original y la señal decodificada) o casi transparente. Para asegurar esta calidad el sistema de codificación debe superar pruebas de calidad muy exigentes. En otras aplicaciones, sin embargo, se permite que una persona entrenada distinga la señal original de la decodificada, aunque las distorsiones en la señal decodificada sean tolerables, se habla entonces de audio de alta calidad.
- ◆ Complejidad. Para aplicaciones de consumo, la complejidad de la codificación, y en especial de la decodificación, es importante, aunque conforme pasa el tiempo estos aspectos están pasando a un segundo plano. Podemos distinguir distintos tipos de complejidad:

- Complejidad computacional. Se refiere al número de instrucciones del procesador necesarias para tratar un bloque de muestras. Si el algoritmo de codificación se implementa en una arquitectura de cálculo de propósito general (PC o estación de trabajo), esta es la figura de complejidad más importante.
  - Requisitos de almacenamiento. Supone un factor de coste importante para realizaciones con dispositivos portátiles o bien en arquitecturas de propósito específico (DSP's).
  - Complejidad del codificador frente a la del decodificador. En la mayoría de los algoritmos que se describen en este capítulo, el codificador es más complejo que el decodificador. Esta asimetría es interesante para aplicaciones como la radiodifusión, donde existe una relación de uno a muchos entre el codificador y los decodificadores.
- ◆ Editabilidad. Desde el punto de vista de codificación el requisito de editabilidad está relacionado con el de break-in, y consiste en la posibilidad de comenzar la decodificación en cualquier punto de la secuencia de bits sin que esto suponga un elevado tiempo de sincronización. Como norma general, un codificador empieza a decodificar antes si no utiliza codificación diferencial entre tramas.

En particular, cuando se trata de compresión de audio es posible efectuarla de manera que en esta no existan pérdidas o que existan pérdidas y como en todo, cada esquema tiene ventajas y desventajas propias. La compresión sin pérdida logra mejores resultados si lo que se pretende es preservar la fidelidad del origen, a costa de un sacrificio en la relación de compresión.

La compresión con pérdida no siempre logra el objetivo de ser perceptualmente transparente, lo cual disminuye drásticamente la calidad (al menos desde un punto de vista subjetivo) de la señal reconstruida, aunque su gran ventaja son las altas relaciones de compresión [31].

### 3.4.1 Codificación Perceptual

La codificación perceptual de audio digital ha sido a lo largo de los últimos 20 años un campo de aplicación del procesado de señales. Durante este tiempo, se han resuelto algunos de los retos asumidos. Sin embargo, la creciente demanda de aplicaciones digitales en redes telemáticas hace, que aún hoy, la codificación de audio sea un tema de actualidad. El objetivo de este capítulo es describir las técnicas de codificación más utilizadas en el mercado y presentar una revisión en codificación perceptual de audio.

Pese a que el ancho de banda global disponible para la transmisión de señales de audio aumenta continuamente, así como la capacidad de los dispositivos de almacenamiento, siguen surgiendo campos de aplicación donde los actuales estándares de codificación no ofrecen una solución satisfactoria. En este sentido cabe destacar la necesaria reducción del régimen binario para la transmisión de audio por internet, o telefonía móvil, manteniendo una alta calidad, lo cual ha provocado el desarrollo de la codificación paramétrica de audio. Pero, en un futuro próximo, tal y como adelanta MPEG, van a seguir apareciendo nuevas aplicaciones para el tratamiento digital de audio como, por ejemplo, la búsqueda basada en contenido.

Una combinación entre la forma de muestrear, la forma de inventanar y la forma de almacenar estos datos se conoce como un codificador de audio. La función tasa-distorsión determina el régimen binario mínimo que se puede conseguir para una buena codificación, que normalmente se consiguen muy buenos resultados combinando la eliminación de redundancia (datos que pueden reconstruirse a partir de los presentes), con la eliminación de datos que no son importantes (eliminación de irrelevancia).

Es por esto que el tipo de compresión a utilizar en este trabajo es la compresión con pérdida, que llamado de otra forma se le dirá codificación perceptual, que se centra en la eliminación de aquellos datos que son irrelevantes para el sistema auditivo. La señal se codifica de forma que la distorsión debida a la codificación no sea audible o, por lo menos,

en que la distorsión que se produzca sea mínima para el régimen binario objetivo. Para tener éxito en esta tarea, es preciso aplicar el conocimiento disponible sobre el funcionamiento del sentido del oído.

Los sistemas tradicionales de digitalización y codificación de señal son insuficientes para el audio debido a la alta tasa binaria que se generaría. Se han desarrollado, pues, métodos de compresión para reducir el tamaño de los ficheros y, por tanto, la tasa de bit requerida para su posterior transmisión en medios de ancho de banda limitado.

Algoritmos importantes, como MPEG, se basan en la llamada compresión con pérdidas (lossy compression), consistente en la eliminación de la parte de señal que no es audible por el oído humano. Para ello, hace uso de dos fenómenos:

- ◆ Parte de la señal queda por debajo del umbral de audibilidad humano.
- ◆ Ciertos sonidos enmascaran a otros e impiden su audibilidad.

Para esto se debe contar con el estudio de algunas transformadas que ayudan a pasar el audio digital basado en el tiempo a una señal que se maneje en la frecuencia para poder asemejar con exactitud la compresión de audio enfocada al comportamiento del oído humano, donde se presenta el estudio de la transformada rápida de Fourier y la transformada coseno modificada [30].

### **3.5 Algoritmo para la Transformada Rápida de Fourier (FFT)**

La compresión que otorga el formato MP3 se basa en aprovechar las limitaciones del oído humano desechando lo que tiende a ser inaudible, y en simplificar las redundancias de sonido. La compresión en el caso de las grabaciones de voz tiene base en las propiedades y limitaciones de la voz humana. Tomando en cuenta el procedimiento básico para la compresión de audio es el siguiente:

Se toman muestras en un intervalo de tiempo, donde a este intervalo temporal se le denomina ventana de análisis, se divide la señal de audio en bandas de frecuencias donde hay que calcular un umbral de enmascaramiento para cada una de estas bandas. Generalmente se usa una FFT (Transformada rápida de Fourier).

Para entender mejor que es la FFT, se hace referencia a la DFT, que es La Transformada Discreta de Fourier (DFT del inglés Discrete Fourier Transform) que es el equivalente discreto de la Transformada de Fourier donde se ha transformado la variable continua  $t$  por la variable discreta  $nTs$  siendo  $Ts$  el periodo de muestreo. Recordemos que la Transformada de Fourier de una señal analógica  $x(t)$  es como se muestra en la ecuación:

$$x(w) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j\omega t} dt \quad (10)$$

La Transformada Discreta de Fourier es un método muy eficiente para determinar el espectro en frecuencia de una señal. Permite convertir una secuencia de valores en el dominio del tiempo a una secuencia de valores equivalente en el dominio de la frecuencia. La Inversa de la Transformada Discreta de Fourier (IDFT) realiza el proceso contrario. Recordemos el par de ecuaciones de la DFT, es como se muestra en la ecuación 11(a) y 11(b):

$$x(k) = \sum_{n=0}^{N-1} x(n) \cdot W^{nk} \quad k = 0, 1, \dots, N-1 \quad (11a)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k) \cdot W^{-nk} \quad n = 0, 1, \dots, N-1 \quad (11b)$$

Donde las constantes  $W$  son conocidas como factores twiddle y definidas como la siguiente ecuación:

$$W = e^{-j2\pi/N} \quad (12)$$

Observar que  $W$  es una función de longitud  $N$ , por ello, también suele expresarse como  $WN$ .

El inconveniente de realizar unos algoritmos que implementen tal cual estas fórmulas es la cantidad de tiempo requerido para computar la salida. Esto es debido a que los índices  $k$  y  $n$  deben variar de 0 a  $N-1$  para conseguir el rango de salida completo y, por tanto, se deben realizar  $N^2$  operaciones. Por esto se debe implementar un algoritmo eficiente que realice la DFT. Este algoritmo se basa en la Transformada Rápida de Fourier. Un algoritmo para la FFT obtiene los mismos resultados que la DFT pero más rápidamente debido a que reduce el número de cálculos requerido para realizar la DFT.

El término genérico Transformada Rápida de Fourier abarca distintos algoritmos con distintas características, ventajas y desventajas. En la fórmula de la Transformada Discreta de Fourier obtener  $x(k)$  para un  $k$  determinado requiere aproximadamente  $N$  sumas complejas y  $N$  productos complejos, ya que:

$$x(k) = x(0) + x(1) \cdot W^k + x(2) \cdot W^{2k} + x(3) \cdot W^{3k} + \dots + x(N-1) \cdot W^{(N-1)k}$$

Para  $k = 0, 1, \dots, N-1$ . Si lo que se desea es obtener  $x(0), x(1), \dots, x(N-1)$  entonces se necesitarán un total de aproximadamente  $N^2$  sumas complejas y  $N^2$  productos complejos. Esto quiere decir que los requerimientos computacionales de la DFT pueden ser excesivos especialmente si el tamaño de  $N$  es grande. La FFT aprovecha la periodicidad y simetría del factor twiddle  $W$  para el cálculo de la Transformada Discreta de Fourier. La periodicidad de  $W$  implica y respectivamente su simetría:

$$W^k = W^{k+N} \qquad W^k = -W^{k+N/2}$$

La FFT descompone la DFT de  $N$  puntos en transformadas más pequeñas. Una DFT de  $N$  puntos es descompuesta en dos DFT's de  $(N/2)$  puntos. Cada DFT de  $(N/2)$  puntos se descompone a su vez en dos DFT's de  $(N/4)$  puntos y así sucesivamente. Al final de la descomposición se obtienen  $(N/2)$  DFT's de 2 puntos cada una. La transformada más

pequeña viene determinada por la base de la FFT. Para una FFT de base 2,  $N$  debe ser una potencia de 2 y la transformada más pequeña es la DFT de 2 puntos [32].

Y con el fin de implementar en forma práctica la Transformada Discreta de Fourier mediante el uso de computadores, a mediados de la década del sesenta J.W Cooley y J.W Tukey desarrollaron un algoritmo denominado la Transformada rápida de Fourier (FFT).

El formato MP3 usa la FFT para ir un poco más a fondo, en la compresión MP3 se utilizan básicamente dos procesos en paralelo donde se eliminan ciertas frecuencias que tienden a no afectar mucho la percepción general del resultado por el oído humano.

Estos procesos se desarrollan con algoritmos basados en la Transformada de coseno discreta es una transformada basada en la Transformada de Fourier discreta, pero utilizando únicamente números reales y La Transformada rápida de Fourier FFT es la abreviatura usual de un eficiente algoritmo que permite calcular la transformada de Fourier discreta (DFT) y su inversa, usando una FFT de 1024 puntos para conseguir una buena resolución de frecuencia y poder calcular correctamente los umbrales de enmascaramiento [33].

Antes de la FFT, se aplica una ventana, a la cual se le llama “ventana  $w(x)$ ” convencional, que es muy similar a la ventana hamming. La ventana  $w(x)$  se realizó en MATLAB, y se utiliza para evitar las discontinuidades en los extremos de la señal, y poder realizar la suma adecuada de cada ventana sin que exista pérdida alguna en el momento de recuperación de la señal original, propiciando que los lóbulos inferiores al principal caigan más rápido y sin perturbaciones, es decir que exista menor ruido en la señal que se desea recuperar, como se mostrara en la siguiente figura. La salida de la FFT se usa primero para analizar qué tipo de señal está siendo procesada: una señal estacionaria hace que el modelo escoja bloques largos, y una señal con muchos transitorios da como resultado bloques cortos. El tipo de bloque se usa luego en la parte MDCT del algoritmo.

Después de esto, el modelo psicoacústico calcula el mínimo umbral de enmascaramiento para cada subbanda. Estos valores de umbral se usan luego para calcular la distorsión permitida. El modelo pasa entonces las distorsiones permitidas a la sección de repartición de ruido, el codificador para uso posterior. Al aplicar esta transformación, el espectro se divide en bandas de anchura creciente con la frecuencia, lo que simula el comportamiento del oído, que tiene más resolución espectral en baja frecuencia. La FFT elimina información redundante que existe en la DFT, ya que está explota las propiedades de periodicidad y simetría.

Otro punto importante es que el algoritmo FFT trabaja en forma más eficiente cuando lo hace sobre una señal donde el número de muestras  $N$  es una potencia de 2. El principio de la FFT se basa en el método denominado “divide y conquista”, ya que divide la señal de  $N$  puntos en dos secuencias de datos de puntos, la señal de entrada o salida respectivamente, según el tipo de algoritmo.

El algoritmo que sigue MATLAB para conseguir realizar la FFT es:

$$X(k) = \sum_{n=1} x(n) \cdot \exp(-j \cdot 2 \cdot \pi \cdot (k-1) \cdot (n-1) / N), \quad 1 \leq k \leq N.$$

la inversa DFT (en la computadora IFFT) esta dad por:

$$x(n) = (1/N) \sum_{k=1}^N X(k) \cdot \exp(j \cdot 2 \cdot \pi \cdot (k-1) \cdot (n-1) / N), \quad 1 \leq n \leq N.$$

Por lo tanto, se utilizará la FFT de Matlab para la rápida transformación de la señal en el proceso de seguir el modelo psicoacústico [32].

### 3.6 Transformada Discreta de Coseno Modificada (MDCT)

La percepción humana del sonido se efectúa por frecuencia, por lo que es necesario llevar la señal de sonido en el dominio del tiempo a una representación en el dominio de la frecuencia. Una transformada que cumple con este requerimiento es la Transformada Discreta de Coseno Modificada (MDCT), que es ampliamente utilizada.

En esta parte del trabajo se dedica a realizar un algoritmo que sea capaz de crear en Matlab La Transformada Discreta de Coseno Modificada (MDCT), ya que es muy importante en la compresión de audio donde existen muchas etapas de esta, y para que tenga una máxima compresión del ancho de banda que proporcionan los filtros, ya que es demasiado amplio para las bajas frecuencias, y demasiado estrecho para las altas frecuencias; así que el número de bits del cuantizador no se puede optimizar para la sensibilidad al ruido dentro de cada banda crítica. Entonces, lo mejor es que al espectro audible se le hagan particiones en bandas críticas (por medio de la transformada MDCT) que reflejen la selectividad en frecuencia del oído humano.

Esta transformada, es del tipo de Cancelación de Aliasing en el Dominio del Tiempo (TDAC por sus siglas en ingles) que utiliza bloques con traslape para poder reconstruir la señal, la MDCT requiere de una técnica de traslape y suma para tener una reconstrucción exacta [33].

La MDCT ofrece reconstrucción exacta, pues lleva los datos a una representación en el dominio de la frecuencia. Además, la MDCT utiliza la técnica de traslape en un 50 % y suma; es decir la Transformada Discreta de Coseno Modificada (MDCT) es críticamente muestreada lo que significa que, aunque se traslapa un 50%, la secuencia de datos después de la transformada, tiene el mismo número de coeficientes como muestras antes de la transformada, (después de ser sobrepuesta y adherida). Es decir, un simple bloque de datos IMDCT (transformada inversa) no corresponde al bloque original en el cual la MDCT fue

realizada. Cuando se suman bloques subsecuentes de la transformada inversa (siempre utilizando una sobreposición del 50%), los errores introducidos por la transformada se cancelan. Gracias a la sobreposición, la MDCT es muy factible para cuantificación.

Una definición un poco modificada de MDCT se indica en la siguiente expresión:

$$x(m) = \sum_{k=0}^{n-1} f(k)x(k) \cos\left(\frac{\pi}{2n}\left(2k+1+\frac{n}{2}\right)(2m+1)\right), \text{ para } m = 0, \dots, \frac{n}{2}-1 \quad (13)$$

En codificación la MDCT es realizada con tramos de 512, de esta manera 256 nuevos muestreos son utilizados para cada bloque [34].

