Proyecto de fin de carrera

Plan (97)

Reconocimiento de locutores a partir de archivos en formato MP3.

Crhistyan Czech
Fabian Miodownik
Alexis Ravaschio

Febrero 2005

Capítulo 0

Introducción

El reconocimiento de locutores, es un gran campo dentro de la ciencia y en el cual se ha investigado mucho durante los últimos años. Es una tecnología que esta en continuo desarrollo, y donde cada vez se logran resultados más asombrosos.

La mayoría de las investigaciones que se llevan a cabo hoy en día, tienen como punto de partida la señal de voz en su estado más primitivo, como una señal analógica, la cual es digitalizada, y luego procesada por diferentes bloques para llegar a las características propias de la voz del locutor. Sin embargo, en este proyecto el punto de partida va a ser otro: **la señal de voz comprimida en MP3**.

Si bien hoy en día existen formatos que tienen una performance comparable o incluso superior que la del MP3 o que se encuentran disponibles al público en forma gratuita, la popularidad con la que cuenta este formato supera la de cualquier otro. Esto es gracias a que MP3 ha sido y es actualmente la principal herramienta para el intercambio de audio en Internet. Buscamos además aprovechar que la señal en MP3 ya se encuentra comprimida, de modo que la información a analizar es sustancialmente menor.

El objetivo principal es poder adquirir las características propias de la voz de cada locutor, tratando de descomprimir lo menos posible a los archivos MP3. Una vez obtenidas dichas características, se probarán distintas técnicas de *Identificación* y *Verificación* de locutores.

Durante el desarrollo del proyecto nos topamos con una gran variedad de obstáculos que iban desde la falta de fuentes de información hasta problemas que en un principio no parecían tener mayor grado de dificultad como lo era adquirir señales de voz en MP3, pasando por problemas de diseño y desarrollo de algoritmos. Todos estos problemas son discutidos junto con las soluciones adoptadas.

El enfoque de este trabajo es el siguiente: en los primeros 3 capítulos presentamos una descripción teórica de los temas directamente involucrados con el proyecto. En el capítulo 4 se discuten los parámetros de diseño, así como el tipo de características que se consideraran. En el capítulo 5 se describe de manera específica como fueron implementados los sistemas de *Identificación, Verificación* y *Búsqueda* así como los resultados obtenidos. Finalmente en el capítulo 6 se presentan las conclusiones.

Capítulo 1

Codificación de señales de voz[15,16]

Este capítulo pretende tratar en forma resumida el proceso de codificación de señales de audio. El desarrollo aquí expuesto, que incluye codificación PCM y codificación en subbandas, servirá como introducción a los temas que se tratan en secciones posteriores.

1.1 Introducción

Para poder representar una señal como una serie de valores discretos recurrimos al proceso de digitalización. Al digitalizar la señal tanto el almacenamiento como el procesado y la transmisión de la misma ofrecen ventajas muy significativas frente a los métodos analógicos. La tecnología digital es más avanzada, ofrece una menor sensibilidad al ruido en la transmisión, y la capacidad de incluir códigos de protección frente a errores, así como encriptación. Sin embargo presenta como desventaja el gran ancho de banda requerido para su transmisión, lo que motiva a realizar diversos estudios respecto a la compresión de datos.

El proceso de digitalización está compuesto de dos etapas: *muestreo* y *cuantificación*. Se llama muestreo a la acción de discretizar el eje del tiempo en segmentos iguales denominados ***período de muestreo (Ts)***. La ***frecuencia de muestreo (fs)*** es la inversa del período entre una muestra y la siguiente. El teorema de *Nyquist* determina la frecuencia mínima de muestreo en 2B donde B es la mayor componente en frecuencia de la señal a muestrear. Por tanto, siendo el margen superior de la audición humana en torno a los 20Khz, la frecuencia que garantiza un muestreo adecuado para cualquier sonido audible será de unos 40 Khz. Concretamente, para obtener sonido de alta calidad se utilizan frecuencias de 44,1Khz, como en el caso del *CD*.

Luego del de muestreo se realiza la cuantificación, que, en su forma más sencilla, consiste en discretizar la amplitud de la señal. Cuantos más bits se utilicen para la división del eje de la amplitud, mayor resolución obtendremos y por tanto menor el error al atribuir una amplitud discreta a un valor continuo en cada instante. Por ejemplo, 8 bits ofrecen 256 niveles de cuantización (2^8) y 16 bits ofrecen 65536 niveles. La división del eje se puede realizar a intervalos iguales, o según una determinada función de densidad, buscando más

resolución en ciertos tramos si la señal en cuestión tiene más componentes en cierta zona de intensidad. El proceso completo se denomina habitualmente PCM (*Pulse Code Modulation*).

La digitalización de la señal mediante PCM es la forma más simple de codificación, y es la utilizada por los CD's. Como toda digitalización, añade ruido a la señal, generalmente indeseable.

Como hemos visto, cuantos menos bits se utilicen en el muestreo y la cuantificación, mayor será el error al aceptar valores discretos para la señal continua, es decir, el ruido será mayor. Para evitar que el ruido alcance un nivel excesivo hay que emplear un gran número de bits, de forma que para 44,1Khz utilizando 16 bits para cuantificar la señal, uno de los dos canales de un CD produce más de 700Kbps. Gran parte de esta información es innecesaria y ocupa un ancho de banda que podría liberarse, a costa de aumentar la complejidad del sistema decodificador obteniendo una cierta pérdida de calidad. El compromiso entre ancho de banda, complejidad y calidad es lo que diferencia los diferentes estándares del mercado.

Calidad	*Muestreo*	*Bits por muestra*	*Modo*
Teléfono	*8Kbps*	*8 bits*	*Mono*
Radio AM	*11025bps*	*8 bits*	*Mono*
Radio FM	*22050bps*	*16 bits*	*Estéreo*
CD ROM	*44100bps*	*16 bits*	*Estéreo*

Tabla 1.1 Comparación de formatos de calidad de audio

Un modo mejor de codificar la señal es mediante PCM no lineal o cuantización logarítmica, que consiste en dividir el eje de la amplitud de tal forma que los escalones sean mayores cuanta más energía tiene la señal, con lo que se consigue una relación señal ruido igual o mejor con menos bits. Con este método se puede reducir el canal de CD audio a 350 kbps, lo cual evidentemente es una mejora sustancial, aunque puede reducirse mucho más.

En lo que respecta a la codificación de voz podemos identificar dos grandes ramas de codificadores, los codificadores de forma de onda y los codificadores de fuente.
Los codificadores de forma de onda son aquellos que aprovechan la redundancia de las muestras de voz de forma de lograr una mejorar eficiencia que PCM con cuantización uniforme.

Una sub categoría de los codificadores de forma de onda son aquellos que se basan en las características espectrales de la voz. Decimos que es una sub categoría pues las características espectrales de la voz se obtienen mediante alguna transformada (como la del coseno) de la forma de onda de la señal, teniendo en fin el mismo origen.

Por otro lado tenemos los codificadores de fuente. Estos se nutren de las características de la voz como tal, y no como una onda. Buscan modelar a la fuente que en este caso es el aparato fonador. Estos sistemas son específicos para la voz, por lo que no tienen la versatilidad de los de la familia de PCM y no se pueden usar en otras fuentes más que la voz.

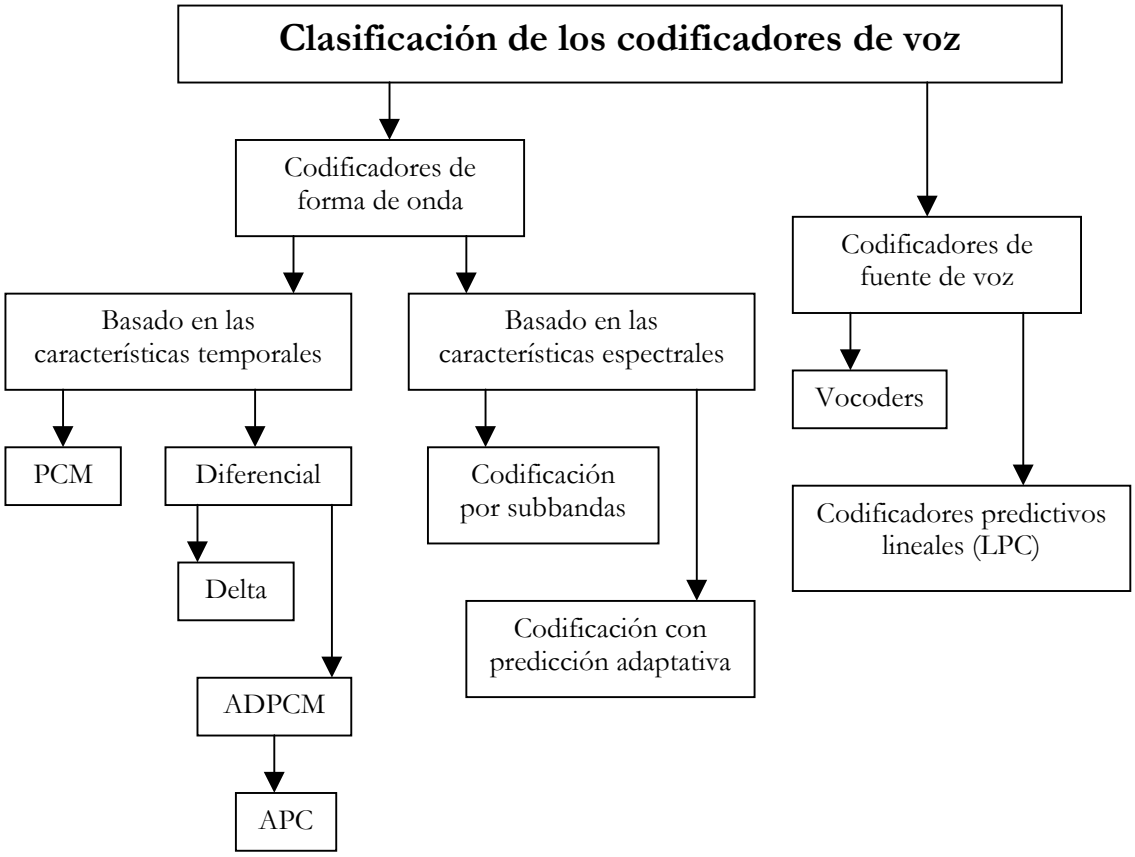


Figura 1.1 Diagrama de clasificación de los codificadores de voz.