Es por esto que se realiza en MATLAB un programa que desarrolle la MDCT que corresponde a la parte del proceso donde se evalúa el contenido de la información de la señal de audio por zonas espectrales y por unidad de tiempo ello quiere decir que por unidad de tiempo determinado y en una zona determinada del espectro de audio se determina lo siguiente: cantidad de información que contiene, de que frecuencia es dicha información y contenido armónico de la misma, las magnitudes obtenidas por este procedimiento, serán los datos transmitidos al siguiente bloque de compresión.

La IMDCT dada por la siguiente expresión:

$$y(p) = f(p) \frac{4}{n} \sum_{m=0}^{\frac{n}{2}} x(m) \cos\left(\frac{\pi}{2n}\left(2p+1+\frac{n}{2}\right)(2m+1)\right), \text{ para } p = 0, \dots, n-1 \quad (14)$$

Para poder reconstruir la señal original es necesario que los bloques a los que se les aplique la MDCT estén traslapados en un 50 %, y posteriormente, sumar las señales generadas por las IMDCT (en las partes que estas se traslapan). Esta técnica es conocida como cancelación de aliasing en el dominio del tiempo.

Para cumplir con los requerimientos de reconstrucción exacta, es necesario que dichas ventanas cumplan con la siguiente condición; las ventanas de análisis y de síntesis deben ser inversas en tiempo entre sí en la porción en que se traslapan.

Cabe hacer hincapié en que es necesario multiplicar  $x$  por una ventana de síntesis válida para completar el proceso apropiadamente. La ventana de  $UP(X)$  tiene como objetivo suavizar los bordes de la ventana para evitar que ésta presente un corte brusco. Por otra parte es posible observar el efecto de la MDCT, la cual reduce la señal considerablemente, o solo en la amplitud de las muestras sino en la cantidad de las mismas. La MDCT reduce en un este caso en un 50% la cantidad de muestras de cada ventana. Al realizar el proceso inverso de esta se obtendrá exactamente la misma señal [35].

### 3.7 Enventanado

El proceso de enventanado o framing es utilizado como primer paso en la compresión de audio para poder editar las canciones de audio con mayor facilidad. Si no dividimos la señal, para poder tratar la canción, sería necesario descomprimir toda la señal, lo cual es un inconveniente en cuanto a tiempo y carga computacional se refiere. Desde este punto de vista, es necesario tener bloques (o frames) lo más pequeños posibles. Sin embargo, es importante notar que cada frame tendrá que tener asociada una cabecera con información para la descompresión, por lo que bloques muy pequeños generarían un exceso de cabeceras, dando como resultado algo que, en ciertos casos, puede neutralizar el propósito de la compresión. Por otro lado, frames muy grandes llevan asociados los problemas de editabilidad que ya hemos mencionado. Por tanto, es necesario llegar a una solución de compromiso.

Matemáticamente, enventanar una señal discreta consiste en multiplicarla por una ventana, que es una señal que toma valores no nulos únicamente en un intervalo  $0 \leq n \leq N-1$ . Existen numerosos tipos de ventanas: rectangular, triangular, Hanning, Hamming, Blackman, etc. Desde el punto de vista puramente teórico, lo ideal es tener

ventanas de longitud elevada, ya que, de esta forma, las señales enventanadas son más parecidas a las reales, y aumenta la resolución frecuencial del sistema. Por otro lado, las ventanas tienen propiedades muy variadas, por ejemplo, la rectangular es la que aporta mayor resolución en frecuencia, si bien también es la de mayores lóbulos laterales.

Es por estas razones que en este trabajo se utiliza funciones de ventana con funciones atómicas (AF), dicho termino fue introducido en 1975 por V.L. and V.A. Rvachev. Estas funciones presentan un mejor desempeño en el análisis armónico. Algunas de las ventajas de estas funciones sobre los polinomios trigonométricos algebraicos clásicos, como en el caso de las típicas ventanas, es que éstos solo satisfacen ecuaciones diferenciales lineales homogéneas con coeficientes constantes, que implican un mayor procedimiento y trabajo.

Las funciones atómicas por su parte satisfacen una forma más general de estas ecuaciones diferenciales, lo cual hace que el rizo de corte sea mucho más suave que las anteriores y con una pequeña curvatura en la superficie de los valores máximos de ella con el fin de que esta al dividir la canción en ficheros y al ser traslapados entre ellos no exista perdida alguna de la información. Y en su debido momento de recuperación está también puede recuperarse por completo a la señal original.

En este caso se utilizara la función de ventana  $up(x)$ . Por definición, las Funciones Atómicas son una solución compacta infinitamente diferenciable a la ecuación diferencial con argumento inverso, es decir, como se define a continuación:

$$Lf(x) = \lambda \sum_{k=1}^M c(k) f(ax - b(k)), \quad |a| > 1 \quad (15)$$

Donde  $L$  es un operador diferencial lineal con coeficientes constantes. Si  $a = 1$  y  $b(k) \equiv 0(k = 1, M)$  donde se convierte en una ecuación diferencial ordinaria.

La AF más simple y más importante se genera por medio de la convolución infinita de impulsos rectangulares. Con la ayuda de la Transformada de Fourier el impulso rectangular puede ser representado en la forma siguiente:

$$\varphi(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \frac{\sin(u/2)}{u/2} du \quad (16)$$

Realizando una convolución de  $n$  impulsos rectangulares variables:

$$\theta_n(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \left( \frac{\sin(u/2)}{u/2} \right)^{n+1} du \quad (17)$$

Así, el resultado de tal operación es una nueva función compacta determinada en el intervalo  $[-1,1]$ , que también satisface la ecuación 15.

La primera vez que este problema fue indicado en 1967 por V.L. Rvachev y su solución fue denotada por la función  $up(x)$ , tiene la siguiente representación en base a la transformada de Fourier:

$$up(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \prod_{k=1}^{\infty} \frac{\sin(u \cdot 2^{-k})}{u \cdot 2^{-k}} du, \quad [-1,1] \quad (18)$$

La propiedad más útil de las funciones atómicas está en la posibilidad para representar algún polinomio por medio de sus traslaciones de la función, mientras que las traslaciones de otro tipo de funciones infinitamente diferenciables no pueden formar incluso un polinomio de orden cero (función constante).

Otra característica importante es que mientras para algunas funciones el cálculo de la Transformada de Fourier es difícil para las funciones atómicas esta transformada se sabe explícitamente

Algunas propiedades de esta función  $up(x)$ :

Es una función par, es decir,

$$up(x) = up(-x) \quad (19)$$

$$up(x) = 1 - up(1-x) \quad (19)$$

$$up(0) = 1 \quad (21)$$

$$\int_{-1}^1 up(x) dx = 1 \quad (22)$$

El espectro de  $up(x)$  es una función par de valor real de tipo exponencial, rápidamente amortiguada y teniendo ceros en los puntos divisibles por  $2\pi$ . El primer lóbulo lateral tiene una magnitud de -23.5 dB.

La función  $up(x)$  no es analítica para todo el intervalo: o la serie de Taylor tiene un radio de convergencia cero o converge a otra función. Esto es porque uno no puede usar una serie de potencias ordinaria para representar  $up(x)$ .

Sin embargo, después de su extensión con período  $2\pi$  sobre el eje real, la función  $up(x)$  tiene una expresión rápidamente convergente en serie de Fourier con respecto a armónicos uniformes. La siguiente expresión representa la aproximación utilizada para obtener la función ventana  $up(x)$ .

$$w(n) = \frac{1}{2\pi} \int_{-150}^{150} e^{jun} \prod_{k=1}^{50} \frac{\sin(u \cdot 2^{-k})}{u \cdot 2^{-k}} du \quad (23)$$

La amplitud máxima de los lóbulos laterales es de 23.2 dB por debajo del lóbulo principal como se muestra en la figura 3.5, con su respectivo espectro de frecuencias que se muestra en la figura 3.6 [36].

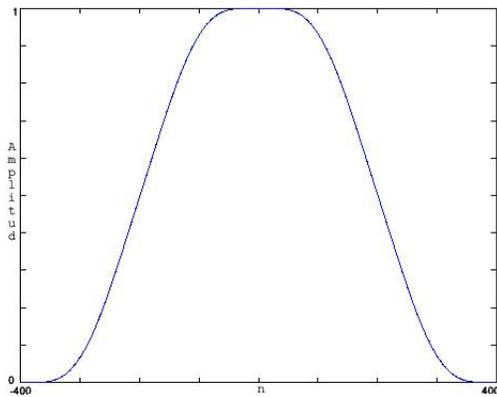


Figura 3.5 Función  $up(x)$

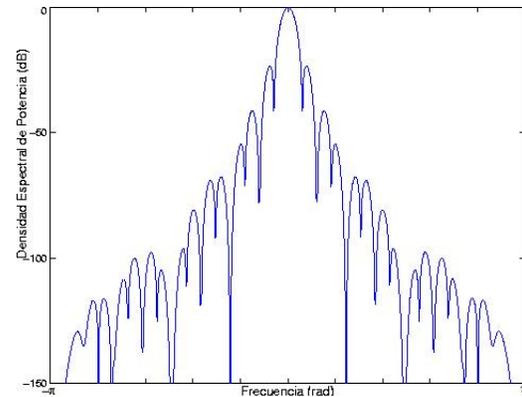


Figura 3.6 Espectro de ventana  $up(x)$

### 3.8 Encabezado de Tramas

No existe encabezado principal de archivo en el formato de audio MPEG y en especial cuando se habla de MPEG Layer 3 (MP3). En éste el encabezado es individual para cada trama (fragmento de archivo).

Cuando se quiere leer información de un archivo MP3, usualmente es suficiente con encontrar la primera trama, leer su encabezado y asumir que las otras tramas son iguales. Pero éste no es siempre el caso; por ejemplo, existen algunos archivos con tasas de bits variables, donde cada trama posee su propia tasa de bits. Esto se hace con el fin de mantener constante la calidad del sonido durante todo el archivo. Otro método usado para mantener constante la calidad de sonido es emplear más bits, en las partes donde se necesite [37].

El encabezado de la trama tiene la siguiente presentación, con las posiciones para cada uno de los 32 bits, como se muestra en la figura 3.6:

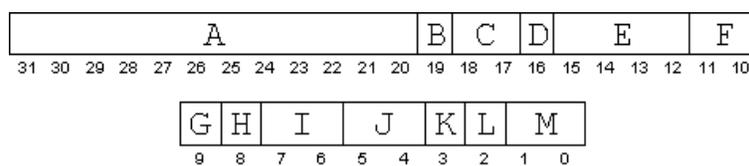


Figura 3.6 Encabezado de la trama

**A:** syncword. Con 12 bits de longitud, todos en '1' para identificar el comienzo de la trama.

**B:** ID. Un (1) bit usado para identificación del audio. Siempre en '1', para indicar que se trata de audio MPEG-1.

**C:** Layer. Dos (2) bits usados para descripción de la capa. Para identificar cuál esquema fue usado durante la codificación del audio, tomando en cuenta la tabla 3.1.

00	Reservado
01	Capa III
10	Capa II
11	Capa I

*Tabla 3.1 descripción de capa*

**D:** protection\_bit. Un (1) bit de protección. Si está en '0' indica que la trama está protegida por un código de redundancia cíclica para detección de errores.

**E:** bitrate\_index. Cuatro (4) bits para proporcionar el índice de la tasa de bits, de acuerdo con la tabla 3.2.

Código	Tasa de bits MPEG-1 (Kbps)		
	Capa I	Capa II	Capa III
0000	Formato libre	Formato libre	Formato libre
0001	32	32	32
0010	64	48	40
0011	96	56	48
0100	128	64	56
0101	160	80	64
0110	192	96	80
0111	224	112	96
1000	256	128	112
1001	288	160	128
1010	320	192	160
1011	352	224	192
1100	384	256	224
1101	416	320	256
1110	448	384	320
1111	No permitido	No permitido	No permitido

**Nota:** Si la trama usa formato libre (una tasa de bits diferente a las listadas), la tasa debe permanecer constante, y debe ser menor a la máxima tasa de bits permitida (320 Kbps para la Capa III).

*Tabla 3.2 Descripción de capa  
bitrate\_index*

**F:** `sampling_frequency`. Dos (2) bits que indican la tasa de muestreo, como se observa en la tabla 3.3.

00	44.1 KHz
01	48 KHz
10	32 KHz
11	Reservado

*Tabla 3.3 Tasa de muestreo*

**G:** `padding_bit`. Un (1) bit usado para relleno. Si está en '1' la trama se rellena con una ranura extra. Únicamente se usa para frecuencias de 44.1 KHz. Por ejemplo, un sonido 128 Kbps 44.1 KHz Capa II usa muchas tramas de 418 bytes de largo y unas pocas de 417 bytes para cumplir exactamente la tasa de transferencia de 128 Kbps. La ranura consume 8 bits (1 byte) para las Capas II y III.

**H:** `private_bit`. Un (1) bit para uso privado. No se usa generalmente.

**I:** `mode`. Dos (2) bits que indican el modo de canal, tal y como se muestra a continuación en la tabla 3.4.

00	<i>Stereo</i>
01	<i>Joint Stereo</i>
10	<i>Dual Channel</i> (2 canales monofónicos independientes)
11	<i>Single Channel</i> (1 canal monofónico)

*Tabla 3.4 Modo de canal*

En el modo Stereo indica que el canal comparte bits, pero no usa codificación Joint Stereo. En el modo Joint Stereo sí se saca provecho de la correlación existente entre los dos canales para representar más eficientemente la señal. El modo Dual Channel está conformado por dos canales mono totalmente independientes (cada uno es un archivo de audio diferente); cada canal usa exactamente media tasa de bits del archivo. La mayoría de los decodificadores los procesan como estéreo, pero no es siempre el caso. Single Channel consiste en un único canal de audio.

**J:** mode\_extension. Dos (2) bits indicando extensión al modo; sólo se usa en modo Joint Stereo. La extensión al modo se usa para información que no es de ninguna utilidad en el efecto estéreo. Estos bits se determinan dinámicamente por un codificador en el modo Joint Stereo, y este modo puede cambiar entre tramas, o incluso se puede dejar de usar en algunas tramas. En la Capa III, estos dos bits indican qué tipo de codificación Joint Stereo se está usando, Intensidad estéreo o Estéreo M/S. Estéreo M/S se refiere a transmitir los canales normalizados Middle/Side (Suma/Diferencia) de los canales izquierdo y derecho en lugar de los habituales Izquierdo/Derecho. En el lado del codificador los canales habituales se reemplazan usando la siguiente expresión:

$$M_i = \frac{\sqrt{2}}{2}(L_i + R_i) \quad \text{y} \quad S_i = \frac{\sqrt{2}}{2}(L_i - R_i) \quad (24)$$

$M_i$  = Middle;  $S_i$  = Side;  $L_i$  = Izquierdo;  $R_i$  = Derecho

Los valores  $M_i$  se transmiten por el canal izquierdo y los valores  $S_i$  se transmiten por el canal derecho. En el lado del decodificador los canales izquierdo y derecho se reconstruyen con la siguiente expresión:

$$L_i = \frac{M_i + S_i}{\sqrt{2}} \quad \text{y} \quad R_i = \frac{M_i - S_i}{\sqrt{2}} \quad (25)$$

Intensidad estéreo se refiere a retener en las frecuencias superiores a 2 KHz sólo la envolvente de los canales izquierdo y derecho.

El código indica que tipo de extensión al modo se está usando de la siguiente manera de acuerdo a la tabla 3.5:

<b>Código para la Capa III</b>		
Código	<i>Intensity stereo</i>	<i>M/S stereo</i>
00	no	no
01	sí	no
10	no	sí
11	sí	sí

Tabla 3.5 Código para capa III

**K:** copyrighth. Un (1) bit usado para copyright. Tiene el mismo significado que el bit de copyright en CD y cintas DAT, indicar que es ilegal copiar el contenido del archivo si el bit está en '1'.

**L:** original/copy. Un (1) bit usado para indicar si se trata de un medio original, si el bit está puesto en '1'. En '0' indica que es una copia del medio original.

**M:** emphasis. Dos (2) bits usados para información del énfasis. Le indica al decodificador que el sonido debe ser "re-ecualizado" después de una supresión de ruido tipo Dolby. Se usa raramente y se codifica como se muestra en la tabla 3.6.