1.2 Codificación en subbandas (SBC)

La codificación en subbandas a diferencia de los métodos específicos de codificación de fuentes de audio, como la voz, puede codificar cualquier señal sin importar su origen, ya sea voz, música o cualquier tipo de sonido. Este es el tipo de codificación utilizada por el estándar MPEG Audio.

A pesar de esto es factible apuntar la codificación en subbandas hacia la codificación de voz obteniendo buenos resultados. De esta manera podemos hacer uso de la falta de uniformidad del espectro de la voz y codificar con una cantidad mayor de bits las bajas frecuencias, así como también despreciar aquellas que presentan niveles bajos de energía. Tomando esto en cuenta podemos considerar a la codificación en subbandas como un método de controlar el ruido de cuantización sobre el espectro de la señal.

El oído humano no percibe el ruido de cuantización de igual forma a todas las frecuencias, en consecuencia hay una ventaja comparativa en dividir el espectro en subbandas para ser codificadas individualmente.

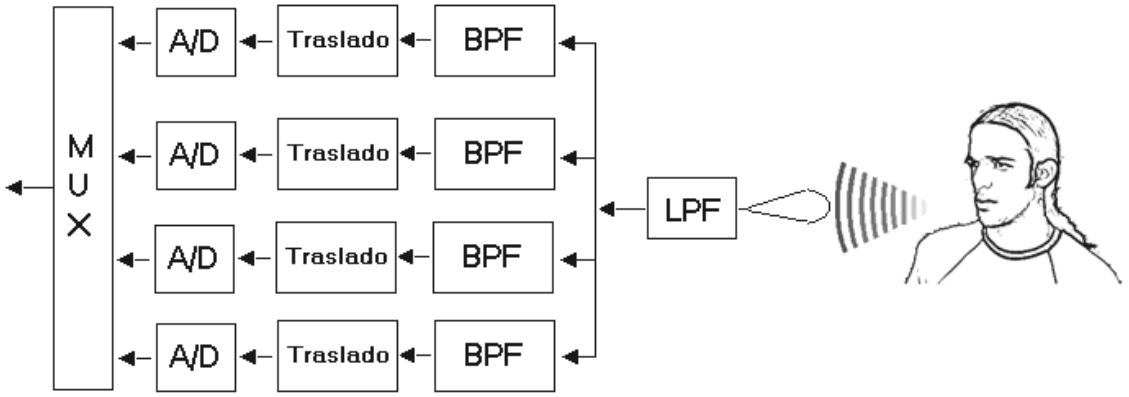


Figura 1.2. Diagrama de bloques de un sistema por codificación en subbandas.

Para aclarar los conceptos anteriormente mencionados consideremos el siguiente ejemplo: En la figura 1.3 tenemos una señal de voz de 1.8 segundos, la cual nos servirá para comprobar la validez del sistema de codificación.

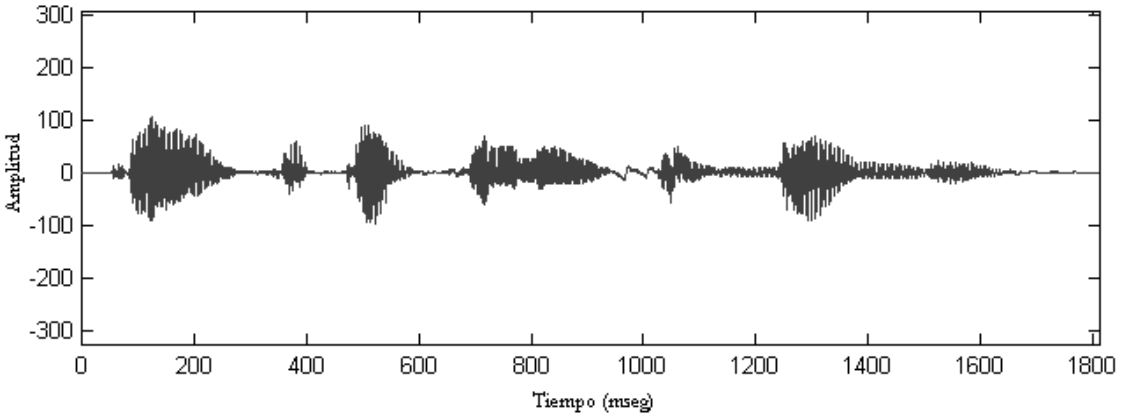


Figura 1.3. Señal de voz en el tiempo.

Si ahora analizamos el espectro de esta señal de voz, mediante el espectrograma de la figura 1.4, notamos que, como era de esperar la mayor parte de la potencia de la señal se encuentra en baja frecuencia.

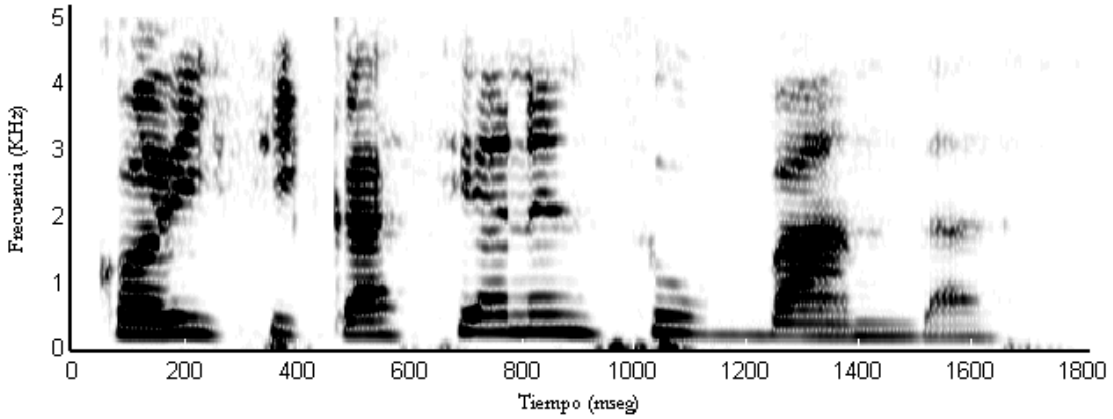


Figura 1.4. Espectrograma de la señal de la figura 1.3

Siendo esto así procedemos a diseñar los filtros de la siguiente manera:

	Rango de frecuencias(Hz)	*Ancho de banda (Hz)*	*Frecuencia de muestreo (Hz)*
Banda 1	*200 – 700*	*500*	*1000*
Banda 2	*700 – 1310*	*610*	*1220*
Banda 3	*1310 – 2020*	*710*	*1420*
Banda 4	*2020 – 3200*	*1180*	*2360*

A estas bandas les adjudicaremos 6, 5, 4 y 3 bits por muestra respectivamente.

Entonces si queremos hallar la taza de transmisión debemos sumar la taza de transmisión de cada subbanda. Tenemos entonces: R1 = 1000·6 b/s, R2 = 1220·5 b/s, R3 = 1420·4 b/s, R4 = 2360·3 b/s sumando un total de 24,86 Kbps.
A modo más general, el principio básico del SBC es la limitación del ancho de banda de la señal codificada mediante la eliminación de información en frecuencias enmascaradas[1]. El resultado no es el mismo que la señal original, sin embargo, el oído humano no es capaz de notar la diferencia. Como se vera más adelante este es uno de los principios en los que se basa el MP3.

La mayoría de los codificadores SBC utilizan el mismo esquema: primero tenemos un filtro o un banco de ellos, o algún otro mecanismo que descompone la señal de entrada en subbandas, a continuación se aplica un modelo psicoacústico[2] que analiza las bandas y determina los niveles de enmascaramiento utilizando los datos psicoacústicos de que dispone. Considerando estos niveles de enmascaramiento, se cuantifican y codifican las muestras de cada banda, si en una frecuencia dentro de una banda hay una componente por debajo de dicho nivel, se desecha. Si lo supera, se calculan los bits necesarios para cuantificarla y se codifica. La decodificación es mucho más sencilla, ya que no hay que aplicar ningún modelo psicoacústico. Se analizan los datos y se recomponen las bandas y sus muestras correspondientes.

[1] El concepto de frecuencias enmascaradas se trata en el apéndice A.2.
[2] El modelo psicoacústico se analiza en el apéndice A.2.

Capítulo 2

Reconocimiento de locutores[8,9,10]

2.1 Introducción

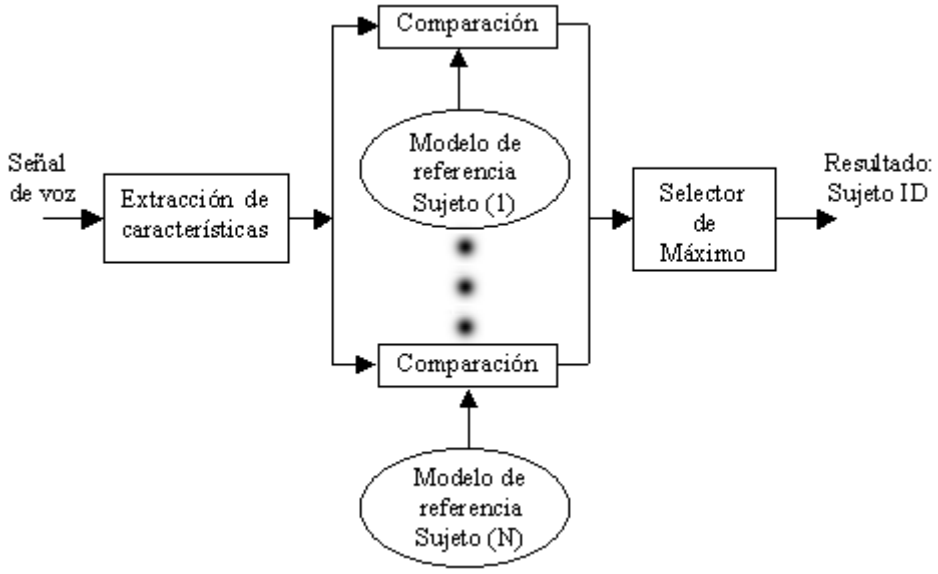
El reconocimiento de locutores es el proceso de reconocer automáticamente, sin intervención de un operador humano, a quién está hablando basándose en características personales incluidas en las señales de voz. Esta técnica hace posible el uso de la voz del locutor para verificar su identidad y controlar el acceso a servicios como por ejemplo, compras telefónicas, acceso a cuenta de banco por teléfono, correo de voz, control de seguridad para áreas de información confidencial y acceso remoto de computadoras.