00	Ninguna
01	50/15 ms
10	Reservado
11	CCITT J.17

*Tabla 3. 6 Codificación de énfasis*

Chequeo de errores: Si el bit de protección en el encabezado es igual a '0', se incluye un CRC de 16 bits después del encabezado. Si el bit de protección está en '1', no hay chequeo de errores y estos bits pueden ser usados para los datos de audio. El método para detección de errores que se utiliza es CRC-16, cuyo polinomio generador es la ecuación 26 [38].

$$CRC_{16} = x^{16} + x^{15} + x^2 + 1 \quad (26)$$

# Capítulo 4

## Resultados de Procesamiento Digital de Señales

Un codificador-decodificador perceptual explota las características del oído humano para comprimir una señal. La finalidad de este trabajo es sencilla: obtener una tasa de transferencia mínima con la mejor calidad posible. Para poder cumplir con ambos requerimientos, es indispensable contar con modelos que asemejen el comportamiento del oído humano y así poder eliminar información sin que esta pérdida sea perceptible.

Es deseable que al reproducir un archivo codificado de audio se tenga una pequeña porción de datos decodificados listos para ser enviados a un dispositivo de reproducción. Por lo que es deseable que en cualquier instante se tenga el número de datos que un dispositivo de reproducción requiere.

Es necesario entonces, ofrecer flexibilidad para que un programa reproductor tome esta decisión. En codificación perceptual es conveniente mantener los datos en una representación en el dominio de la frecuencia, pues es posible modelar con mejor precisión el oído. El audio digital, está representado en tiempo, por lo que se debe utilizar algún tipo de transformación. Los codificadores que llevan los datos a una representación diferente y los comprimen se conocen como codificadores de transformación.

Una opción posible es utilizar los coeficientes de una transformada Rápida de Fourier (FFT por sus siglas en inglés). Sin embargo, para tener una buena resolución en tiempo y poder ofrecer flexibilidad en el proceso de decodificación, es necesario dividir la señal en bloques. Además, para hacer un análisis más preciso en el dominio de la frecuencia, es necesario utilizar una ventana para que los bordes de cada bloque tengan una transición suave hacia

cero. Esta transición previene un corte abrupto que introduce aliasing en el dominio de la frecuencia.

Al momento de calcular la inversa de la FFT, la señal obtenida correspondería al bloque multiplicado por la ventana utilizada en la fase de análisis. Para obtener la señal original uno podría pensar en dividir por la ventana, puesto que ésta es conocida en todo momento. Al hacer esto, los errores generados por los coeficientes de la FFT se amplificarían en gran medida cerca de los bordes (donde la ventana tiene un valor cercano a 0), generando ruido perceptible por el oído humano.

Una técnica para resolver este problema es hacer que bloques adyacentes tengan un traslape. Al momento de síntesis se efectúa traslape y suma de éstos. Se pretende que la señal original pueda ser reconstruida exactamente. Esto elimina la necesidad de dividir por valores cercanos a 0, haciendo que el ruido de cuantización no sea amplificado. Sin embargo, esto crea restricciones en el tipo de ventanas que se pueden utilizar.

La inversa de la FFT requiere  $N$  coeficientes para reconstruir  $N$  muestras. Asumiendo un traslape de 50% entre bloques adyacentes, si se pretende reconstruir un bloque con traslape y suma, serán necesarios  $2N$  coeficientes de la FFT. Lo que nos dice que estamos doblando efectivamente la cantidad de datos necesarios para reconstruir nuestra señal. Aun que se comprobó que al utilizar un traslape menor es posible a costa de aumentar los artefactos audibles. Una transformada que resuelve todos estos problemas es la Transformada Discreta Coseno Modificada (MDCT), utilizada en codificadores como el MP3.

La MDCT ha sido ampliamente adoptada pues matemáticamente ofrece reconstrucción exacta, lleva los datos a una representación en el dominio de la frecuencia y está muestreada críticamente. El muestreo mantiene fija la cantidad de datos para reconstruir una señal, en otras palabras, para reconstruir una señal de tamaño  $N$  son necesarios  $N$  coeficientes de la MDCT. Además, la MDCT utiliza la técnica de traslape (en un 50 %) y suma.

Para hacer un análisis psicoacústico de mayor precisión, es común que se utilice una FFT, aunque en este caso se efectuó todo el proceso exclusivamente con la MDCT, para la capa 3 del estándar MPEG 1 Layer 3.

A continuación, se darán a conocer los resultados obtenidos con detalle de las fases que se requieren para realizar una compresión con MDCT para un codificador de transformación perceptual.

### 4.1 Fase de Enventanado

Se deben de tomar en cuenta diferentes consideraciones para obtener una ventana que cumpla con las características requerida para que en conjunto con la MDCT y su inversa sea posible obtener la reconstrucción exacta de la señal.

- ◆ Su espectro en frecuencia debe introducir el menor ruido posible al sistema.
- ◆ Las ventanas de análisis y síntesis deben ser inversas en el tiempo.

Para determinar la ventana que cumpla con estas condiciones, se compararon los espectros de varias de ellas, se eligieron estas ventanas ya que son las más comúnmente utilizadas en la compresión de audio digital

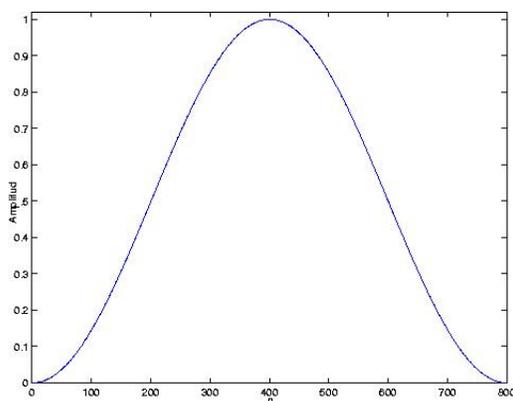


Figura. 4.1. Ventana de Hanning.

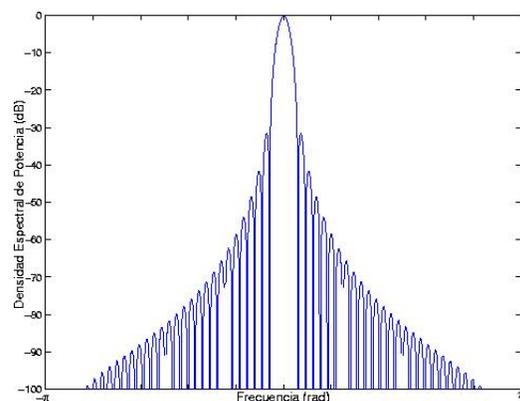


Figura. 4.2. Espectro de ventana de Hanning.

Esta ventana tiene una caída abrupta en sus lóbulos laterales, por lo que llegará a introducir aliasing al sistema. Las figuras 4.3 y 4.4 corresponden a la ventana Hamming y su espectro.

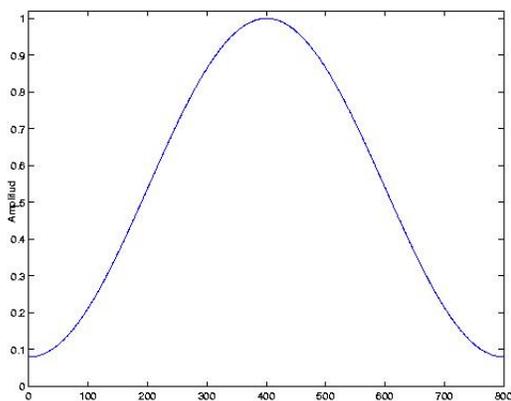


Figura 4.3 Ventana de Hamming.

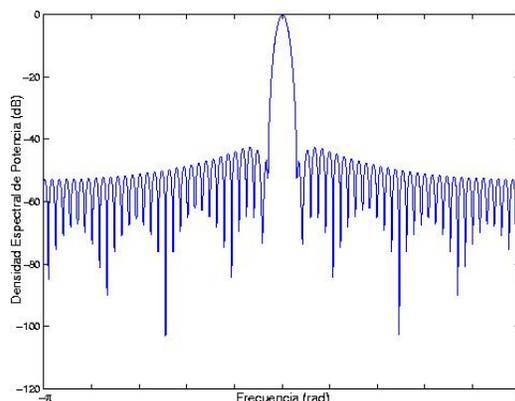


Figura 4.4 Espectro de ventana de Hamming.

La ventana anterior nunca llega a cero por lo que no alcanzaría a cubrir completamente al bloque de la señal, además debido a su problema, sus lóbulos laterales se expanden infinitamente en la frecuencia.

Finalmente, y entre las ventanas clásicas se muestran en las figuras 4.5 y 4.6 la ventana Kaiser-Bessel y su respectivo espectro.

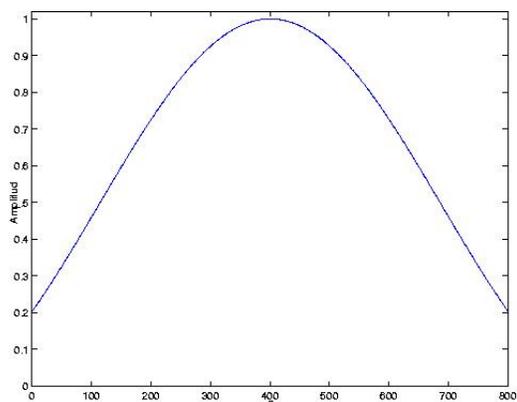


Figura 4.5 Ventana Kaiser-Bessel,  $\alpha = 3$ .

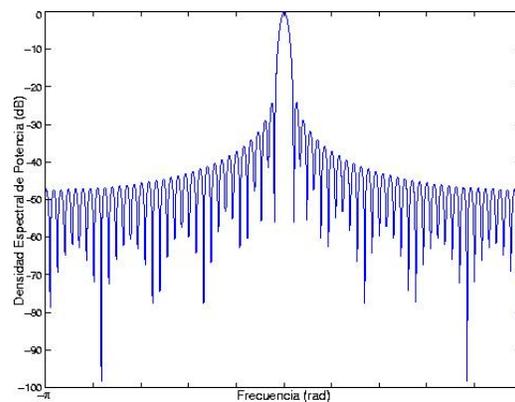


Figura 4.6 Espectro de ventana Kaiser-Bessel.

A diferencia de las otras ventanas, la ventana de Kaiser variando  $\alpha$  se puede ajustar la amplitud de los lóbulos laterales y el ancho del lóbulo principal. Sin embargo, ninguna de las anteriores satisface las restricciones de traslape y suma para poder ser utilizada en la MDCT, por lo que requerirían un proceso de normalización.

Una ventana que cumple con las características de traslape y suma, así como la introducción de mínimo aliasing, es la ventana atómica  $up(x)$ . La cual es mostrada en la figura 4.7 y 4.8 con su espectro de frecuencias, respectivamente.

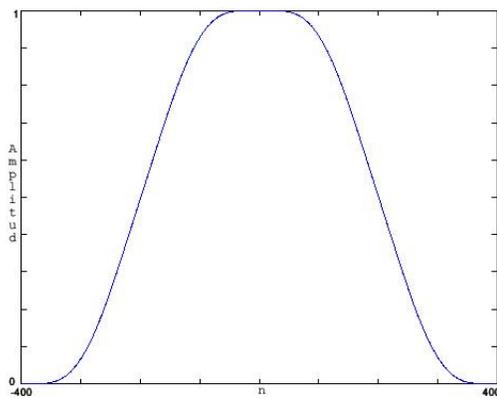


Figura 4.7 Función  $up(x)$

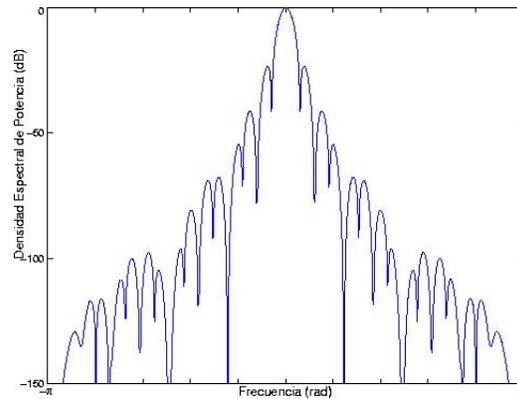


Figura 4.8 Espectro de ventana  $up(x)$ .

En la Figura 4.9 se muestra el comportamiento de la ventana  $up(x)$  para el análisis y su inversa para la síntesis, que satisface la condición de ser inversas entre sí.

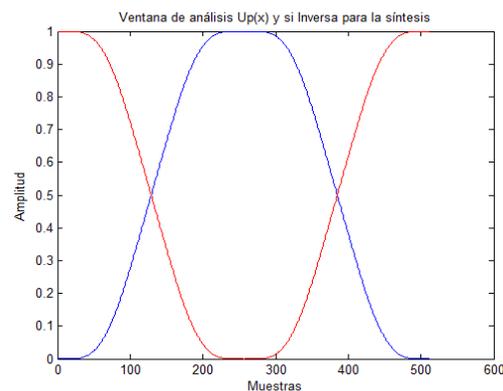


Figura 4.9 Ventana  $up(x)$  y su inversa

Esta ventana, además de cumplir con las condiciones anteriores hace que al ser multiplicada por un bloque de la señal original cumpla con su función de suavizar los bordes y su transición hacia cero de cada uno de ellos.

## 4.2 Fase de Compresión con la MDCT

Esta transformada, es del tipo de Cancelación de Aliasing en el Dominio del Tiempo que utiliza bloques con traslape para poder reconstruir la señal, ya que su transformada inversa no es del estilo de una FFT. Pues para que la MDCT pueda tener una reconstrucción exacta requiere de una técnica de traslape y suma.

### 4.2.1 Análisis y Síntesis utilizando la MDCT

Ahora se expondrá como es el proceso de análisis y síntesis con la MDCT. Comprobando que una señal cualquiera, podrá ser reconstruida exactamente utilizando MDCT y su inversa si se utiliza una ventana válida y las transformadas inversas se suman en la parte en que se traslapan los bloques.

Por ejemplo, para una señal de prueba una señal cosenoidal con una frecuencia de 14KHz con una frecuencia de muestre de 48KHz como la mostrada en la siguiente figura:

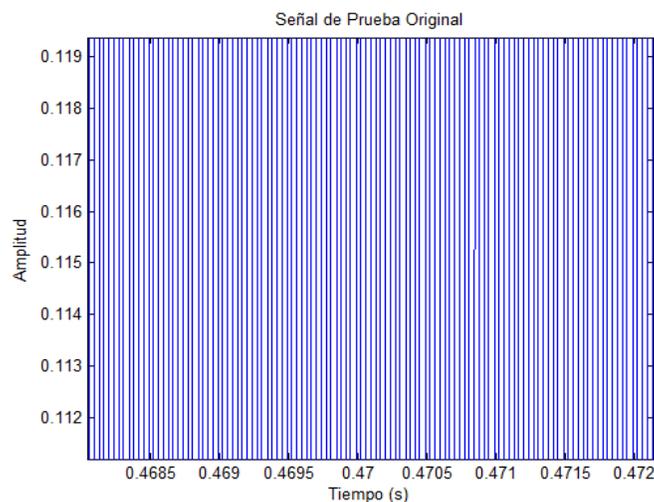


Figura 4.10 Señal de prueba original

En la fase de análisis se obtienen bloques de  $N = 512$  como el mostrado en la figura 4.11, con un traslape del 50% y al multiplicarse por una ventana de análisis válida, el bloque obtenido se muestra en la figura 4.12

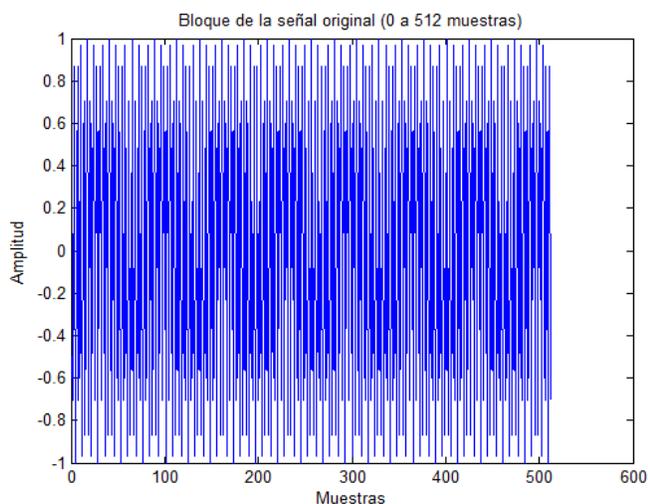


Figura 4.11 Bloque de 512 de la señal original

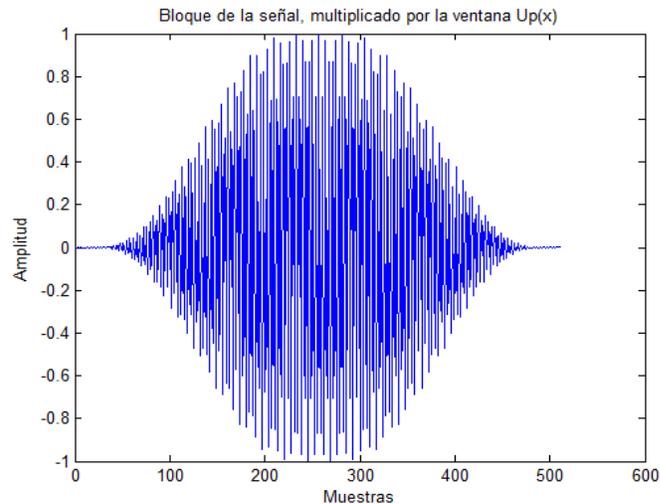


Figura 4.12 Bloque de la señal original multiplicado por la ventana  $up(x)$

Posteriormente se procede a calcular la MDCT, en la figura 4.13 se puede observar la MDCT de un bloque.

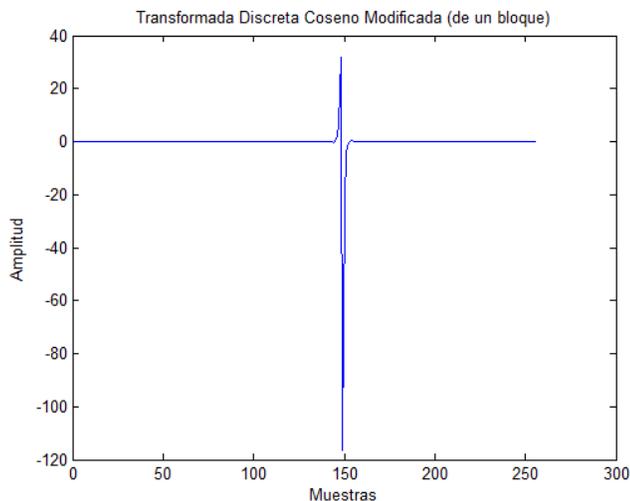


Figura 4.13 MDCT (de un bloque)

En la fase de síntesis se aplica la inversa a cada uno de las transformadas resultantes (figura 4.13). Se observó como el bloque no corresponde a una porción exacta de la señal de prueba, ni al bloque multiplicado por la ventana.

Sin embargo, si sumamos cada uno de los bloques como el de la figura 4.13, se podrá observar como la señal original es progresivamente reconstruida. El traslape y suma de las ventanas se realizará de tal manera que  $n = 0, \dots, N/2 - 1$  e  $i$  indica el número de bloque:

$$w_a^i[n]w_s^i[n] + w_a^{i-1}\left[\frac{N}{2} + n\right]w_s^{i-1}\left[\frac{N}{2} + n\right] = 1$$

En el primer bloque sólo se suma la inversa del bloque 0, como se muestra en la figura 4.14, posteriormente se suma y se traslapa las inversas de los bloques adyacentes; finalmente bloques han sido apropiadamente traslapados y sumados, y la señal obtenida es idéntica a nuestra señal de prueba, como se muestra en la figura 4.15.

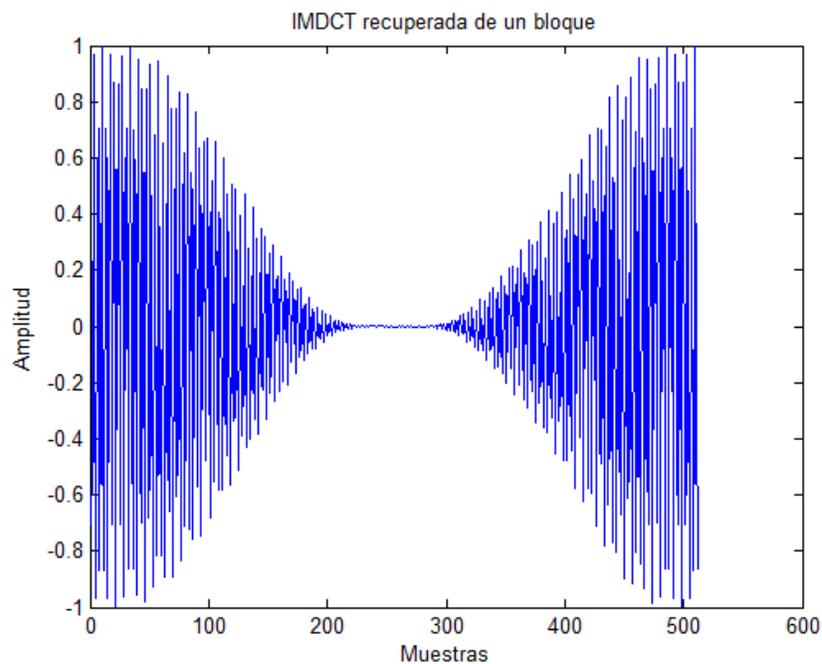
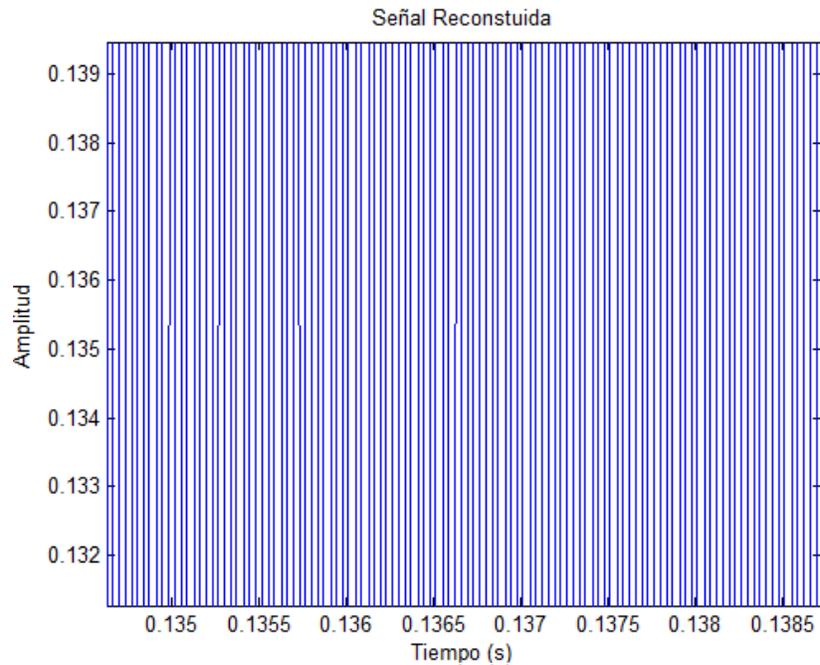
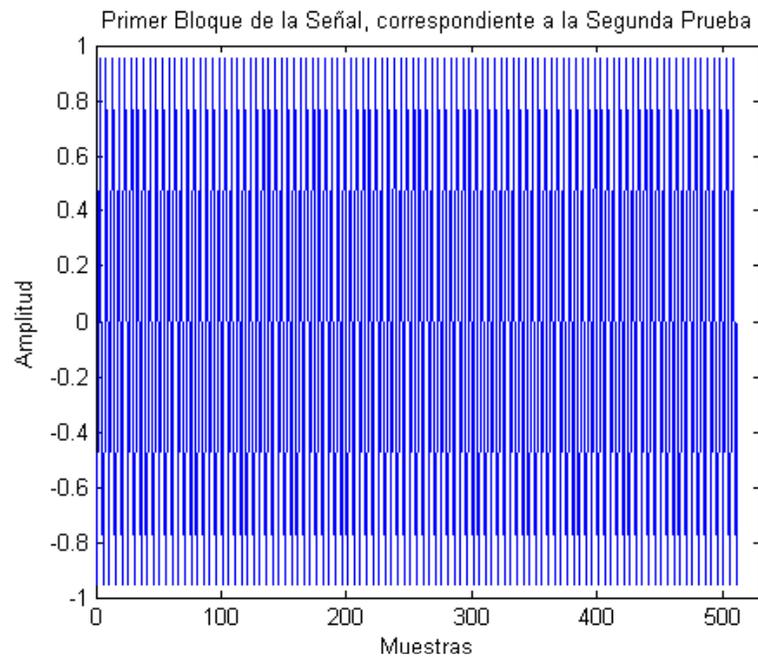


Figura 4.14 IMDCT recuperada

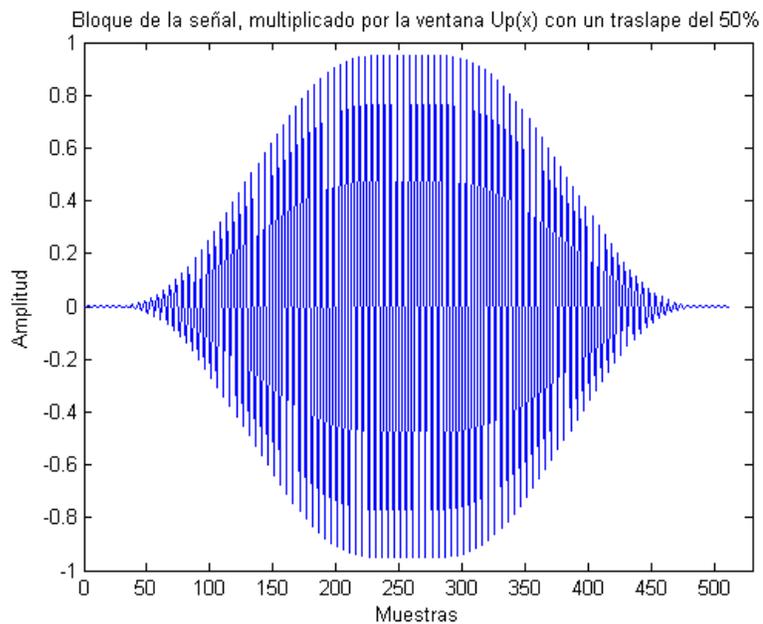


*Figura 4.15 Reconstrucción de la señal con cancelación de aliasing en el dominio del tiempo*

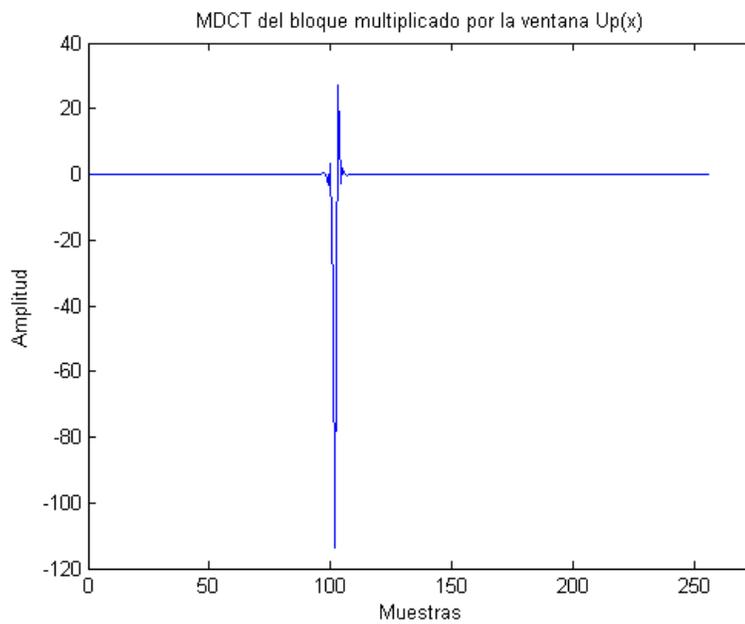
Para mostrar que el proceso es funcional para cualquier señal de entrada, se realizó una segunda prueba con una señal de entrada senoidal de 8 KHz, con una frecuencia de muestreo de 40KHz.



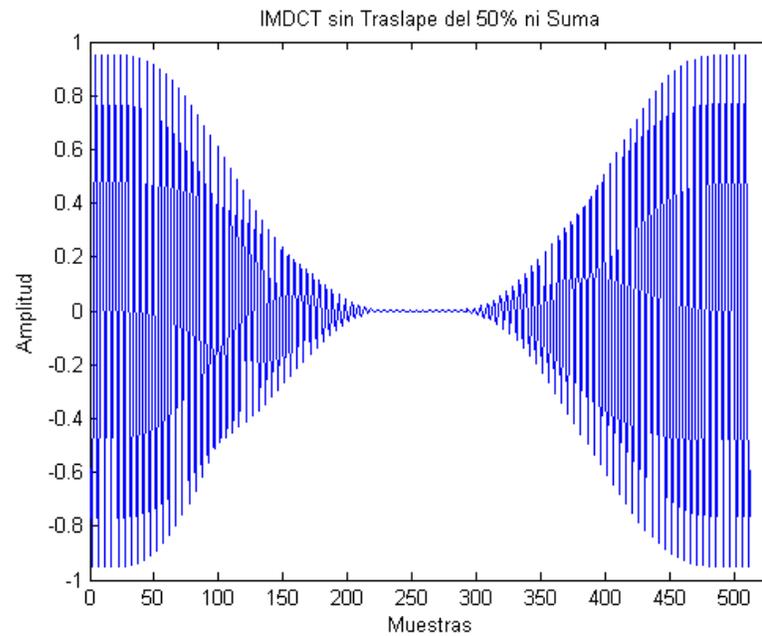
*Figura 4.16 Bloque de 512 muestras*



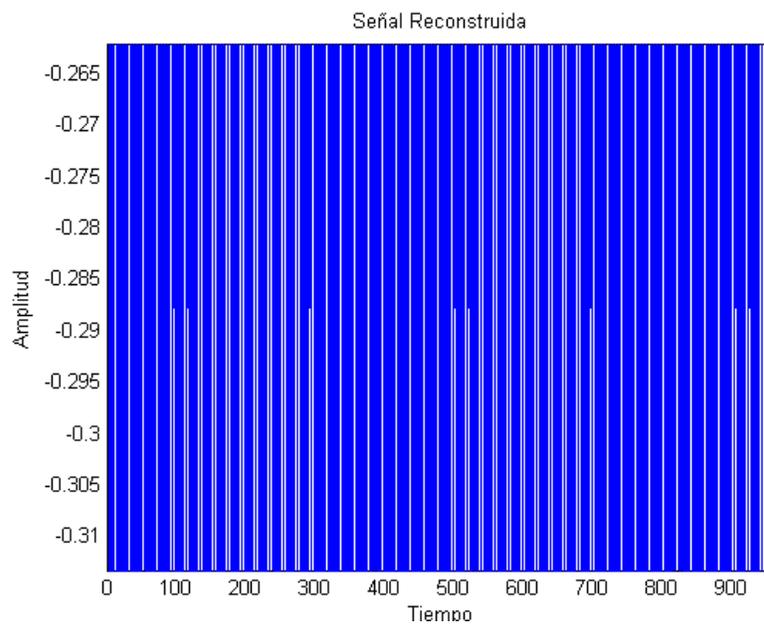
*Figura 4.17 Multiplicación del bloque por la ventana  $up(x)$*



*Figura 4.18 MDCT del bloque multiplicado por la ventana*



*Figura 4.19 IMDCT sin traslape del 50% ni suma*



*Figura 4.20 Reconstrucción de la señal original de prueba 2*

Por último, se realizó una tercera prueba con una señal de 6KHz y una frecuencia de muestreo de 60Khz.