El reconocimiento automático de locutores (RAL) se enmarca dentro de un área de trabajo más amplia: el reconocimiento de personas mediante parámetros biológicos cuyo valor es diferente en cada persona, como por ejemplo el iris, la forma de la cara o las huellas dactilares. Son los que se denominan parámetros biométricos.

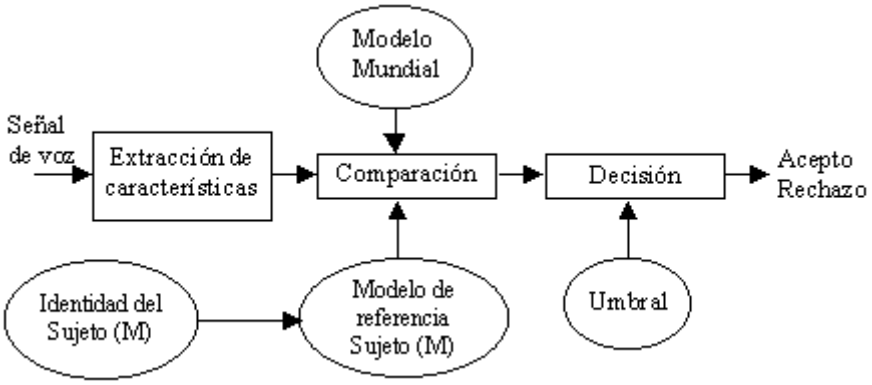
2.2 Conceptos y principios básicos.

El reconocimiento de locutores puede dividirse en dos tares básicas: *Identificación* y *Verificación*.

Identificación Automática del Locutor (IAL): El objetivo de esta aplicación es, dada una señal de voz, determinar, dentro de un grupo de personas predeterminadas, la identidad de su "propietario". Hablamos de *IAL* de *Grupo Cerrado* si el locutor desconocido es con certeza uno de los del grupo, y de *IAL* de *Grupo Abierto* si el locutor puede ser alguien ajeno a ese grupo de personas. En el primer caso la respuesta del sistema será siempre una identidad, mientras que en el segundo existe la posibilidad de la respuesta "locutor rechazado" al no pertenecer al grupo de personas de referencia.



a) Identificación de locutores



b) Verificación de locutores

Figura 2.1 Estructuras básicas de los sistemas de *Identificación* y *Verificación*.

Verificación Automática del Locutor (VAL): También llamada autenticación o detección, aquí el objetivo es verificar si el locutor es quien asegura ser. Mas formalmente, dada una señal de voz ***x*** y un locutor ***S***, la tarea a realizar es ver si ***x*** fue dicha por ***S*** o no; la respuesta del sistema será binaria: identidad aceptada o rechazada. Esta decisión binaria puede ser reformulada como un test de hipótesis entre las siguientes hipótesis:

$\mathbf{H_0}$: x pertenece al locutor buscado.
$\mathbf{H_1}$: x no pertenece al locutor buscado.

La decisión óptima se obtiene a partir de:

$$T(x) = \frac{f(H_0 / x)}{f(H_1 / x)} \begin{cases} > \Theta & acepta\ H_0 \\ < \Theta & acepta\ H_1 \end{cases} \qquad (2.1)$$

si las funciones de probabilidad f(...) se conocen exactamente y para un umbral Θ dado. T(x) se conoce como test de cociente de verosimilitudes (likelihood ratio test). Algunas

elecciones comunes para f(...) son *Modelos Ocultos de Markov (HMM)*, *Modelos de Mezclas Gaussianas (GMM)* y *Redes Neuronales Artificiales (ANN)*.

Un sistema de *Verificación* de locutores típico opera de la siguiente manera: el modelo definido por la función $f(H_1/x)$ se entrena con las voces de muchos usuarios diferentes y se denota como *UBM (Universal Background Model)* o modelo universal. El modelo del usuario definido por la función $f(H_0/x)$ se entrena solamente con la voz del locutor buscado. Finalmente cuando un locutor clama cierta identidad, se toma su señal de voz y se decide.

Hay diversas formas de medir la efectividad de este tipo de sistemas, una de las más utilizadas es la denominada tasa de equi-error *(EER)*. Esta es el error del sistema cuando el umbral de decisión es tal que el porcentaje de falsas aceptaciones (FA) es igual al de falsos rechazos (FR) (figura 1.2). Las FA se dan cuando aceptamos a un locutor que no es el buscado y los FR se dan cuando rechazamos al locutor que estamos buscando.

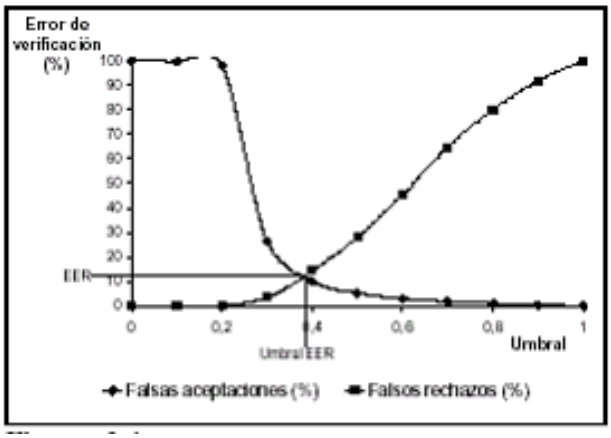


Figura 2.2. Curvas de falsas aceptaciones (FA) y falsos rechazos (FR).

Según el contenido de la señal de voz empleada las modalidades de reconocimiento se clasifican en *dependientes del texto* e *independientes de texto*.

En los métodos *independientes del texto*, lo que el locutor dice no se tiene en cuenta ni a la hora de hallar sus características, ni tampoco en la fase de testeo. Si se tiene en cuenta la manera en que lo dijo. La señal de voz es escogida por el locutor en forma aleatoria, de manera que el sistema desconoce a priori su contenido.

Por otro lado en los métodos *dependientes del texto*, el reconocimiento de la identidad del locutor está basado en frases o palabras específicas que el sistema conoce de antemano, típicamente claves, cédula de identidad, etc. El texto con el que se hallan las características del locutor y el usado para reconocerlo son los mismos.

Todas estas formas de reconocimiento tienen sus ventajas y desventajas, y además cada una de estas requiere de un tratamiento específico. Así, por ejemplo, el rendimiento de los sistemas que utilizan reconocimiento *dependiente del texto* es mayor, sin embargo son más vulnerables cuando la clave es conocida por personas ajenas a su propietario. El riesgo de la clave es evitado usando independencia del texto, pero el rendimiento de los sistemas que utilizan esta opción es menor, y además, el peligro radica en que cualquier grabación de la voz de una persona puede ser utilizada para un acceso no permitido. Los sistemas

independientes del texto necesitan también más entrenamiento que los *dependientes del texto* ya que las características específicas de la frase no están disponibles.

Como alternativa surge una tercera vía intermedia entre las dos anteriores: t*exto solicitado (Text Prompted)*. Aquí es el sistema el que escoge el contenido de la muestra de voz. Esta elección puede ser hecha dentro de un conjunto reducido de palabras o frases, o sin ningún tipo de restricción en el texto a pronunciar por la persona. En el reconocimiento dependiente del texto, el sistema conoce a priori el contenido de la señal de voz, por lo tanto los modelos pueden ser mejor aproximados mejorando el reconocimiento. Al mismo tiempo, se evita la violación de la seguridad del sistema por la perdida del secreto de la clave (ya que esta no es fija), pues es imposible conocer a priori el texto que pedirá el sistema.

A grandes rasgos, todo sistema de reconocimiento se compone de dos bloques principales que son el de *extracción de características* y el de *comparación de características o testeo*. La extracción de características es el proceso en el cual se extrae una pequeña cantidad de información de la señal para luego representar al locutor mediante un modelo propio. Por otro lado durante en la fase de comparación de características, la voz de entrada al sistema es comparada con los modelos de referencia almacenados para luego tomar una decisión.

El reconocimiento automático de locutores se basa en la premisa de que la voz de una persona exhibe características únicas de ese locutor. Sin embargo existen una gran cantidad de factores que alteran la voz haciendo la tarea de reconocimiento más complicada de lo que ya es. La principal fuente de distorsión proviene de los mismos locutores. Las señales de voz en las sesiones de entrenamiento y testeo pueden ser enormemente diferentes debido a hechos como el cambio de la voz con el paso del tiempo (*variabilidad intra-locutor*), condiciones de salud (i.e. el locutor está resfriado), velocidad a la que se habla, etc.

La manera de resolver esto es crear un modelo más robusto que contenga diversas muestras de voz cada una con un estado de ánimo diferente. Lamentablemente, en la práctica, no siempre se dispone de tal cantidad de datos.

Hay también otros factores que representan un desafío para la tecnología de reconocimiento de locutores. Algunos ejemplos son el ruido acústico y las variaciones en los ambientes de grabación (ej. el locutor utiliza micrófonos diferentes).