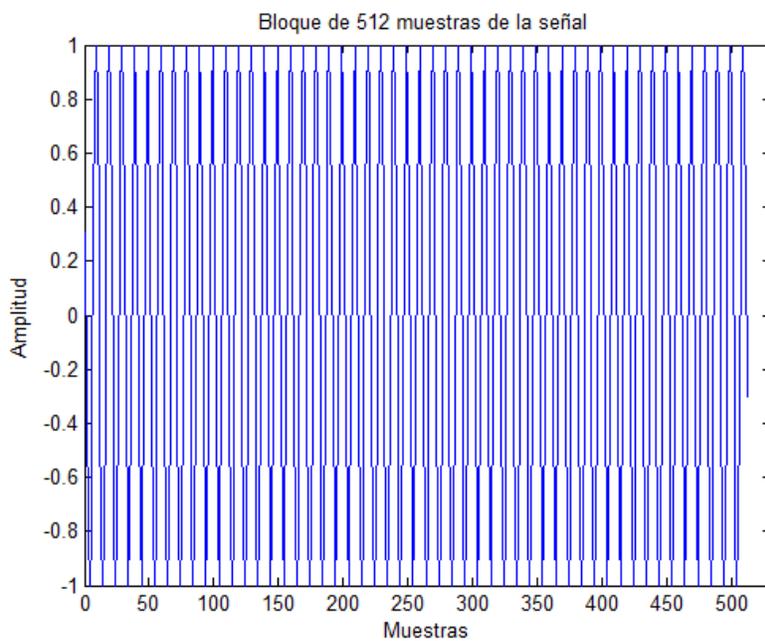


Figura 4.21 Bloque de 512 muestras

Bloque multiplicado por la ventana  $up(x)$

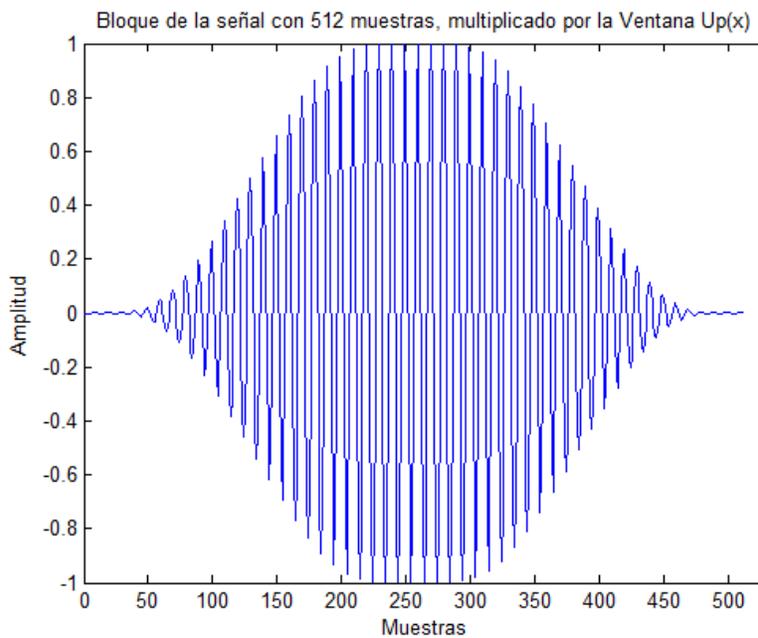
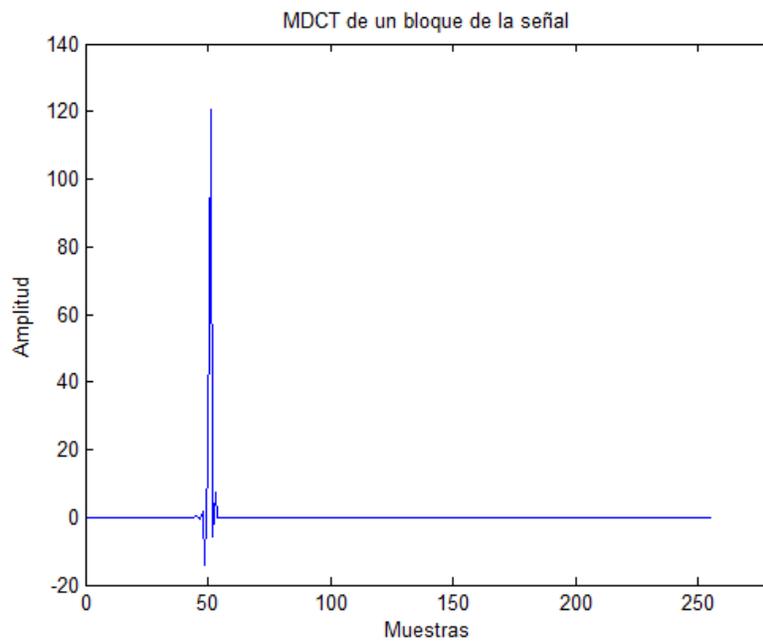
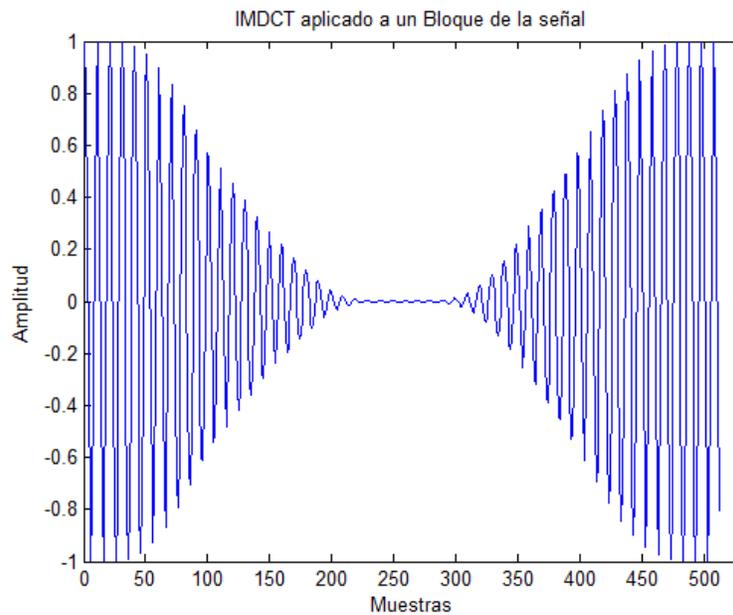


Figura 4.22 Bloque multiplicado por la ventana  $up(x)$

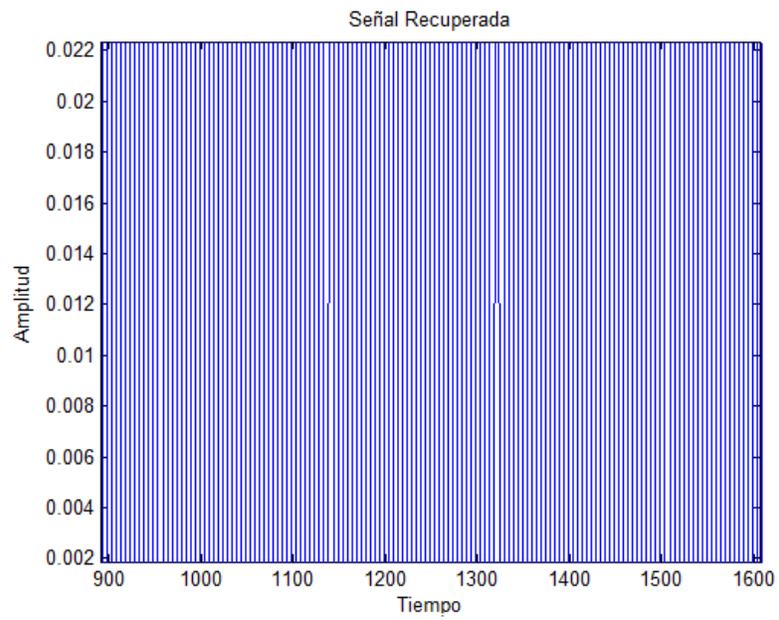
## MDCT del bloque anterior

*Figura 4.23 MDCT de un bloque de 512 muestras*

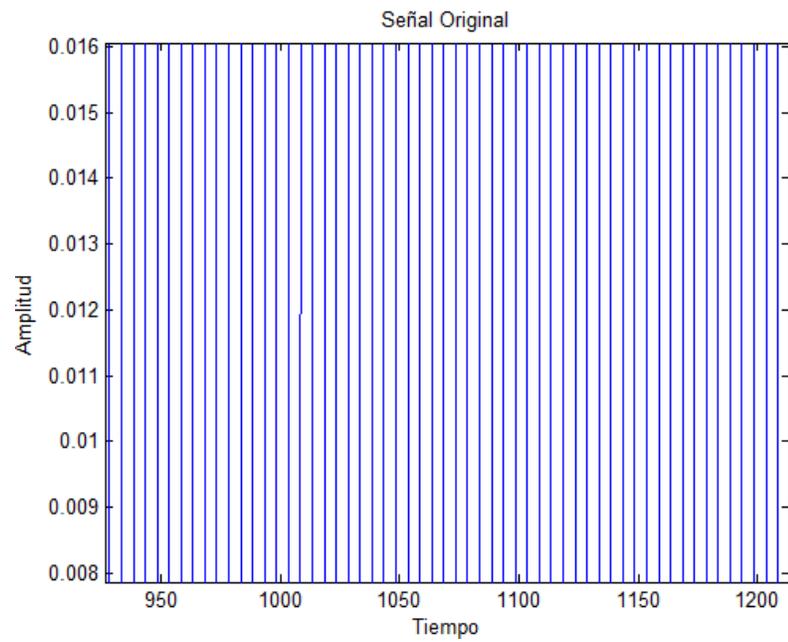
## Bloque de la señal recuperada

*Figura 4.24 IMDCT del bloque de 512 muestras*

## Señal recuperada

*Figura 4.25 Señal recuperada*

## Señal original

*Figura 4.26 Señal original*

La MDCT garantiza matemáticamente que la reconstrucción de la señal será exacta en ausencia de cuantización. Para poder comprimir será necesario utilizar el menor número posible de bits para almacenar los coeficientes de la MDCT, con lo que se introducirá un ruido de cuantización que hará que la señal reconstruida no sea igual a la original. Sin embargo, es posible utilizar cuantizadores con pocos bits si el ruido de cuantización producido es imperceptible. Para poder asignar este ruido de cuantización, es necesario conocer la forma en la que el oído humano percibe el sonido, y esta área del conocimiento es tratada en la psicoacústica.

La información obtenida del modelo psicoacústico es utilizada para escoger la precisión con la que serán utilizados los coeficientes de la MDCT. Una precisión grande ocasionará que la señal reconstruida tenga un error muy pequeño, sin embargo, al utilizar muchos bits la relación de compresión se verá afectada. Utilizar poca precisión hará que la señal reconstruida tenga un error muy grande, pero se tendrá una muy buena compresión.

Al codificar en el dominio de la frecuencia, se sabe que frecuencias tendrán más distorsión. Es aquí donde el modelo psicoacústico juega un papel decisivo, pues éste nos indicará en que porciones del espectro el oído no será capaz de percibir el ruido introducido por la cuantización y, aún más, nos dará información sobre la cantidad máxima de ruido que se puede localizar en dicha frecuencia sin que sea audible.

Una vez que los coeficientes se han cuantizado se procede a empaquetar los bits en algún formato especificado, este formato deberá contener toda la información necesaria para reconstruir cada bloque.

# Conclusiones

El audio digital es una representación de una señal de audio continua, la calidad de ésta no siempre es el deseable, pero con una velocidad de muestreo (al menos del doble de lo perceptible por el oído humano) y una compresión adecuada es posible tener una representación fiel, aunque no exacta, del original. Es por esto que se revisaron aspectos fundamentales de la codificación perceptual. Comprobando que la MDCT es una transformada que utiliza cancelación de aliasing en el dominio del tiempo y es usada en codecs para obtener formatos como MP3 y AAC, concluyendo así que para que la MDCT ofrezca reconstrucción exacta es necesario que las ventanas de análisis y de síntesis cumplan con ciertas restricciones; la ventana derivada de  $U_p(x)$  es una ventana que cumple con estas restricciones.

Al realizar la reconstrucción de la señal de audio a través de la IMDCT se observó que esta no presenta pérdida alguna, por lo que es la ideal para el proceso de decodificación, sin embargo, al ser ésta la misma señal que la de entrada, necesita de etapas tales como la del modelo psicoacústico, filtrado y codificación para obtener un producto terminado, con calidad igual o mejor a la del formato MPEG Layer III.

Por lo tanto, se concluye que se cumplió el objetivo planteado al obtener un algoritmo funcional que realiza la etapa de compresión y descompresión de la señal fundamental en el estándar MPEG.

# Apéndice A

## Professional Audio Development Kit

### A.1 Introducción

Un DSP o Digital Signal Processor (Procesador Digital de Señales) es un microprocesador especializado en el análisis de señales en tiempo real. Se caracteriza por poseer un set de instrucciones que permiten manejar de manera eficiente el procesado de una señal.

Todo el diseño de un DSP se basa en el manejo eficiente de una señal digitalizada, ya que, debido a la latencia típica de cualquier procesador, la operación matemática debe hacerse en un tiempo acotado, para que se pueda trabajar en tiempo real. Cuando se convierte una señal analógica en digital, ésta es muestreada cada cierto intervalo, por lo que cualquier operación matemática debe hacerse entre muestras, es decir, una vez que llegue una muestra se debe realizar la operación y debe estar lista antes de que llegue la otra. Un DSP proporciona una rapidez y baja latencia, que permite el procesado de señales en tiempo real, y aún así siendo una herramienta de bajo consumo.

Entre las aplicaciones que más utilizan un DSP se encuentran:

- ◆ Ámbito industrial (automatización, control, bombas, compresión, robótica, etc.).
- ◆ Automotriz (dirección electrónica, sensado de presión de torque y de inercia, sensor de colisión, etc.).
- ◆ Digital (AC/DC, UPS, DC/AC, displays LCD, etc.).

- ◆ Sensado Avanzado (RFID, medición óptica, lector de código de barras, sensado y análisis médico, medidor de flujo, control de pantallas Touch Screen, efectos de audio y control retroalimentado, etc.)

Las aplicaciones de DSP en el campo del audio profesional son cada día más variadas y complejas, al punto de que ya no podemos concebir el proceso de producción de un disco sin la intervención de alguna forma de DSP.

Por suerte, la tecnología permite hoy que estos dispositivos sean relativamente económicos y fáciles de conseguir. Sin embargo, no es suficiente con disponer de gran potencia de procesamiento y rango dinámico, ya que la respuesta de un sistema depende en gran medida de la correcta elección de los algoritmos y de las arquitecturas de sistema, que se dividen en dos grandes grupos:

- ◆ Arquitectura de Punto Fijo y
- ◆ Arquitectura de Punto Flotante

No debe confundirse la capacidad de procesamiento de un chip de DSP (medible, entre otras cosas, por la cantidad de bits que puede manejar en forma simultánea) con la resolución inherente al sistema de audio que utilizamos. Por ejemplo, mientras que el estándar de resolución en que trabajamos nuestras mezclas es de 24 bits, los circuitos integrados trabajan internamente con longitudes de palabra de 48 o 64 bits, o incluso bastante más en algunos casos. Lo que diferencia internamente a estos chips es precisamente su arquitectura, es decir la forma que utilizan para representar digitalmente la señal que estamos procesando.

La arquitectura de Punto Fijo fue introducida a comienzos de la década del '80, y está basada en una representación que contiene una cantidad fija de dígitos después del punto decimal. Al no requerir de Unidad de Punto Flotante (FPU), la mayoría de los chips DSP de

bajo costo utilizan esta arquitectura, aunque en determinados casos esta alternativa ofrece también mejor performance o mayor exactitud.