2.3 Extracción de características de la voz[8].

Es el proceso de convertir la señal de voz en algún tipo de representación paramétrica sumamente compactada para su posterior análisis. Como podemos ver en la figura 2.3, la señal de voz varía muy lentamente en el tiempo cuando se la observa en periodos entre los 5 y 100 ms (decimos que tiene características estacionarias). Sin embargo cuando la observamos por más de 200 ms las características de la señal cambian para reflejar los diferentes sonidos de la voz. Por esto la mejor manera de representar a la señal de voz es mediante un análisis espectral de cortos periodos de tiempo.
n

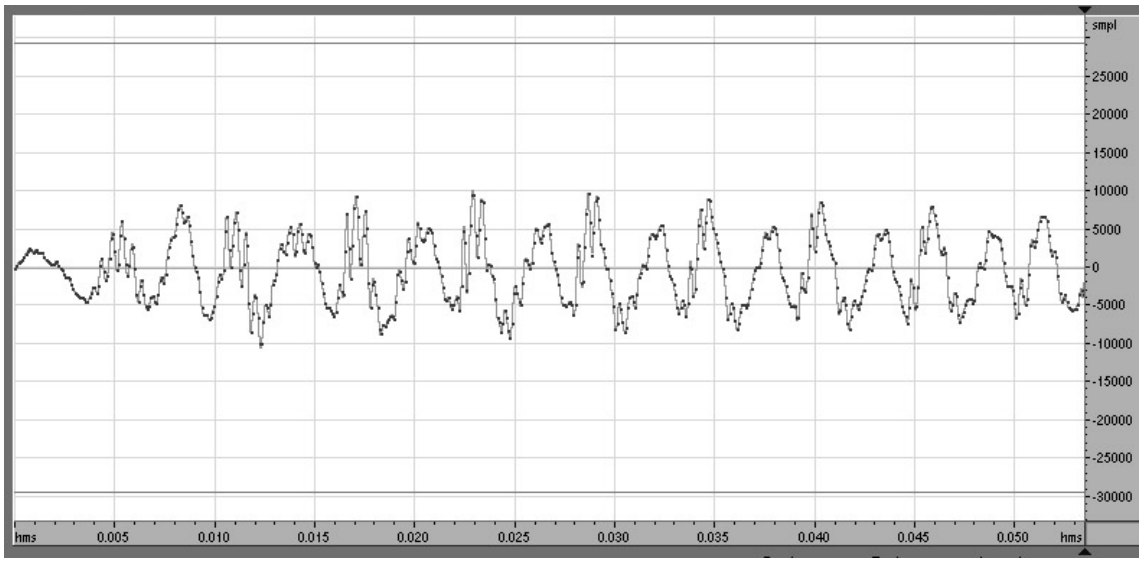


Figura 2.3. Ejemplo de señal de voz en el tiempo.

Existe un gran marco de posibilidades para representar a la señal de voz en un sistema de reconocimiento, entre estas se encuentran los coeficientes ***LPC*** [3] (Linear Prediction Coefficients) y los ***MFCC*** (Mel Frequency Cepstrum Coefficients).

Tal cual fue planteado en la introducción, el objetivo principal del proyecto es descomprimir los archivos MP3 lo menos posible. Como los ***LPC*** se calculan a partir de la señal temporal deberíamos descomprimir el archivo completamente para hallarlos. Por otro lado los ***MFCC*** se pueden calcular trabajando en el dominio de la frecuencia.

Luego de estudiado el capítulo 3 se verá que es posible calcular los ***MFCC*** en algún paso intermedio previo a la descompresión total. Por esta razón elegiremos los coeficientes ***MFCC***.

2.3.1 MFCC

Los MFCC se basan en la variación de las bandas críticas del oído con la frecuencia[3]. Para capturar las características fonéticas más importantes del habla se usan filtros separados de forma lineal para las bajas frecuencias y de manera logarítmica para las altas frecuencias. Esta separación se representa mediante la llamada escala de frecuencias Mel.

Un Diagrama de bloque del proceso para obtener los ***MFCC*** a partir de una señal de audio se ve en la figura 2.4.

Entramado: En este paso la señal de voz continua es dividida en tramas de N muestras, en donde tramas adyacentes están separadas por M muestras (M<N). La primera trama consiste en la primeras N muestras. La segunda trama empieza M muestras después que la primera y esta solapada por N-M muestras. Se hace lo mismo con el resto de las tramas hasta procesar toda la señal de voz.

[3] Una descripción detallada sobre LPC se puede encontrar en el apéndice A-1.

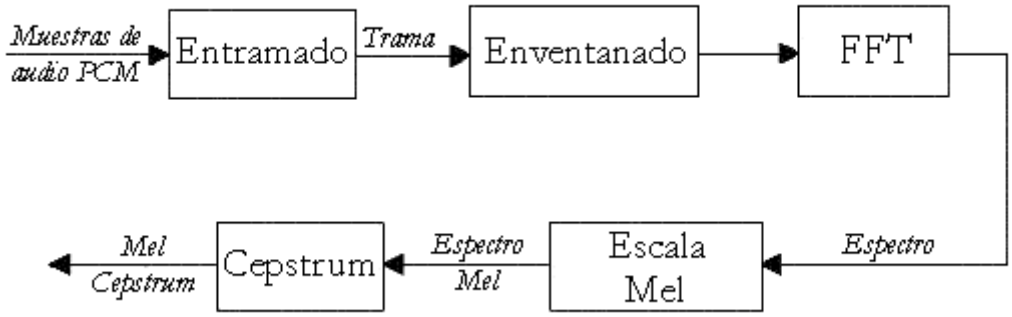


Figura 2.4. Proceso de obtención de los MFCC.

Enventanado: El paso siguiente en el proceso es enventanar cada trama para así minimizar las discontinuidades de la señal en el comienzo y fin en las mismas. La idea es minimizar la distorsión espectral empleando la ventana para llevar a cero el principio y final de cada trama. Si definimos la ventana como w(n), $0 \le n \le N - 1$ en donde N es el número de muestra en cada trama, entonces el resultado de la señal enventanada es la señal:

$$y_1(n) = x_1(n).w(n) \; con \; 0 \le n \le N-1 \quad (2.2)$$

Típicamente se utiliza la ventana de Hamming que tiene la forma:

$$w(n) = 0.54 - 0.46.\cos(\frac{2\pi.n}{N-1}) \; con \; 0 \le n \le N-1 \qquad (2.3)$$

FFT: A continuación se hace la transformada rápida de Fourier, que convierte cada trama de N muestras del dominio temporal al dominio de la frecuencia. El resultado de este paso se conoce como el espectro de la señal.

Transformación a escala Mel: Los estudios psicoacústicos muestran que la percepción humana de las frecuencias no sigue una escala lineal[4]. Por esto se debe hacer un escalado de las frecuencias en Hz a una escala subjetiva conocida como la escala Mel. Lo que se hace es dividir el espectro en un banco de filtros, mucho más estrechos y espaciados en forma lineal por debajo de 1 kHz y muy amplios y espaciados de manera logarítmica por encima de esta cantidad. De este modo, se da mayor importancia a la información contenida en las bajas frecuencias en consonancia con el comportamiento del oído humano.

Este banco de filtros tiene un diseño triangular 50% solapado y con ancho de banda y separación determinado por un intervalo Mel constante. En la figura 2.5 se muestra este banco de filtros para solapamiento de 200 Mels.

A modo de referencia quedan definidos 1000 Mels como el tono de una señal de 1kHz de 40 dB por encima del umbral de audición. En consecuencia se puede usar la aproximación:

$$Mel(f) = 2595.\log_{10}(1 + f/700) \qquad (2.4)$$

[4] Ver apéndice A-3.3.

Se le aplica al espectro de la señal el banco de filtros y luego se suma la salida para cada banda, dando lugar a los distintos coeficientes Mel, denotados por S_K con k = 1,2,.....,K. siendo K la cantidad de filtros.

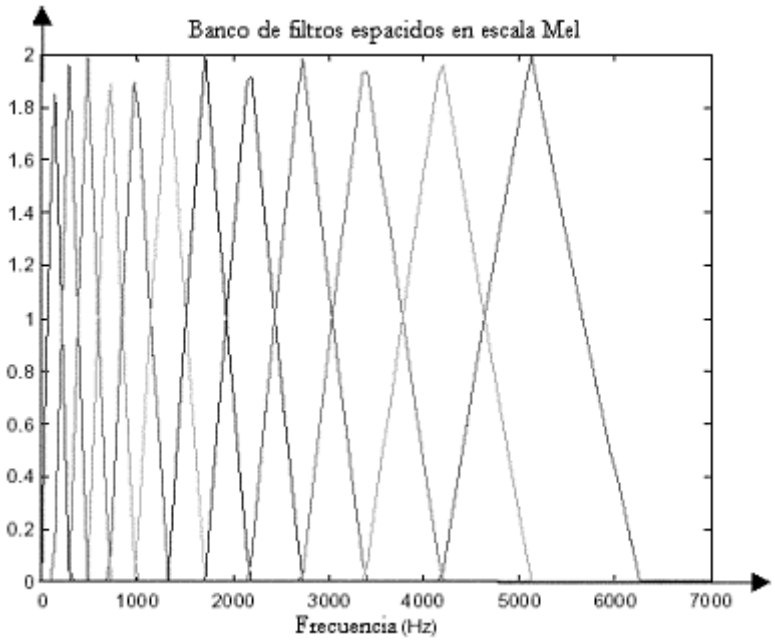


Figura 2.5. Ejemplo de banco de filtros espaciados en escala Mel.

Cepstrum: En este último paso, se convierte el espectro Log Mel al tiempo. Para esto primero se calcula el logaritmo de S_K. Luego como los coeficientes Mel (y su logaritmo) son números reales, se los puede convertir al dominio temporal usando la Transformada Discreta del Coseno (***DCT***). Este resultado es el que llamamos ***MFCC***.

Podemos calcular los ***MFCC*** mediante:

$$c_n = \sum_{k=1}^{K} (\log S_k) \cos\{n(k - \frac{1}{2})\frac{\pi}{K}\} \quad \text{con } n = 1,.., K\text{-}1 \qquad (1.5)$$

los elementos de la ecuación representan:

- k es la banda de frecuencias.
- n es el coeficiente MFCC en cuestión.
- ($\log S_k$) es el logaritmo del los coeficientes Mel.
- K es el número total de bandas o filtros.

2.4 Modelado de características.[17]

Las técnicas de modelado de características más usadas en reconocimiento de locutores incluyen:

- Cuantización Vectorial (VQ).
- Alineación temporal dinámica (DTW), basada en el cálculo y comparación de distancias.

- Técnicas probabilísticas: Modelos ocultos de Markov (HMM), Modelos de Mezclas Gaussianas (GMM).