Los bits a la izquierda del punto decimal se denominan bits de magnitud y representan valores enteros, en cambio los bits a la derecha del punto decimal representan valores fraccionales (potencias inversas de 2). Es decir que el primer bit fraccional es  $\frac{1}{2}$ , el segundo es  $\frac{1}{4}$ , el tercero es  $\frac{1}{8}$ , etc.

La arquitectura de Punto Flotante es moderna y resulta suficientemente exacta y rápida para la mayoría de las aplicaciones. Es muy frecuentemente utilizada para lograr una buena aproximación del número que se desea representar, pero a menudo requiere de un “redondeo”, debido a su limitada precisión. Su representación involucra un número entero (la mantissa) multiplicado por una base elevado a un exponente, de tal forma que cualquier número de punto flotante  $a$  puede ser representado como se indica en la siguiente expresión:

$$a = mb^e$$

Es posible especificar cuántos dígitos de precisión se requieren, asignando un valor al parámetro  $p$ . La gran ventaja de esta arquitectura reside en que permite la representación de un rango de magnitudes mucho más amplio que el de la arquitectura de punto fijo. De acuerdo a la cantidad de bits utilizados para almacenar un determinado número de punto flotante, decimos que éste es de precisión simple (32 bits) o precisión doble (64 bits). En el caso de precisión simple, típicamente se le asignan a la mantissa los 23 bits menos significativos (bit 0 a bit 22), luego el exponente ocupa los siguientes 8 bits (bit 23 a bit 30) y el bit 31 está destinado a indicar el signo (0 = positivo, 1= negativo). Existen implementaciones de software que emplean hasta 128 bits de punto flotante.

## A.2 Code Composer Studio V3.3

El Code Composer Studio™ (CCS) es un IDE (Integrated Development Environment), es decir, un ambiente de desarrollo integrado para programar los DSP de Texas Instruments. CCS entrega todas las herramientas necesarias para soporte de software en tiempo real para el DSP TMS320C6727.

El CCS permite un manejo rápido y sencillo para moverse en todas las fases del proceso de diseño de una aplicación, incluyendo código y compilación, análisis y afinamiento. Entre las principales propiedades del CCS se encuentran las siguientes:

- ◆ Algoritmo TMS320 DSP estándar.
- ◆ Familiar y fácil de usar IDE con editor CodeWright integrado.
- ◆ Kernel DSP/BIOS™ escalable en tiempo real y herramientas de configuración.
- ◆ Herramientas para generación de código en lenguajes C/C++/Assembly.
- ◆ Compilador de código fuente con análisis en tiempo real.
- ◆ Variedad de simuladores preconfigurados y tarjetas para todas las plataformas.
- ◆ Códigos de ejemplo y librerías.
- ◆ Soporte para TMS320C2000™, TMS320C5000™ (incluyendo los procesadores OMAP™), y TMS320C6000™ (incluyendo procesadores the DaVinci™).

Entre las características que posee el ambiente de desarrollo de CCS se encuentran las herramientas de trabajo que mejoran el tiempo de trabajo para desarrollo de software y optimizan el funcionamiento para todas las aplicaciones DSP en tiempo real.

Las herramientas más importantes del Code Composer Studio incluyen:

- ◆ Herramientas para generación de código DSP TMS320: compilador C/C++, ensamblador
- ◆ Viene con Visual Project Manager con interfaz de control de código y soporte para multiproyectos que permite manejar sobre 100 archivos
- ◆ Configuración Drag and Drop CCStudio que soporta:
  - Drivers XDS560™ de emulación de alta velocidad
  - Drivers XDS510™ de emulación
  - Simuladores para todos los dispositivos, solo para CPU y CPU más memoria para funcionamiento óptimo
- ◆ Opciones de definición configurable para tarjetas
- ◆ Editor CodeWright totalmente integrado
- ◆ Administrador de componentes, el cual soporta múltiples versiones de DSP/BIOS y de herramientas de generación de código dentro del IDE
- ◆ El compilador posee las siguientes propiedades:
  - Debug Rewind: Retrocede en la simulación
  - Breakpoint simples
  - Breakpoints avanzados y de Hardware
  - Puntos de prueba para inyección/extracción de datos
  - Navegador para Símbolos y Watch Windows avanzado
  - Administrador unificado de Breakpoints para configurar y administrar breakpoints por software y hardware desde una sola y simple interfaz
- ◆ Conexión robusta entre el cliente y la tarjeta.
- ◆ Transmisor de datos para intercambio en tiempo real entre la tarjeta y el cliente.
- ◆ Un sin número de guías y códigos de ejemplo para hardware y software. Además de tutoriales y manuales.

## A.2.1 Instalación y Configuración del CCS

A continuación, se explicará la configuración básica necesaria para el CCS. Para la versión actual 3.3 será CCStudio v3.3 y Setup CCStudio v3.3. El primero corresponde al CCS en sí, mientras que el otro se debe correr antes de empezar cualquier proyecto. Este programa se encarga de la configuración básica del DSP así como también de sus librerías. Para el correcto funcionamiento con la placa, se debe actualizar la versión del CCS con el último Service Pack disponible en la página de Texas Instruments, luego bajar el driver XDS510. Este último se encarga de emular la tarjeta a través del JTAG de la tarjeta con el puerto USB de la PC y éste a la vez con el CCS.

Una vez instalado todo lo anterior se ejecuta Setup CCStudio v3.3 donde aparecerá la ventana que se muestra en la figura A.1:

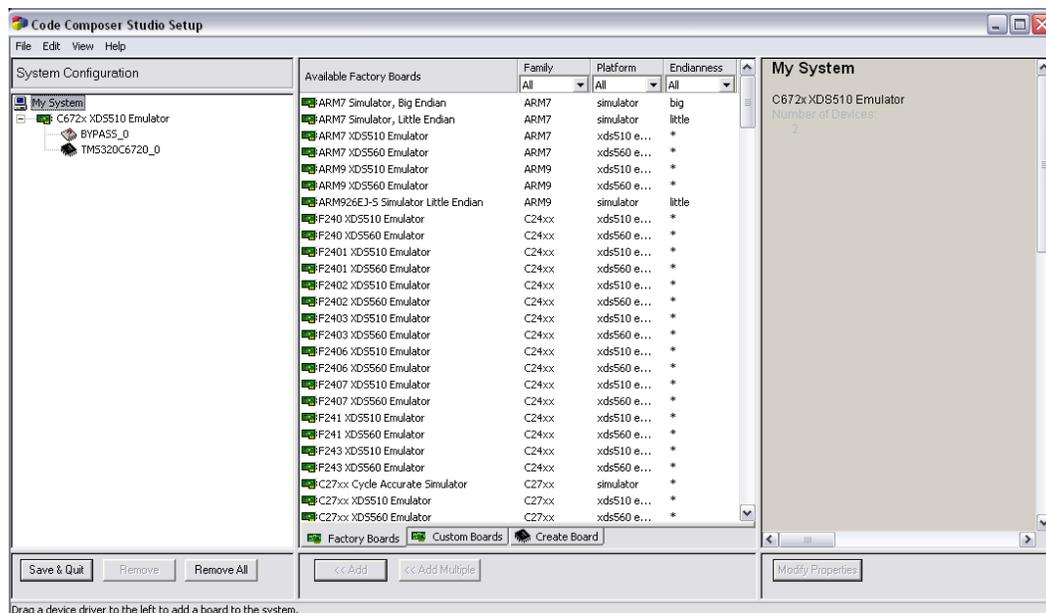


Figura A.1 Setup CCStudio

En seguida se debe seleccionar Create Board y luego elegir la conexión C672x XDS510 emulador y presionar el botón Add. En la siguiente ventana escribir un nombre arbitrariamente presionar el botón Next y luego Finish. Esa ventana se muestra en la figura A.2:

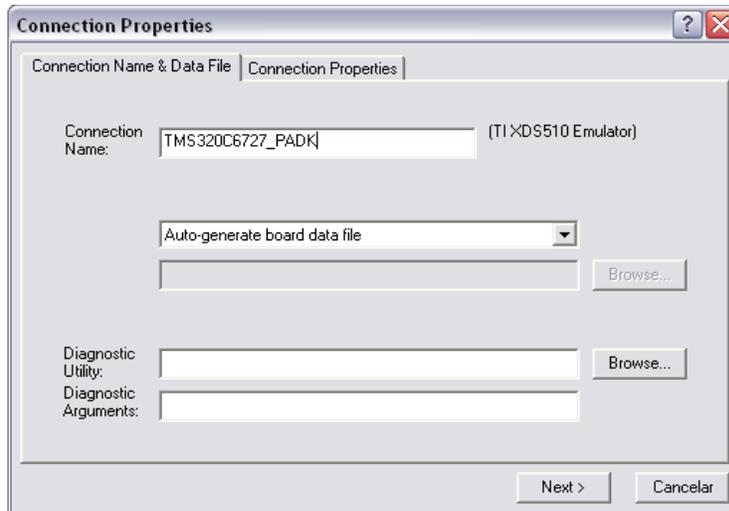


Figura A.2 Create Board

Luego se marca la tarjeta creada y se escoge la tarjeta DSP modelo TMS320C6727 y se agrega a la conexión, tal como se muestra en la figura A.3:

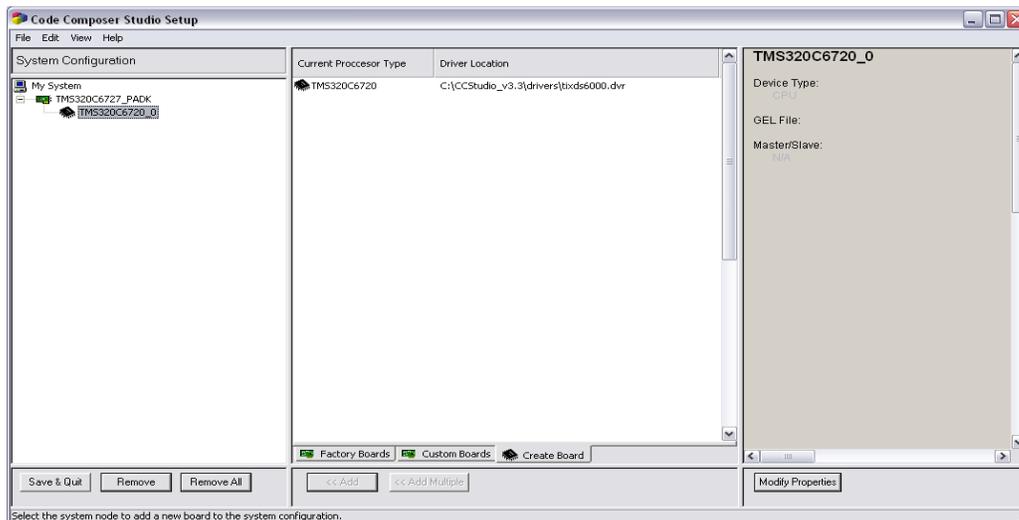
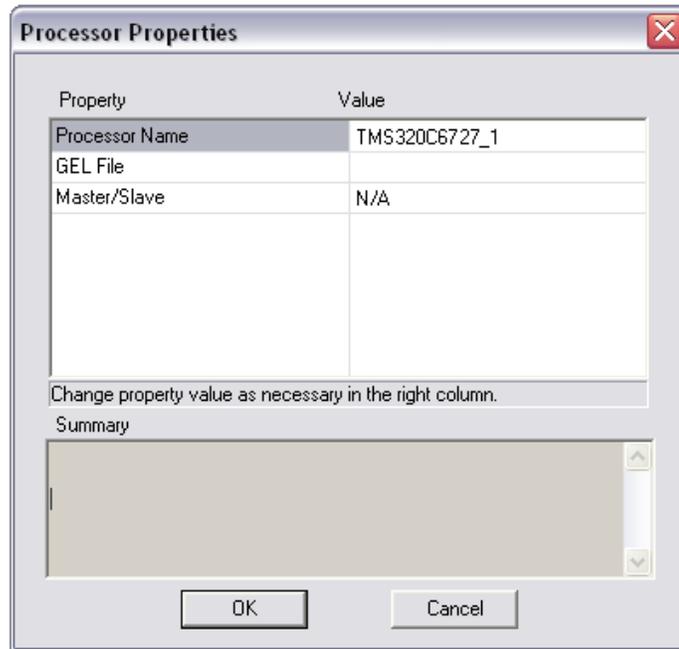


Figura A.3 Tarjeta DSP modelo TMS320C6727

Para agregarla simplemente se presiona el botón Add y aparecerá una ventana tal y como la que se muestra en la figura A.4:



*Figura A.4 Propiedades del Procesador*

Posteriormente se presiona OK y se guarda desde el panel de configuraciones en la barra de herramientas o presionando el botón Save & Quit. Luego de ello, el CCS se iniciará automáticamente [39].

### **A.3 Professional Audio Development Kit**

El Professional Audio Development Kit (PADK) fue desarrollado por Texas Instruments™ en colaboración con Lyrtech™. El PADK es un hardware/software de solución basada en la nueva generación de DSP's de punto flotante, pertenecientes a la gama TMS320C672x. El modelo específico del DSP es el C6727, el cual está diseñado para aplicaciones de audio, lo que lo hace ideal para lograr los objetivos planteados.

Es importante mencionar que el PADK es programable con Code Composer Studio en su versión 3.3.

Las más importantes aplicaciones que el PADK ofrece, tienen lugar en los siguientes campos:

- ◆ Audio Profesional: Emisión de audio, decodificadores y aplicaciones de investigación.
- ◆ Música y Sonido: Modificación y generación
- ◆ Generación de Sonido Embebido: Juegos, Karaoke, música de fondo y síntesis de sonido.

#### **4.4 Características**

Las características principales del PADK en cuanto a hardware son las siguientes:

- ◆ Modelo del DSP TMS320C6727
- ◆ Alta calidad en los Convertidores ADC y DAC.
- ◆ Resolución de 24 bits
- ◆ Velocidad de muestreo de 192 KHz
- ◆ 8 canales, los cuales soportan el desarrollo de aplicaciones de audio envolvente 7.1
- ◆ Conector de entrada y salida para coaxial y audio digital óptico
- ◆ Ranura de expansión, lo cual le permite actualizarse en cuanto a hardware
- ◆ Puerto USB 2.0

El PADK cuenta con 2 convertidores Analógico-Digital y dos Digital-Analógico, cada uno con cuatro canales, cada convertidor muestrea a 192 KHz y se pueden utilizar tanto para la codificación multicanal, como para el procesamiento y generación de sonido. La entrada de audio digital incorpora una alta velocidad de muestreo, lo cual le permite funcionar de entrada y salida para diferentes velocidades de muestreo, así mismo le permite bloquear la entrada de reloj de las entradas de audio.

El puerto de USB le permite transmitir audio en tiempo real, descargar grandes bancos de sonido y manejar cualquier otra aplicación que requiera un ancho de banda considerable.

Los formatos de audio compatibles con el PADK son los siguientes:

- ◆ .wav hasta 44.1 KHz (monofónico o estereofónico)
- ◆ .mp3 hasta 44.1 KHz (monofónico o estereofónico)
- ◆ .mid
- ◆ .sf2

#### **4.5 Requerimientos técnicos**

Los principales requerimientos técnicos que debe poseer un ordenador para soportar la interfaz del PADK son los siguientes:

- ◆ Windows XP con Service Pack 2
- ◆ Tener instalado el Code Composer Studio v3.3 actualizado completamente
- ◆ Actualizaciones para las librerías necesarias de la serie C672x
- ◆ Un emulador del tipo XDS510 o XDS560

#### **4.6 Especificaciones Técnicas del PADK**

Dimensiones

245,0 mm x 50,0 mm x 192,0 mm (ancho x profundidad x altura)

Peso

761,82 g (26,87 oz.)