Cada uno de estos métodos tiene sus particularidades así como sus ventajas y desventajas. En este apartado se describen en detalle los métodos VQ y GMM, que fueron los empleados en nuestro proyecto, pero también se da una somera descripción de los otros dos métodos a modo de introducción. Plantearemos también las razones que determinaron nuestra elección.

2.4.1 Vector Quantization [12]

La cuantización vectorial es el mapeo de un espacio Euclídeo k - dimensional en un conjunto *C* finito con N puntos de salida (llamados palabras de código o centroides). Es decir:

$$Q:\Re^k \rightarrow C \qquad (2.6)$$

Donde $C=\{y_1, y_2 ....y_N\}$ e $y_i \in \Re^k \ \ \forall i=1,2...N$. El conjunto *C* es llamado *codebook* y tiene N vectores diferentes de dimensión k. Asociado con cada uno de los N puntos del *codebook* tenemos una región o celda. La i-ésima celda queda definida por:

$$R_i = \left\{x \in \Re^k : Q(x) = y_i\right\} \qquad (2.7)$$

De la definición de celda se deduce que $\bigcup_i R_i = \Re^k$ y $R_i \cap R_j \neq \phi \ \ \forall i \neq j$ o sea que las celdas forman una partición de $\Re^k$.
Un cuantizador vectorial puede ser descompuesto en dos funciones más elementales como lo son: un codificador y un decodificador vectorial.

El Codificador Vectorial. El codificador *E* es el mapeo de $\Re^k$ en un conjunto de índices *J:* $E:\Re^k \rightarrow J$. Es importante notar que una partición dada de $\Re^k$, determina completamente la forma como el codificador asignará un índice a cada entrada dada. El codificador no necesita conocer el *codebook* para cumplir su función.

El Decodificador Vectorial. El decodificador D mapea el conjunto de índices *J* en el conjunto de representaciones C: $D:J \rightarrow C$.
Análogamente al caso del codificador, dado un codebook tenemos perfectamente determinado como el decodificador generará la salida a partir de un índice dado. El procedimiento de decodificación viene dado por una tabla de correspondencias, no es necesario conocer la geometría de la partición para llevarla a cabo.

Cuantizadores de vecino más cercano. Una clase especial de cuantizadores vectoriales, es la de los llamados cuantizadores Voronoi o de vecino más cercano. Veremos más adelante que un cuantizador vectorial debe pertenecer a esta clase para ser óptimo en el sentido de minimizar la distorsión promedio.

Se define un cuantizador de vecino más cercano como uno cuya partición en celdas viene dada por:

$$R_i = \left\{x / d(x, y_i) \leq d(x, y_j) \ \forall \ j \in J\right\} \qquad (2.8)$$

donde el *codebook* está dado por $C = \{y_i\} \forall i \in J$.

Esto tiene como ventaja que durante el proceso de codificación no se necesita una descripción geométrica de las celdas de manera explícita. En lugar de eso, basta con conocer la distancia al *codebook* almacenado.

2.4.1.1. Condiciones de optimalidad.

En esta sección se estudiarán las propiedades de optimalidad de los cuantizadores vectoriales. Estas propiedades son de gran ayuda para el diseño de cuantizadores pues proporcionan condiciones simples que un cuantizador vectorial debe cumplir para ser óptimo y de ellas se deduce una técnica iterativa sencilla para mejorar un cuantizador dado (*Algoritmo de Lloyd*).

La meta principal en el diseño de un cuantizador vectorial es encontrar un *codebook* y una partición que minimicen una medida de distorsión, considerando la secuencia completa de vectores a ser codificados.

Para el caso continuo, la distorsión *D* queda definida como: $D = E[d(x, Q(x))]$ o lo que es lo mismo:

$$D = \int d(x, Q(x)) f_X(x) dx \qquad (2.9)$$

Donde $f_X(x)$ es la *pdf* conjunta del vector *x* y la integral es sobre todo el espacio k - dimensional.

Existen condiciones necesarias para que un codificador sea óptimo para un determinado decodificador y viceversa. El codificador queda determinado para la partición R_i y el decodificador queda determinado por el *codebook*. Entonces las condiciones de optimalidad son:

- Condición de vecino más cercano (*E* óptimo dado *D*)
- Condición de centroide (*D* óptimo dado *E*)
- Condición de probabilidad cero en los bordes

Condición de vecino más cercano. En primer lugar se considerará la optimización del codificador, dejando al decodificador fijo. Para un *codebook* dado, una partición óptima es la que satisface la condición de vecino más cercano: es decir a la región R_i se le asignan todos aquellos *i* que distan de y_i menos que a cualquier otro vector del código.

Para un *codebook* la distorsión media puede ser acotada por:

$$D = \int d(x, Q(x)) f_X(x) dx \geq \int \min_{i \in I} d(x, y_i) f_X(x) dx \qquad (2.10)$$

y la igualdad se alcanza si *Q(x)* es la palabra de código que genera menor distorsión, es decir, el vecino más cercano. Entonces la partición óptima satisface: $Q(x) = y_i$ solo si

$$d(x, y_i) \leq d(x, y_j) \ \forall \ j \in J \qquad (2.11)$$

Condición de centroide. Ahora se estudiará la optimización del decodificador, dado al codificador. Podemos expresar la distorsión como:

$$D = \sum_{i=1}^{N} \int_{t_i}^{t_{i+1}} (x - y_i)^2 f_X(x) dx \qquad (2.12)$$

Basta con derivar la distorsión respecto a y_i para hallar el y_i óptimo. Finalmente cada región R_i será representada por su centroide definido como:

$$y_i = \frac{\int_{R_i} x f_X(x) dx}{\int_{R_i} f_X(x) dx} \qquad (2.13)$$

También es fácil ver que el centroide concuerda con la definición de centro de gravedad, de modo que el centroide es único. Para el caso discreto, la definición de centroide sigue siendo válida y en el caso en que cada vector tenga la misma probabilidad y la medida de distorsión sea la asociada al error cuadrático medio, el centroide coincide con el promedio aritmético.

Este resultado asume que todas las regiones tienen probabilidades distintas de cero de contener al vector de entrada. De no ser así se tiene el problema de *celda vacía*. Para una región con probabilidad nula el centroide no esta definido, además tampoco tiene sentido malgastar una palabra de código en semejante región. La solución que se toma generalmente, consiste en eliminar la celda vacía y partir la celda con mayor distorsión en dos. De esta manera el tamaño del *codebook* se mantiene constante y la distorsión disminuye.

Condición de probabilidad cero en los bordes. Existe una tercera condición necesaria para la optimalidad de un cuantizador que es útil en el caso discreto. Esta es la condición de probabilidad cero en los bordes o lo que es lo mismo que el vecino más cercano sea único.

Si un punto de la entrada, x_0, coincide con la frontera de las regiones R_i y R_j entonces dos particiones diferentes se pueden formar, asignando x_0 a R_i o bien a R_j. En ambos casos la distorsión promedio es la misma, sin embargo, estamos cambiando de celda un punto de entrada con probabilidad distinta de cero, lo cual mueve los centroides de R_i y R_j, lo que implica que el codebook ya no es óptimo para la nueva partición.

Esta condición se cumple siempre cuando la entrada es una variable aleatoria continua pues el borde tiene volumen cero (y por lo tanto probabilidad cero). Esta condición es útil para el caso discreto pues la probabilidad puede ser colocada en puntos del borde, esto es, un vector de la secuencia de entrenamiento puede ser equidistante de dos vectores del código.

2.4.1.2. Diseño de un cuantizador vectorial.

Las condiciones necesarias estudiadas para la optimalidad proporcionan las bases para un algoritmo iterativo que mejore un cuantizador vectorial dado. Se ha visto que las condiciones de optimalidad no aseguran que el cuantizador sea globalmente óptimo. Por lo tanto, la condición inicial torna un aspecto importante a tener en cuenta para obtener un

buen resultado final: si se parte de un cuantizador bien diseñado la probabilidad de converger al óptimo será mayor.

2.4.1.3. Técnicas para diseñar cuantizadores.

Se empezará estudiando algunas formas de obtener un buen codebook inicial. De hecho si éste es lo suficientemente bueno, no valdrá la pena correr algoritmos de mejora.

Random Coding. La idea más simple para encontrar un codebook de tamaño N es elegir aleatoriamente los vectores del código de acuerdo con la distribución de probabilidad de la fuente, lo que puede ser visto como un diseño Monte Carlo. La opción más simple cuando se diseña el codebook basándose en una secuencia de entrenamiento es elegir los primeros N vectores. Si la secuencia de entrenamiento es muy correlacionada es mejor elegir entonces, uno de cada K vectores. Desgraciadamente, este codebook no tendrá ninguna estructura útil y podrá comportarse bastante mal.

Pruning. Esta técnica se basa en la idea de comenzar con todos los vectores de la secuencia de entrenamiento como candidatos a integrar el codebook; y eliminarlos uno a uno según cierto criterio hasta que el conjunto final resulte ser el codebook.

Un método posible podrá ser el siguiente: se considera el primer vector de la secuencia como el primer vector del codebook. Luego se calcula la distorsión entre éste y el siguiente vector de entrenamiento. Si la distorsión es mayor que cierto umbral el vector siguiente se incluye también, sino es desechado. Con cada vector de entrenamiento nuevo se busca su vecino más cercano entre los vectores ya integrados al codebook, y, si la distorsión entre ambos es mayor que cierto umbral, el vector se integra al codebook sino se desecha. Este paso se repite hasta que se completa el codebook. Para una secuencia de entrenamiento finita puede resultar que no se encuentre el número de vectores necesario; en este caso se debe reducir el umbral de decisión y re comenzar.

Splitting. Inicialmente se elige el centroide de la secuencia de entrada y_0 como codebook de resolución[5] 0 (con un solo elemento). Esta única palabra puede ser divida en dos palabras de código: y_0 e $y_0 + \varepsilon$ donde ε es un vector de módulo pequeño.

Este nuevo codebook tiene dos elementos por lo cual no puede ser peor que el original. El algoritmo iterativo de mejora puede ahora aplicarse a este codebook para obtener un buen codebook de resolución 1. Como siguiente paso de divide cada uno de los vectores del codebook en dos, repitiendo lo anterior. Se continúa de ésta manera, hallando un codebook de resolución r + 1 partiendo de un buen codebook de resolución r. Esta técnica provee un algoritmo completo de diseño de codebook desde la secuencia de entrenamiento.