Rango de temperatura operativa

De 10 ° C a 40 ° C (50 ° F a 104 ° F)

Temperatura de almacenamiento

-10 ° C a 50 ° C (14 ° F a 122 ° F)

Consumo de energía

Mínimo: 1,0 A (5 W)

Típico: 2,4 A (12 W)

#### 4.7 Descripción General del DSP TMS320C6727

La figura A.5 muestra la relación de las entradas y las salidas del PADK con el DSP TMS320C6727:

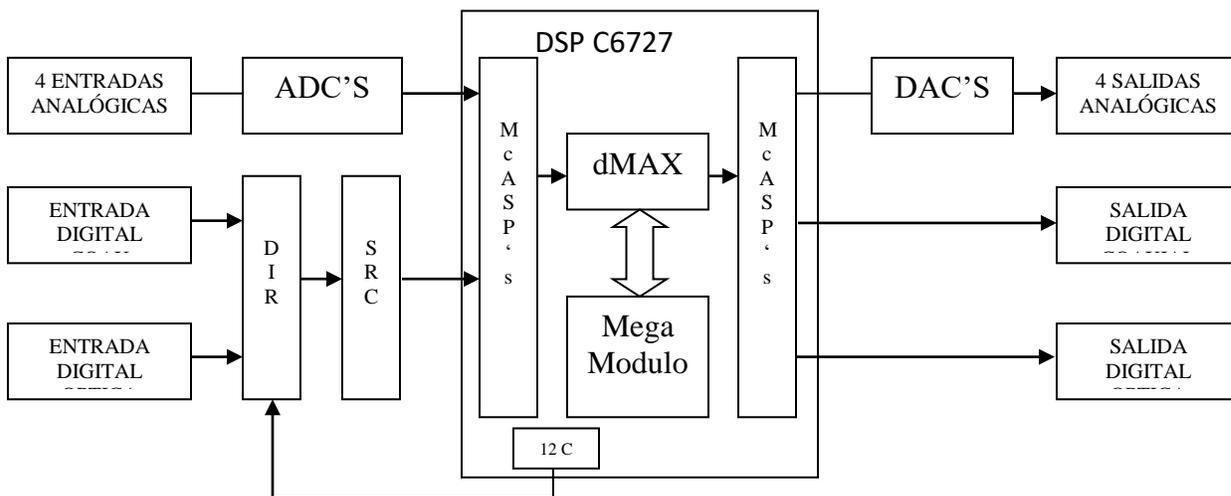
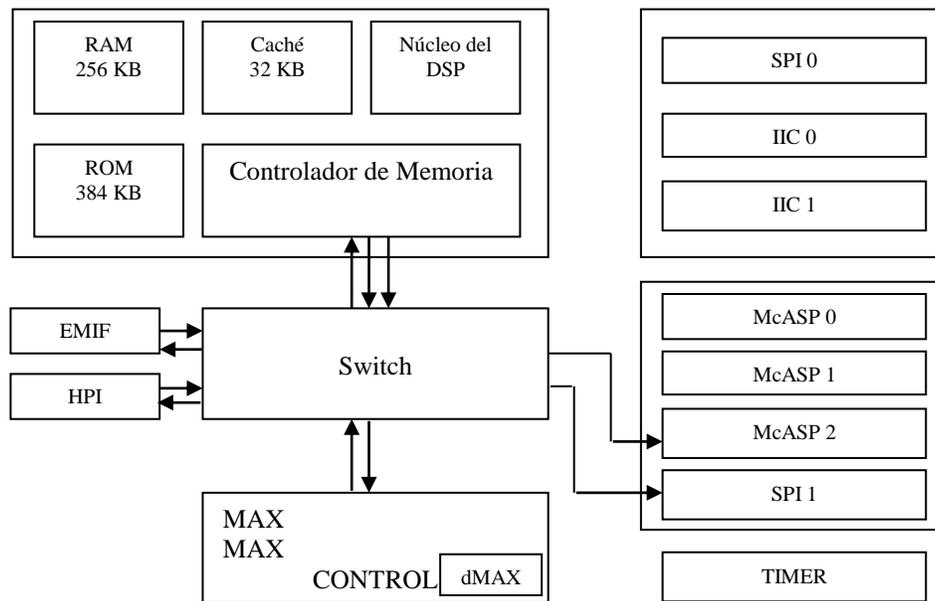


Figura A.5 Entradas y las Salidas del PADK TMS320C6727

Debido a la arquitectura Harvard, el espacio de memoria de programa y de datos se encuentran separados utilizando un bus independiente para cada memoria. Esto permite que mientras se accede a un espacio de memoria de programa, sea posible acceder a la

memoria de datos al mismo tiempo, es decir, es posible acceder a ellos casi al mismo tiempo.

El núcleo del DSP tiene una velocidad de 300MHz y 64 registros de punto flotante, con una memoria ROM de 384KB y 256KB de RAM, así como 32KB en caché y EMIF para expansión de memoria. La figura A.6 muestra un esquema de la disposición de memorias en el DSP:



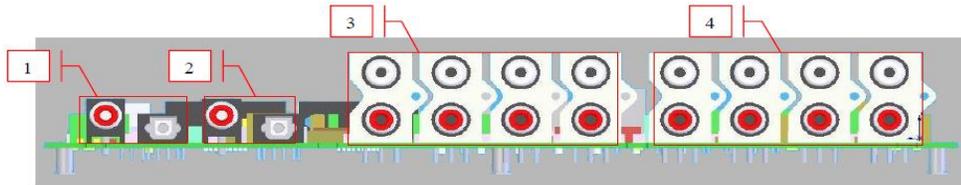
*Figura A.6 Memorias en el DSP*

La unidad dMAX ejecuta las interrupciones y el procesamiento de los registros, la codificación, la generación de la dirección del hardware para eventos de alta y baja prioridad HiMAX y LoMAX respectivamente.

En el bloque de McASP, se encuentran tramas de tipo TDM (de 2 hasta 32) y 384 time slots. Así mismo este bloque contiene un formateador de datos por manipulación de bit y aproximadamente 16 pines de datos seriales asignables, ancho variable de la trama y recuperación y chequeo de errores.

## 4.8 Descripción General del PADK

La figura A.7 presenta la parte frontal del PADK



*Figura A.7 Parte frontal de PADK*

1. Conectores de entrada estéreo digital (S/PDIF): Se utiliza para señales de entrada digital estéreo a la PADK a través de su conector coaxial u óptico.
2. Conectores de salida estéreo digital (S/PDIF): Usada para la salida de estéreo digital de señales procedentes del PADK a través del conector coaxial u óptico.
3. Conectores de entrada analógica: El PADK está tiene con ocho conectores RCA de entrada analógica. Las señales de entrada pasan por los convertidores ADC.
4. Conectores de salida analógica: Este dispositivo cuenta con ocho salidas analógica RCA. Las señales de salida de estos conectores se convierten a través de los convertidores digital-analógico. La figura A.8 muestra la parte frontal del PADK.



*Figura A.8 Parte frontal*

El interior del PADK es esquematizado con la figura A.9 con el fin de dar una descripción general de las partes que integran dicha placa.



*Figura A.9 Interior del PADK*

1. FPGA grupo de acción conjunta de prueba JTAG por sus siglas en inglés: Este dispositivo de 6-pines se utiliza como una interfaz para el desarrollo FPGA.
2. Conector DSK: Se utiliza para conectar el usuario suministra módulos de expansión basada en el estándar de DSK de Texas Instruments.
3. Memoria flash: El PADK tiene una capacidad de memoria flash de 8 MB.
4. Field Programmable Gate Array (FPGA): El PADK utiliza un Xilinx Spartan-II-E como su prototipo FPGA a aplicaciones de audio profesional.
5. Procesador de señal digital (DSP): El PADK está equipado con un Texas Instruments TMS320C6727 de alto rendimiento de 32 bits de punto flotante de

6. DSP. El procesador utiliza un 300-MHz o 350 MHz, reloj interno, dando un rendimiento que pueden alcanzar hasta el 2800 o 2100 MFLOPS MIPS.
7. Conector DSK: Se utiliza para conectar el usuario suministra módulos de expansión basada en el estándar de DSK de Texas Instruments.
8. Microcontrolador Bus serie universal (USB): Se utiliza un microcontrolador de Cypress Semiconductor EZ-USB FX2 (CY7C68013) para que el PADK sea compatible con USB de alta velocidad y conexiones USB.
9. Musical Instrument Digital Interface (MIDI) UART: El TL16C550 maneja la señal asíncrona de instrumentos musicales y también es capaz de un modo alternativo FIFO, esto significa el alivio del DSP para la excesiva sobrecarga de software.
10. Receptor/Transmisor Asíncrono universal (UART): El TL16C550 maneja la asíncrona de señales provenientes de dispositivos diferentes del MIDI.
11. Convertidor analógico a digital de 8-bits (ADC): El ADS7830 es un solo dispositivo, de baja potencia, de 8-bit ADC equipado con una interfaz serial I2C externa utilizada para el control de entrada, de diferentes equipos externos tales como los pedales y conmutadores.
12. Interfaz DSP Joint Test Action Group (JTAG): Esta parte de 14-pines se utiliza como una interfaz de depuración para el Code Composer Studio™ de Texas Instruments.
13. EEPROM serie I2C: Esta es una memoria eléctricamente programable y borrable de sólo lectura.

14. Convertidor Asíncrono de frecuencia de muestreo de tipo estéreo (SRC): El SRC4192 combina una gran aportación a la tasa de muestreo de salida (de 16:1 a 1:16) con un excelente rango dinámico (144 dB). Se puede detectar automáticamente la entrada a la tasa de muestreo de salida y es capaz de división de tiempo en modo multiplexado.
15. Receptor de Interfaz digital (DIR): Cirrus Logic CS8416 capaz de realizar el muestreo digital de señales de audio a tasas de hasta 192 kHz, lo que es ideal para la próxima generación de formatos de audio tales como DVD-Audio. También es capaz de conexiones coaxiales y ópticas.
16. SPI EEPROM serie: Eléctricamente programable y borrrable memoria de sólo lectura.
17. Display de siete segmentos: La pantalla de siete segmentos puede ser programada para mostrar códigos de error, mensajes, etc.
18. DIP: El DIP se puede utilizar como fines generales de selección de los interruptores. Una función que suministra el PADK le permite determinar el estado de los interruptores DIP.
19. Convertidor de audio digital a analógico (CAD): El PADK está equipado con dos, de cuatro canales cada uno, 24-bit Burr-Brown PCM4104 capaz de trabajar con velocidades de muestreo de hasta 192 kHz.
20. Convertidor de audio analógico a digital (ADC): El PADK está equipado con dos, de cuatro canales, 24-bit Burr-Brown PCM4204 capaz de manejar velocidades de muestreo de hasta 192 kHz.

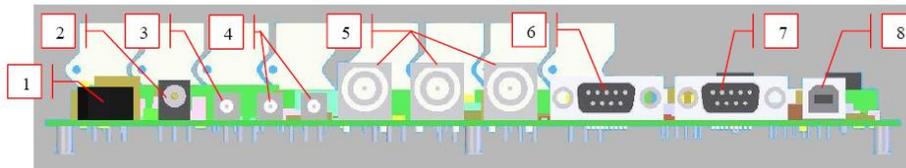
21. Conector DSK: Se utiliza para conectar el usuario suministra módulos de expansión basada en el estándar de DSK de Texas Instruments.
22. Memoria Síncrona de Acceso Aleatorio (SDRAM): El PADK está equipado con un 128-MB SDRAM.

La figura A.10 muestra la parte física de la placa en la cual podemos encontrar los elementos descritos con anterioridad:



*Figura A.10 Placa física del PADK*

La vista posterior del Kit PADK es ejemplificado con la figura A.11 para ubicar y describir sus componentes:

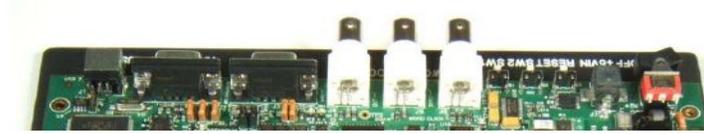


*Figura A.11 Componentes de entradas salidas del PADK*

1. Interruptor de encendido: Se usa para encender y apagar el PADK.

2. Conector de alimentación: Se utiliza para conectar el PADK a los +5 V DC de alimentación.
3. Botón de reinicio: Este botón se utiliza para reiniciar el PADK. En otras palabras, al pulsar este botón reinicializa todos los registros de DSP y FPGA, volviendo a sus valores predeterminados. Sin embargo, al pulsar este botón no borra la programación de la FPGA.
4. Switch de usuario: Para fines generales estos interruptores pueden ser utilizados para configurar las diversas opciones.
5. Conectores para la palabra reloj: Debido a que la exactitud y precisión el calendario de muestras de audio digital es fundamental para el correcto funcionamiento de los equipos de interconexión de audio digital, el PADK está equipado con conectores de reloj, respectivamente (de izquierda a derecha) en, por, y OUT.
6. Interfaz serie (RS-232): La serie permite a los PADK interfaz para comunicarse con el puerto COM de su ordenador.
7. Conector analógico MIDI: El PADK está equipado con un conector MIDI TL16C550, controlado a través de una UART. Está compuesto de entrada y salida MIDI de los puertos, así como con un puerto analógico.
8. Puerto USB: El PADK está cuenta con una alta velocidad de puerto USB (USB 2.0), controlada a través de un Cypress Semiconductor EZ-USB FX2 microcontrolador (CY7C68013).

En la figura A.12 se puede apreciar la distribución física de los componentes descritos con anterioridad.



*Figura A.12 Puertos de Entrada Salida*

## **A.9 Desarrollo de un Bucle de Audio Digital con el PADK**

El archivo DigitalLoopback.pjt demuestra cómo realizar un bucle de Audio digital con el PADK. La tasa de muestreo se fija en 48 kHz. La frecuencia de muestreo se configura con el módulo CLKGEN de la biblioteca del PADK.

Los datos recibidos en el canal de entrada RCA se envían a los canales de salida RCA. El DIR es inicializado con las funciones de la librería del PADK. El dispositivo está conectado a través del serial 0 en la McASP 1. Ambas señales de entrada, tanto la proveniente de la entrada óptica como la de la entrada S/PDIF (entrada digital estéreo), están conectados a la DIR y se puede seleccionar la fuente de la señal digital con una función de la biblioteca del PADK. En esta demostración, la entrada es la entrada S/PDIF. El DSP McASP 2 se utiliza para llevar a cabo la transmisión de la señal digital. Este módulo de la DSP C6727, es usado en modo DIT para transmitir las señales digitalizadas S/PDIF a través de sus datos indicadores 0 y 1. El indicador 0 se utiliza para transmitir a la salida RCA, mientras que el indicador 1 está conectado a la salida óptica.

El McASP y el dMAX se inicializan con la biblioteca CSL de Texas Instruments. El dMAX se utiliza para ordenar los datos de entrada de la DIR en un buffer diferente para cada canal estéreo (izquierdo y derecho buffers). La misma organización se utiliza para los datos de salida de la DIT, como se muestra a continuación:

DIR izquierda muestra 0  
DIR izquierda muestra 1  
...  
DIR izquierda muestra n-1  
DIR izquierda muestra n  
...  
DIR derecho muestra 0  
DIR derecho muestra 1  
...  
DIR derecho muestra n-1  
DIR derecho muestra n

### **A.10 Desarrollo de un Bucle de Audio Analógico con el PADK**

Realización de un bucle de audio analógico se puede realizar al cargar el archivo AnalogLoopback.pjt, el cual demuestra cómo realizar bucle de audio analógico con el PADK. La tasa de muestreo se fija en 96 kHz. La frecuencia de muestreo se configura CLKGEN con el módulo de la biblioteca del PADK.