2.4.1.4. El Algoritmo de Lloyd.

Se discute ahora con detenimiento un algoritmo iterativo de mejora de codebook. Si la iteración continúa hasta la convergencia, un buen cuantizador vectorial (se desea que óptimo) se alcanza. La iteración comienza con un codebook que cumpla la condición de vecino más cercano hallado con alguna de las técnicas ya vistas. Entonces se encuentra un

[5] La resolución r se define como $r = \log_2 N$

nuevo codebook (usando la condición del centroide) que sea optimo para la partición dada, luego encuentra una nueva partición para el nuevo codebook (usando la condición de vecino más cercano). Este nuevo cuantizador vectorial tiene una distorsión menor o igual al original. La aplicación repetitiva de este paso proporciona un algoritmo que reduce o no cambia la distorsión en cada paso. Si bien cada uno de los pasos es óptimo y directo, nada nos asegura que se hallará el cuantizador vectorial óptimo (el de menor distorsión para el conjunto de todas la posibles inicializaciones), ni siquiera uno que cumpla ambas condiciones a la vez.

1. m = 1; Se inicializa el *codebook* C_m
2. Dado C_m se halla la partición optima mediante el vecino más cercano;
   Se resuelven los casos de igualdad.
3. Dada la partición se halla el *codebook* óptimo C_{m+1} mediante la condición de centroide.
   Se resuelven las celdas vacías generadas en el paso 2.
4. Se calcula la distorsión media para C_{m+1}, si el cambio desde la última iteración es menor que cierto umbral FIN, de lo contrario m = m + 1, y se vuelve al paso 2.

Muchos criterios de parada pueden ser usados. Uno de los más comunes es chequear si $1 - \frac{D_{m+1}}{D_m}$ es menor que cierto umbral adecuado.
Si el umbral se fija como cero, se tiene una sucesión de codebook cuyos valores de distorsión asociados son no crecientes. Si el algoritmo converge a un codebook en el sentido de que sucesivas iteraciones no producen cambios en él, entonces éste debe satisfacer las dos primeras condiciones necesarias de optimalidad.

Para un conjunto de entrenamiento finito, el Algoritmo de Lloyd converge en un número finito de pasos. Esto es fácil de ver dado que hay sólo un número finito de particiones, y la distorsión nunca crece, por lo cual, el algoritmo no puede volver a una partición que entregue un valor mayor de distorsión. De ahí que, la distorsión promedio asociada a la sucesión de cuantizadores vectoriales producidos por el Algoritmo de Lloyd debe converger en un número finito de pasos.

2.4.2 Dynamic Time Warping [15]

Los sistemas de reconocimiento de locutores basados en técnicas de ***DTW*** han sido los primeros que han alcanzado un nivel de fiabilidad suficientemente alto como para dar lugar al desarrollo de productos comerciales. Los sistemas de reconocimiento basados en DTW funcionan de la siguiente manera: primero se parametriza la señal de voz a reconocer; para ello se divide en pequeñas ventanas de análisis (unos 20 mseg), y sobre cada una de esas ventanas se realiza un proceso de análisis que extrae un conjunto de parámetros (que pueden ser acústicos o coeficientes espectrales). Ese conjunto o vector de parámetros se puede ver como un punto en un espacio n - dimensional. El conjunto de todas las ventanas de análisis se convertirá así en una secuencia de puntos en ese espacio, y esa secuencia de puntos es lo que se llama "patrón".

El sistema de reconocimiento dispone de un conjunto de patrones de "referencia" que se habrán calculado en la fase de entrenamiento, y que representan al conjunto de locutores que el sistema puede reconocer. De esta forma, una vez obtenido el "patrón", la tarea del sistema de reconocimiento consiste en compararlo con todos los patrones de referencia

que el sistema tiene, calculando la "distancia" que lo separa de las referencias, y elegir como locutor reconocido aquel cuyo patrón de referencia de la menor distancia en la comparación.

Si bien DTW realiza la clasificación basado en la medición de distancias en el espacio de características de manera similar a VQ, utiliza el hecho de que durante el entrenamiento la frase dicha es la misma que en el test. DTW compara la secuencia de vectores de testeo con la secuencia de entrenamiento, tomando en cuenta que ambas frases o palabras nunca son idénticas ya que los fonemas pueden pronunciarse de manera más larga o corta. Para esto, se encuentra una alineación temporal de las secuencias de entrenamiento y testeo la cual es óptima en el sentido de que no hay otra que de una distancia global menor y que satisfaga ciertas restricciones.

2.4.3 Hidden Markov Models [15]

Otro enfoque alternativo al de medir distancias entre patrones (enfoque topográfico) es el de adoptar un modelo estadístico (paramétrico) para cada uno de los locutores a reconocer, como son los modelos ocultos de Markov..

Estos sistemas han sido posteriores en el tiempo, y hoy en día la mayoría de los sistemas de reconocimiento en funcionamiento se basan en esta técnica estadística, ya que aunque sus prestaciones son similares a las de los sistemas basados en DTW, requieren menos memoria física y ofrecen un mejor tiempo de respuesta. Tienen como contrapartida una fase de entrenamiento mucho más lenta y costosa, pero como esta tarea se realiza una única vez, no debería ser mayor problema.

Un HMM se puede ver como una máquina de estados finitos en que el siguiente estado depende únicamente del estado actual, y asociado a cada transición entre estados se produce un vector de observaciones o parámetros (correspondiente a un punto del espacio n - dimensional). Se puede así decir que un modelo de Markov lleva asociados dos procesos: uno oculto (no observable directamente) correspondiente a las transiciones entre estados, y otro observable (y directamente relacionado con el primero), cuyas realizaciones son los vectores de parámetros que se producen desde cada estado y que forman el modelo a reconocer.

La clasificación usando HMM puede pensarse como una combinación de GMM y DTW. Un HMM consiste en varios estados, cada uno de los cuales modela una parte especifica de la señal de entrada. La distribución en el espacio de características que corresponde a un estado particular se modela estadísticamente por ejemplo, mediante una GMM. Una desventaja de este método es la dificultad que se presenta en hallar una estimación robusta cuando se tienen pocos datos de entrenamiento. Debido a la gran cantidad de parámetros que se deben estimar, se puede llegar a un modelo inexacto.

2.4.4 Gaussian Mixture Models[9,11]

Para implementar el test de hipótesis mencionado en **2.2** se debe elegir la función de probabilidad $f(\lambda / x)$ que de aquí en más llamaremos $P(\lambda / x)$. Para las aplicaciones *dependientes del texto* HMM tiene una buena performance pero para aplicaciones independientes del texto GMM ha probado ser el más exitoso[13].

GMM es ampliamente utilizado para el modelado de la voz a partir de los vectores de características (en nuestro caso ***MFCC***) adquiridos de cada locutor. Una vez obtenida cierta cantidad de estos vectores por cada locutor, se crea un modelo probabilístico que lo representa de forma singular.

Dado un vector de características: $\vec{x}$, la mezcla de densidades Gaussianas esta dada por:

$$P(\vec{x}/\lambda) = \sum_{i=1}^{M} w_i b_i(\vec{x})$$

que no es más que la combinación lineal ponderada de M densidades Gaussianas b_i, y que representa la probabilidad de observar un determinado vector de características $\vec{x}$ de cierto locutor λ, en donde:

- $\vec{x}$ es el vector de dimensión D a observar.
- w_i son los pesos de cada componente Gaussiana y cumplen $\sum_{i=1}^{M} w_i = 1$.
- $b_i(\vec{x})$ son las densidades Gaussianas D – dimensionales, cada una con la forma:

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}} e^{-(1/2)(\vec{x}-\vec{\mu}_i)^t \Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)} \quad \text{con i : 1...M}$$

  con $\vec{\mu}_i$ y Σ_i los vectores de medias y matrices de covarianza respectivamente.
- M es el orden del modelo o número de Gaussianas que tiene el modelo.

De esta forma cada locutor será representado por un modelo de muestras Gaussianas λ cuyos parámetros son: $\{ w_i, \mu_i, \Sigma_i \}$ con i =1...M

Se debe ser precavido a la hora de elegir la cantidad M de mezclas Gaussianas con la cual se va a trabajar. Elegir un numero elevado puede provocar que el modelo hallado sobre ajuste demasiado a los datos extraídos (*overfitting*). Por otro lado elegir un M pequeño puede llevar a que el modelo no sea lo suficientemente diferente a los demás modelos y no se pueda reconocer adecuadamente al locutor en cuestión. Experimentalmente se encuentra que generalmente dieciséis Gaussianas es un número apropiado[14].
El modelo general consta de matrices de covarianza Σ_i completas, pero lo más común en la bibliografía consultada es emplear modelos en los cuales las matrices de covarianza son diagonales. Esto reduce el número de parámetros que deben ser optimizados y además simplifica enormemente los cálculos a realizar. Sin embargo, esta limitación sobre las matrices de covarianza reduce las capacidades de modelado e incluso puede que se necesite incrementar el número de componentes empleadas.

2.4.4.1 Proceso de entrenamiento.

A partir de una colección de vectores X = $\{ \vec{x}_1, ..., \vec{x}_T \}$ de entrenamiento de una persona, se estiman los parámetros del modelo usando el algoritmo EM (estimación - maximización) descrito en el apéndice A-2.

Partiendo de un modelo inicial, el algoritmo EM refina iterativamente el modelo GMM incrementando de manera monótona su verosimilitud. Esto es, en la k - ésima iteración se encuentra el modelo $\lambda^{(k)}$ y se cumple: $P\left(X/\lambda^{(k)}\right) > P\left(X/\lambda^{(k-1)}\right)$. Este es el nuevo modelo inicial para repetir el proceso hasta llegar a un nivel de convergencia predeterminado.

En general el conjunto de vectores de características es muy grande, y por tanto, los valores de *P*(...) son a menudo muy chicos. Por esta razón es común calcular el logaritmo de la verosimilitudes que viene dado por:

$$LogP(X/\lambda) = \frac{1}{T}\sum_{t=1}^{T} \log P(\vec{x}_t/\lambda)$$

A este valor lo llamaremos Logl (Log - Likelihood) y es la medida que nos dice que tan probable es que los vectores X pertenezcan al modelo λ

La condición para detener la iteración puede ser: $LogP(X/\lambda^{(k)}) - LogP(X/\lambda^{(k-1)}) < \varepsilon$, o se puede imponer un número máximo de iteraciones.