Cada canal ADC es enviado a su canal correspondiente CAD:

9. ADC de 1 a CAD 1
10. 2 a ADC DAC 2
11. ADC de 3 a CAD 3
12. ADC de 4 a CAD 4

El ADC y el CAD son inicializadas con las librerías del PADK. Los dispositivos están conectados a la DSP a través de los serializadores de la McASP 0, tal como se muestra a continuación:

- ◆ ADC de 1 a McASP 0 serializador 0 (AXR0\_0)
- ◆ ADC de 2 a McASP 0 serializador 1 (AXR0\_1)
- ◆ ADC de 3 a McASP 0 serializador 2 (AXR0\_2)
- ◆ ADC de 4 a McASP 0 serializador 3 (AXR0\_3)
- ◆ DAC de 1 a McASP 0 serializador 4 (AXR0\_4)
- ◆ 2 a McASP CAD 0 serializador 5 (AXR0\_5)
- ◆ CAD 3 a McASP 0 serializador 6 (AXR0\_6)
- ◆ DAC de 4 a McASP 0 serializador 7 (AXR0\_7)

El McASP y la dMAX se inicializan con la biblioteca CSL de Texas Instruments. El dMAX se utiliza para ordenar los datos de entrada de la ADC en buffers separados para cada canal estéreo (izquierdo y derecho). La misma organización es utilizada para los datos DAC de salida.

ADC 0 izquierda muestra 0

ADC 0 izquierda muestra 1

...

ADC 0 izquierda muestra n-1

ADC 0 izquierda muestra n

ADC 1 izquierda muestra 0

ADC 1 izquierda muestra 1

...

ADC 1 izquierda muestra n-1

ADC 1 izquierda muestra n

ADC 2 izquierda muestra 0

ADC 2 izquierda muestra 1

...

ADC izquierda muestra 2 n-1

ADC 2 izquierdo muestra n

3 ADC izquierda muestra 0

3 ADC izquierda muestra 1

...

3 ADC izquierda muestra n-1

3 ADC izquierda muestra n

### **A.11 Demostración de MP3 Reverberado**

Como ya se ha mencionado con anterioridad, el sonido se produce por la colisión de varios cuerpos, generando una onda de presión en todas direcciones a través de la cual se libera energía. Esta onda de presión se recibe en nuestros tímpanos y es interpretada por el cerebro. Cuando el sonido llega a un micrófono los cambios de presión modifican el diafragma del micrófono traduciéndose en cambios en voltaje.

El cerebro posee además la capacidad de localizar la dirección de la fuente de sonido. El sonido que recibimos es una combinación de la emisión directa de la fuente (Primer orden) y los reflejos (Segundo orden) en el entorno (El entorno y sus propiedades de absorción modifican la percepción). Las ondas reflejadas además llegan con retraso respecto a la emisión directa debido a que recorren una distancia mayor.

El cerebro combina todas las reflexiones generándose un efecto de eco llamado reverberación. El PADK contiene una herramienta llamada MP3/Reverb. La ventana que aparece al cargar esta aplicación es la mostrada en la figura A.13:

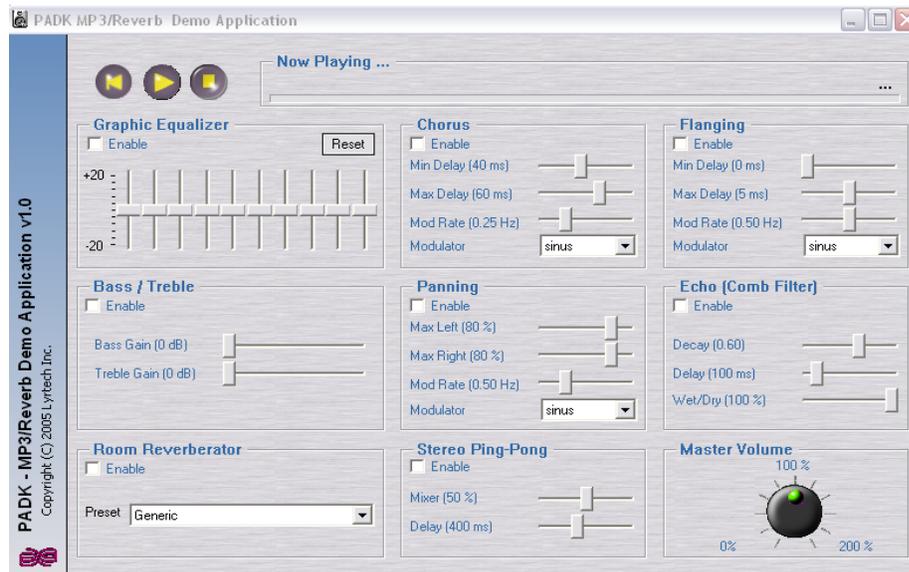


Figura A.13 MP3/Reverb

En esta ventana se pueden observar las diversas características del PADK y su tecnología envolvente de MP3. Está permite reproducir archivos de 44,1 kHz del tipo MP3 y ajustar las preferencias de audio con un ecualizador gráfico, así como la mezcla de efectos especiales, como coro, medios, graves y agudos, paneo, eco, reverberación, estéreo, etc.

Es importante mencionar que para que esta aplicación sea funcional, el cargador de USB debe de encontrarse en la memoria flash del PADK [40].

# Apéndice B

## Código Fuente

El código fuente en MATLAB, fue compilado con Matlab 7.0 en una arquitectura XP

### B.1. Principal.m

```

clc
clear all
close all

t=0:1/48000:1;
f=14000;
sing=2*pi*f*t;
audio=cos(sing);

N=input('Ahora Ingrese el tamaño de la ventana ');

Upx % crea la ventana Up(x) de tamaño N; variable ventana
%ventana_derivada % Modifica ventana Up(x); y regresa en modifwindow;

tamano=length(audio); % Longitud del vector de audio
traslape=N/2;

apunta_inicio=1; %apuntador al inicio del archivo de audio
apunta_fin=N; % apuntador la final del archivo de audio
apunta_out=0;
c=1;
while apunta_fin < tamano
    t=1;
    for i=apunta_inicio:(apunta_fin) %Este for obtiene la ventana de la
señal de audio
        signal(t)=audio(i);
        t=t+1;
    end

    Cosenomodif; % Realiza la Transformada Discreta Coseno regresa en
'transform_cos'
    Invercosmodif;
    apunta_out=apunta_out+traslape;
    apunta_inicio=apunta_inicio+traslape;
    apunta_fin=apunta_inicio+(N-1);
    c=c+1;
end

Fourier=abs(fft(signal));

```

```
figure(7)
plot (Fourier)
```

## B.2. UP(x).m

```
%Crea ventana Atomica

load funciones_2.mat;

%Ventana Up(x)
num=400000;
incre=300/(num-1);
dtt=-150:incre:150;
incre_x=2/N;
dx=-1:incre_x:1;
for jj=1:N;
    var1=dx(jj)*dtt;
    F2=cos(var1);
    F3=F2.*F1;
    %Se realiza la integral de cada producto
    simp=0;
    for i=2:num-1
        simp=simp+4*F3(i);
        i=i+1;
        simp=simp+2*F3(i);
    end
    simp=simp+F3(1)+F3(num);
    hh=300/(3*2*num);
    simp=simp*hh;
    %Almacenamiento del vaor de la integral
    ventana(jj)=simp/(2*pi);
end
%Normalizacion de la ventana maximo = 1...
yy=min(ventana);
ventana=ventana-yy;

yy=max(ventana);
ventana=ventana./yy;

modifwindow=ventana;
```

### B.3. Ventana\_derivada.m

```

% subrutina que modifica ventana para MDCT

[filas,colum]=size(ventana);
tam=colum;
for i=1:tam
    aux=0;
    for k=1:(tam/2)
        aux=aux+ventana(k);
    end
    if i<(tam/2)
        aux2=0;
        for k=1:i
            aux2=aux2+ventana(k);
        end
        modifwindow(i)=sqrt(aux2/aux);
    else
        aux2=0;
        for k=(i-(tam/2)+1):(tam/2)
            aux2=aux2+ventana(k);
        end
        modifwindow(i)=sqrt(aux2/aux);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k=1;
while modifwindow(k) < max(modifwindow)
    k=k+1;
end
traslape_ventana_IMDCT_inicio=k-1; %Inicio de ventana con valor de uno

while modifwindow(k) == max(modifwindow)
    k=k+1;
end
traslape_ventana_IMDCT_fin=k-1;

for i=1:512
    newventana(i)=modifwindow(i)+modifwindow(512+i);
end

```

## B.4. MDCT.m

```
%Programa para realizar la transformada coseno modificada

for k=1:traslape
    mdctpar=0;
    multi=ventana.*signal; %w(n).input(n)
    %   plot(multi);
    for n=1:N

mdctpar=mdctpar+multi(n)*cos((pi/(2*N))*(2*n+1+traslape)*((2*k)+1));
        end
        transform_cos(k)=mdctpar;
        MDCT(apunta_out+k)=mdctpar;
        mdctpar=0;
    end
```

## B.5. IMDCT.m

```
% Realiza la transformada Inversa Coseno Discreta Modificada

ventanainv= 1-ventana;

for n=1:N
    imdctpar=0;
    for k=1:traslape
        imdctpar=(imdctpar + transform_cos
(k)*cos((pi/(2*N))*(2*n+1+traslape)*((2*k)+1)));
        IMDCT= (4/N)* imdctpar * ventanainv(n);
        recuperado(n)= IMDCT;
    end
end

%-----
-

for so=0:traslape
    ndx = so+1:so+traslape ;           % current window location
    nueva = recuperado(ndx) + ventanainv; % window overlap-add
    so = so + traslape;                 % hop along now
end

%-----

figure (1)
plot(signal);

figure(2)
```

```
plot (ventana)

figure (3)
plot (ventana)
hold on;
plot (ventanainv, 'r')
hold off;

figure(4)
plot(multi);

figure(5)
plot(transform_cos);

figure(6)
plot(recuperado);

figure(8)
plot(nueva);
```

# Referencias Bibliográficas

- [1] FOOTE, J. Automatic Audio Segmentation A Measure of Audio Novelty. In: IEEE International Conference on Multimedia and Expo. 2000.
- [2] Leon W. Couch / Sistemas de comunicaciones digitales y analógicos / Editorial Pearson Educación / Quinta edición / 1998.
- [3] AMAT, Nuria, Documentación científica y nuevas tecnologías de la información, Madrid, Pirámide, 2003.
- [4] <http://www.audiodigital.com.bo/>
- [5] <http://www.hispamp3.com/tallerm3/como/queesunmp3.shtml>
- [6]<http://www.hispamp3.com/tallerm3/tutoriales/mp3profundidad/2.shtml>
- [7] MPEG. Coding of moving pictures and associated audio for digital storage media at up to 1.5mbit/s, part 3: audio, 1992. International Standard IS 11172-3, ISO/IEC JTC1/SC29 WG11.
- [8] Documentación e investigación audiovisual en línea. Comunicación presentada en el III Congreso de Investigadores Audiovisuales. Facultad de Ciencias de la Información, Universidad Complutense de Madrid, 10-12 noviembre 1999 (en prensa).
- [9] Miyaoka, S., Manufacturing technology of the Compact disc. In Digital Audio, op. cit., 196-201, 2005.
- [10] IEEE 754-1985 IEEE Standard for Binary Floating-Point Arithmetic. IEEE, 1985.
- [11] Antonio Bonafonte, Pablo Aibar, Nuria Castell, Eduardo Lleida, José B. Mariño, Emilio Sanchís y M. Inés Torres "Desarrollo de un Sistema de Diálogo Oral en Dominios Restringidos", Jornadas en Tecnologías del Habla, Sevilla, Noviembre 2000.
- [12] John Watkinson, the Art of Digital audio, third edition, 2004
- [13] MPEG. ISO/IEC 11172-3, Coding of Moving Pictures And Associated Audio For Digital Storage Media at up to About 1.5 Mbit/s Part 3 Audio. ISO/IEC, 2003.
- [14] MPEG. ISO/IEC 13818-7, Generic coding of moving pictures and associated audio information Part 7: Advanced Audio Coding (AAC). ISO/IEC, 2002.
- [15]Advanced Television Systems Committee Digital Audio Compression (AC3) Standard, Doc A/52/10, December 1995.

- [16] MPEG. Mpeg-2 advanced audio coding, aac, 1997. International Standard IS 13818-7, ISO/IEC JTC1/SC29 WG11.
- [17] Vries, L.B. and Odaka, K., CIRC – the error correcting code for the Compact Disc digital audio system. In Digital Audio, op. cit., 178-186, 2004.
- [18] <http://members.fortunecity.com/alex1944/>
- [19] Marinus M.Boone, “Acoustic rendering with wave-field synthesis”. ACM Siggraph and Eurographics Campfire. May 2001.
- [20] John G. Proakis and Dimitris G. Manolakis, “Digital Signal Processing: Principles, Algorithms, and Applications”, 3<sup>o</sup> Edition, Prentice Hall, pp. 614-637.
- [21] KIRANYAZ, S., QURESHI, A.F., GABBOUJ, M. A Fuzzy Approach Towards Perceptual Classification And Segmentation of MP3/AAC Audio. In: First International Symposium on Control, Communications and Signal Processing. 2004.
- [22] Ted Painter and Andreas Spanias. Perceptual coding of digital audio. Proceedings of the IEEE, 88(4):449-513. Abril 2000.
- [27] H. Fletcher. Auditory patterns. Rev. Mod. Phys., pages 47\_65, Enero 1940.
- [28] E. Terhardt. Calculation virtual pitch. Hearing Res., 1:155\_182, 1979.
- [25] E. Zwicker and H. Fastl. Psychoacoustics, Facts and Models. Springer, 1990.
- [26] MPEG. Call for proposals for new tools for audio coding, 2001. ISO/IEC Technical Report JTSC1/SC29/WG11 N3793.
- [29] Eberhard Zwicker, H. F. Psychoacoustics: Facts and Models. Information Sciences Springer, 2a edition., 1999. ISBN 3-540-65063-6.
- [28] Marina Bosi, R. E. G. Introduction to Digital Audio Coding and Standards. Engineering and Computer Science. Kluwer Academic Publishers, 1a ed<sup>o</sup>on., 2002. ISBN 1-4020-7357.
- [29] MPEG. ISO/IEC 14496-3, Coding of audio-visual objects Part 3: Audio. ISO/IEC, 1999-2001.
- [30] Moore, B. C. J. Masking in the human auditory system. Collected Papers on Digital Audio Bit-Rate Reduction, págs. 9-19, May 1996.
- [31] Shannon, C. E. A mathematical theory of communications. Bell System Technical Journal, 27:379{423, July 1948.
- [32] J.D. Johnston and A.J. Ferreira, “Sum-Difference Stereo Transform Coding” Proc. ICASSP 1992, pag 569-571

[33] Aki Härmä, "Coding Principles for Virtual Acoustic Openings" AES 22nd International Conference, pag 159-165.

[34] Sporer, E., Brandenburg. The use of multirate filter banks for coding of high quality digital audio. 6th European Signal Processing Conference (EUSIPCO), 211-214, June 1992.

[35] H.W. Strube. Linear prediction on a warped frequency scale. J. Acoust. Soc. Am.,68:1071-1076, 2003.

[36] Gamboa Guzmán Luis Eduardo. Compresion Perceptual de Audio. Noviembre 2005

[37] VENUGOPAL, S., RAMAKRISHNAN, K.R., SRINIVAS, S.H., BALAKRISHNAN, N. Audio Scene Analysis and Scene Change Detection in the MPEG Compressed Domain. In: IEEE 3rd Workshop on Multimedia Signal Processing. 1999.

[38] SHAO, X., XU, C., WANG, Y., KANKANHALLI, M.S. Automatic Music Summarization in Compressed Domain. In: ICASSP 2004, Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing. 2004.

[39] Code Composer Studio Development Tools v3.3- Getting Started guide, Texas instrument, 2007

[40] Lyrtech PADK user Guide

# Acrónimos

MPEG	Moving Picture Experts Group
ISO	International Organization for Standardization
AAC	Advanced Audio Coding
CD	Compact Disc
ISDN	Integrated Services Digital Network
DCC	Cassette Compact Digital
DAT	Digital Audio Tape
HDTV	High Definition Television
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
CDA	Converter Digital Analogic
CAD	Converter Analogic Digital
SLP	Sound Level Pressure
FFT	Fast Fourier Transform
DFT	Discret Fourier Transform
MDCT	Modified Discrete Cosine Transform
IMDCT	Inverse Modified Discrete Cosine Transform
TDAC	Time Domain Aliasing Cancelation