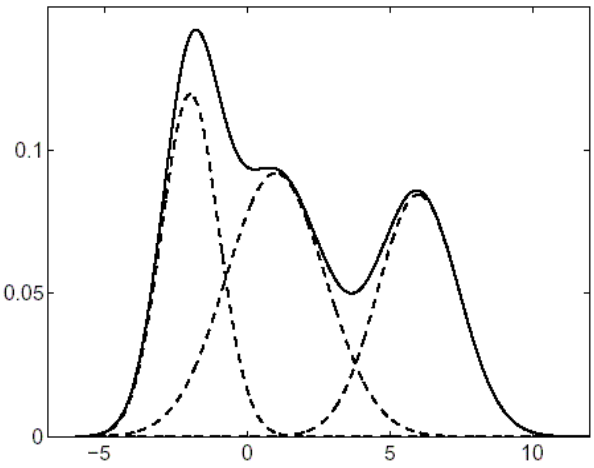


Figura 2.6. Modelo GMM (línea sólida) con 3 mezclas Gaussianas (línea punteada).

2.4.4.2 Proceso de testeo.

El logl es comúnmente usado para medir que tan bien un modelo se ajusta a los datos experimentales. Veamos a continuación como es que se lleva a cabo esta medida.

La *Identificación* de locutores se asocia con un problema de cercanía, de que modelo se acerca más a los datos de entrada El sistema supone que los vectores $\vec{x}$ de entrada pertenecen a un locutor que ya tiene su modelo correspondiente λ creado en la base de datos. Simplemente se deben evaluar los logl para cada modelo, aquel con mayor argumento es el que tiene mayor probabilidad de que los vectores de entrada pertenezcan a ese modelo. Cabe acotar que el logl con mayor argumento será el de $P(x/\lambda)$ más cercana a uno.

En *Verificación*, en cambio, se debe decidir si los vectores $\vec{x}$ de entrada de un locutor desconocido, pertenecen o no a un locutor buscado, cuyo modelo llamaremos: λ_i. Aquí la decisión se debe tomar sin tener en cuenta los otros modelos existentes en la base de datos. Para esto se crea el modelo universal o mundial[6] (***UBM***) λ_M a partir de voces de diferentes personas que pueden o no estar incluidas en la base de datos.
El sistema acepta o se rechaza la hipótesis de que los vectores $\vec{x}$ son de la persona en cuestión. Para eso se deben evaluar los logl para el modelo de la persona buscada λ_i y para el modelo mundial: λ_M, compararlos y decidir de acuerdo a:

$$\Lambda(X) = LogP(X/\lambda_i) - LogP(X/\lambda_M) \quad si \quad \begin{cases} \Lambda(X) > \Theta & \Rightarrow \quad aceptacion \\ \Lambda(X) > \Theta & \Rightarrow \quad rechazo \end{cases}$$

Si Λ(X) es mayor que cero, significa que X tiene más probabilidad de pertenecer al modelo del locutor buscado que al modelo mundial, lo contrario sucede si Λ(X) es menor que cero. A partir de esta deducción se podría fijar el umbral Θ en cero, pero experimentalmente se comprueba que esta no es una conclusión del todo acertada. Si bien el umbral es muy cercano a cero no es exactamente cero, más aún, puede ser negativo.

[6] La definición de Modelo Universal se vió en la sección 1.2.

Capítulo 3

MPEG Audio[1,2,4]

3.1 Introducción.

MPEG Audio, es un estándar genérico de compresión que a diferencia de los codificadores basados en el modelo de tracto vocal, realiza su compresión sin hacer suposiciones sobre la naturaleza de la fuente de audio. El codificador explota las limitaciones del sistema auditivo humano. Es decir, gran parte de la compresión está basada en la remoción de aquellas partes de la señal de audio que nos son imperceptibles. Debido a esto, la señal comprimida no es igual a la original, sin embargo, el oído no es capaz de percibir estas diferencias. Por esto decimos que se trata de una compresión con perdidas.

MPEG Audio ofrece tres opciones independientes de compresión, brindando una amplia gama de posibilidades para combinar complejidad y calidad de compresión.

La capa 1 (Layer 1) es la más simple y la más adecuada para bitrates por encima de 128 kBits/s por canal. Incluye la división del mapeado básico de la señal de audio digital en 32 subbandas, segmentación para el formato de los datos, modelo psicoacústico y cuantización fija siendo el retraso mínimo teórico de 19ms.

La capa 2 (Layer 2) tiene cierta complejidad y está destinada para bitrates entorno a los 128 kBits/s por canal. Incluye codificación lateral, factores de escala y diferentes composiciones de trama, siendo el retraso mínimo teórico de 35ms.

La capa 3 (Layer 3) es por lejos la más compleja pero ofrece la mejor calidad de audio, en particular para bitrates del orden de los 64 kBits/s por canal. Incluye un incremento de la resolución en frecuencia, basado en el uso de un banco de filtros híbrido. Permite cuantización no uniforme, segmentación adaptativa y codificación entrópica de los valores cuantizados mediante la codificación Huffman, siendo el retraso mínimo teórico de 59ms.

3.2 Codificador y decodificador

La figura 3.1. muestra el codificador al nivel de diagramas de bloques para las tres capas.

La señal a la entrada pasa a por un banco de filtros que la divide en múltiples bandas de frecuencia "Mapeo Tiempo - Frecuencia".
A su vez la señal también pasa por un bloque representativo del modelo psicoacústico que determina la relación entre la energía de la señal y el umbral de enmascaramiento para cada subbanda "Modelo Psicoacústico". De esta forma se calcula el nivel a partir del cual el ruido comienza a ser perceptible para cada banda.

En el bloque "Cuantización y Codificación" se realiza la asignación del numero de bits por subbanda. Para ello se examina tanto las muestras de salida del banco de filtros como la SMR[7] proporcionada por el modelo psicoacústico.

Por último, el bloque "Empaquetado de la Trama" se encarga de agrupar todos los datos, en una estructura llamada ***trama***, pudiendo añadir datos auxiliares, como por ejemplo un Código de Redundancia Cíclica (CRC), o información del usuario.

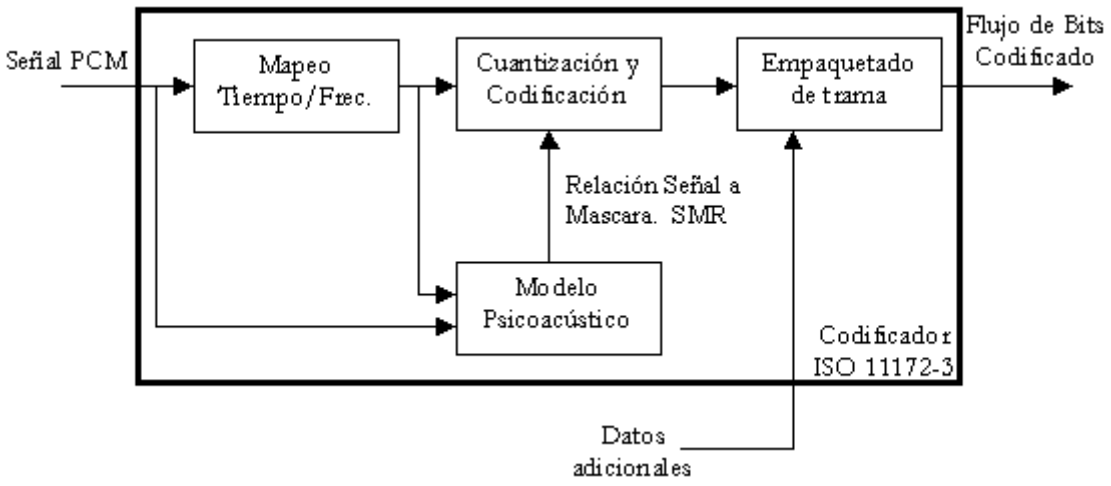


Figura 3.1. Diagrama de un codificador ISO 11172 - 3

En el decodificador de la figura 3.2. los datos de la trama son desempaquetados y decodificados para recuperar las diversas partes de la información. El bloque de "Reconstrucción" recompone la versión cuantizada de la serie de muestras mapeadas. El "Mapeo Frecuencia - Tiempo" transforma estas muestras de nuevo a PCM.

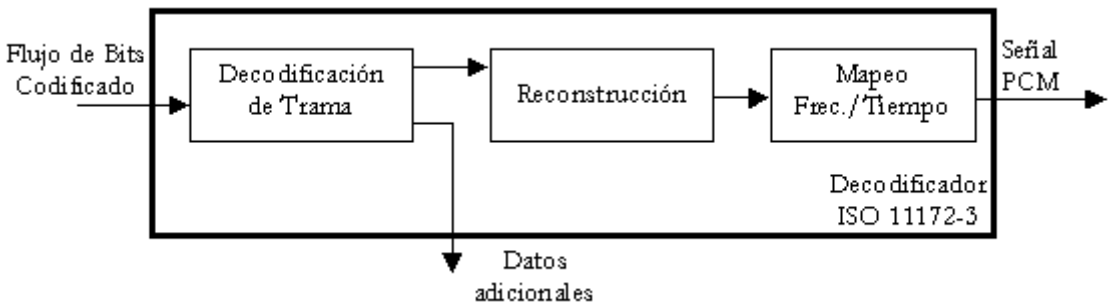


Figura 3.2. Diagrama de un decodificador ISO 11172 - 3

[7] Ver apéndice A-3.

3.3 MPEG-1 capa 3

En este apartado veremos las características más importantes de la capa 3 de mpeg - 1 que es la que nos interesa. Desarrollos similares sobre las capas 1 y 2 pueden encontrarse por ejemplo en[1].

3.3.1 Introducción. [1,2,4]

La capa 3 es sustancialmente más compleja que las dos anteriores. El mapeo tiempo – frecuencia está basado en el mismo banco de filtros que las capas 1 y 2 pero compensa algunas de las deficiencias que aquel banco presenta, procesando las salidas filtradas con una Trasformada del Coseno Discreta Modificada (*MDCT*). Estos dos bloques conforman el denominado filtro híbrido, el cual proporciona una resolución en frecuencia variable, 6 x 32 o 18 x 32 subbandas, ajustándose mucho mejor a las bandas críticas de las diferentes frecuencias.

A diferencia del banco de filtros polifásicos, sin cuantización, la transformación MDCT no presenta perdidas. La MDCT también subdivide las salidas de subbanda en frecuencia para incrementar la resolución y así poder dividir el audio en bandas que se ajustan mejor a las bandas críticas del oído (las subbandas no están equi - espaciadas). Lo que es más, una vez que los componentes de subbanda son subdivididos en frecuencia, el codificador de la capa 3 cancela parte del aliasing causado por el banco de filtros polifásicos.
El decodificador de la capa 3 debe deshacer la cancelación de aliasing para que la MDCT inversa pueda reconstruir las muestras de subbandas en su forma original.

Además del procesado con la MDCT la capa 3 emplea un modelo psicoacústico que incluye los efectos totales del enmascaramiento tanto en frecuencia como en el tiempo[8]. También utiliza un sofisticado esquema de codificación entrópica y cuantización no uniforme donde se involucran la redundancia estéreo[4] y codificación Huffman de longitud variable, un método de codificación entrópica sin pérdida de información.
La gran diferencia con las otras dos capas es que la variable controlada es el ruido, a través de bucles iterativos que lo reducen al mínimo posible en cada paso.

La definición de trama según ISO varía respecto de la de las capas anteriores[9]. Las tramas contienen información de 1152 muestras y empiezan con la misma cabecera de sincronización y diferenciación, pero la información perteneciente a una misma trama no se encuentra generalmente entre dos cabeceras. El empaquetado de trama incluye el uso de una reserva de bits[10] (*bit reservoir*), que hace posible emplear más bits en partes de la señal que lo necesiten. La longitud de la trama puede variarse en caso de necesidad.
Además, permite alta calidad en el audio a tasas tan bajas como 64 Kbps.

En la figura 3.3 se muestra un diagrama de bloques detallado del codificador que se usa en la Capa 3.

[8] Ver apéndice A-3.4
[9] Para las capas 1 y 2 una trama consiste en los datos a codificar, un encabezado, codigos CRC y posibles datos auxiliares.
[10] Ver apéndice A-4.4.3.

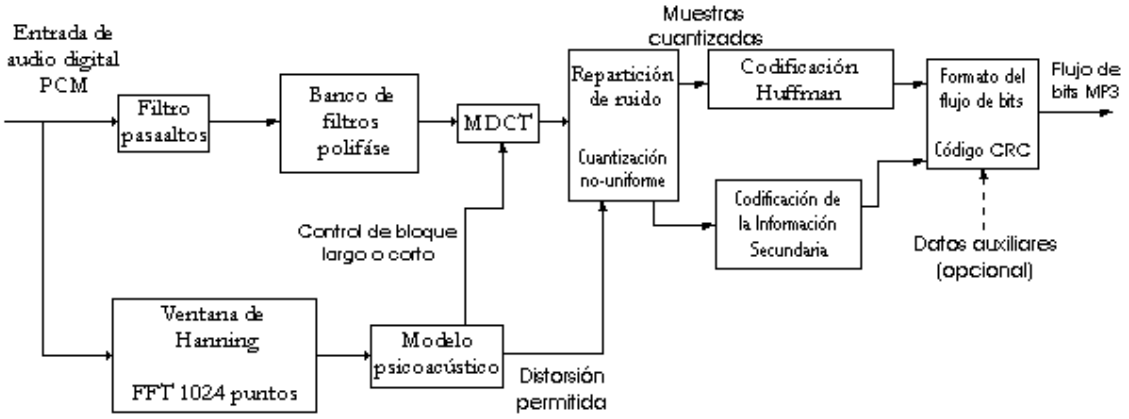


Figura 3.3 Diagrama de bloques de un codificador MP3

En forma resumida el proceso de codificación es el siguiente: el flujo de audio a la entrada pasa a través de un banco de filtros que divide la señal en múltiples subbandas. En forma paralela al filtrado se realiza el análisis psicoacústico que determina el ruido (distorsión permitida) en cada subbanda. La etapa "Repartición de ruido" usa las distorsiones permitidas para decidir cómo dividir el número total de bits de código disponibles. Por último, las muestras codificadas mediante Huffman junto con la información secundaria son convertidas en un flujo de bits MP3 válido.

3.3.2. Análisis psicoacústico[4]

El estándar 11172-3 (que describe la manera de generar flujos de audio MP3 válidos), proporciona dos modelos psicoacústicos; el modelo psicoacústico I es menos complejo que el modelo psicoacústico II y simplifica mucho los cálculos. Ambos modelos trabajan para cualquiera de las capas, aunque requieren adaptaciones específicas para el esquema de la capa 3. Existe considerable libertad en la implementación del modelo psicoacústico; la precisión que se requiera del modelo es dependiente de la aplicación y de la tasa de bits que se quiere lograr. Para bajos niveles de compresión, donde hay un número generoso de bits para realizar la codificación, el modelo psicoacústico puede ser completamente omitido, en cuyo caso, sólo se calcula la SNR[11] más baja, y con este valor se realiza el proceso de repartición de ruido para la subbanda.

El modelo psicoacústico II que se usa en la Capa 3 tiene mejoras adicionales que se adaptan mejor a las propiedades del oído humano, en comparación con el modelo empleado en las otras dos capas (modelo I).

Primero el modelo convierte el audio al dominio espectral, usando una FFT de 1024 puntos para conseguir una buena resolución de frecuencia y poder calcular correctamente los umbrales de enmascaramiento. Antes de la FFT, se aplica una ventana de Hanning convencional para evitar las discontinuidades en los extremos de la señal. 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

[11] Relación señal – ruido.

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 "Repartición de ruido" en el codificador para uso posterior. Veamos a continuación algunas de las funciones del modelo psicoacústico para la capa tres.

1) Alineación en tiempo. Se debe tener en cuenta que cuando se hace la evaluación psicoacústica, los datos de audio que son enviados al modelo deben ser concurrentes con los datos de audio a ser codificados. El modelo psicoacústico debe tener en cuenta el retardo de los datos al pasar por el banco de filtros y aplicar un desplazamiento adicional, de tal manera que los datos relevantes queden centrados en la ventana del análisis psicoacústico. Por ejemplo, usando el modelo I para la Capa 1, el retardo a través del banco de filtros es 256 muestras y el desplazamiento necesario para centrar las 384 muestras, dentro de la FFT de 512 puntos, es: (512 - 384) / 2 = 64 puntos
El desplazamiento requerido es, entonces, de 320 puntos para alinear los datos del modelo I con la salida del banco de filtros polifásico.

2) Representación espectral. El modelo psicoacústico realiza una conversión del tiempo a la frecuencia totalmente independiente del mapeo realizado por el banco de filtros porque necesita una mejor resolución en frecuencia para calcular con gran precisión los umbrales de enmascaramiento. Ambos modelos usan una transformada de Fourier para realizar el mapeo.

El modelo I usa una FFT de 512 puntos para la Capa 1 y una FFT de 1024 puntos para las Capas 2 y 3. Debido a que el análisis se realiza para 384 muestras en la Capa 1, la FFT de 512 puntos proporciona la cobertura adecuada. El análisis psicoacústico para las Capas 2 y 3 se realiza sobre 1152 muestras, así que la FFT de 1024 puntos no proporciona cobertura total. Idealmente, la FFT debería cubrir todas las 1152 muestras; aunque 1024 puntos es un compromiso razonable ya que las muestras que se omiten, no tienen mayor impacto en el análisis psicoacústico.

El modelo II usa una FFT de 1024 puntos para todas las capas. En la Capa 1, el modelo centra las 384 muestras dentro de la FFT de 1024 puntos. Para las Capas 2 y 3, el modelo ejecuta dos cálculos psicoacústicos de 1024 puntos. El primer cálculo se encarga de las 576 muestras iniciales, y el segundo cálculo se realiza sobre las últimas 576 muestras. El modelo II combina los resultados de ambos cálculos, de tal manera que el resultado total implique la selección del umbral de enmascaramiento de ruido (*Noise Masking Treshold*) más bajo en cada subbanda. Para simplificar los cálculos, ambos modelos procesan los valores espectrales en unidades perceptuales, el Bark.

3) Componentes tonales y no tonales. Ambos modelos identifican y separan las componentes tonales y las componentes de ruido en la señal de audio. Esto se debe a que cada componente presenta un tipo de enmascaramiento diferente.
El modelo I identifica las componentes tonales, basado en los picos locales del espectro de potencias. Después de procesar todas las componentes tonales, el modelo concentra los valores espectrales restantes en una única componente no tonal por banda crítica. El índice de frecuencia de cada una de estas componentes no tonales es el valor más cercano a la media geométrica de la banda crítica a la cual pertenece cada componente no tonal.
El modelo II realmente nunca separa las componentes tonales ni las no tonales, sino que calcula un índice de tonalidad en función de la frecuencia.

4) Estimación del índice de tonalidad[8]. Este índice mide el comportamiento que presenta cada tipo de componente. El modelo II usa este índice para interpolar entre valores puros de TMN y valores puros de NMT[12]. El índice de tonalidad se basa en una predicción mediante una extrapolación lineal de los últimos dos cálculos, para predecir los valores de la componente que está siendo procesada. Las componentes tonales son más predecibles y, por lo tanto, tienen índices de tonalidad más altos. Este método de discriminación es mejor que el usado por el modelo I.

5) Función de dispersión. La capacidad de enmascarar de una componente determinada se distribuye por toda la banda crítica que la rodea. Ambos modelos determinan el umbral de enmascaramiento de ruido para ambos tipos de componentes; para lograr esto, el modelo I compara con un enmascaramiento determinado empíricamente, mientras que el modelo II aplica la función de dispersión descripta en la ecuación A3.7. En las aplicaciones de la capa 3, solo se toman en cuenta aquellos valores de la función de dispersión mayores a 60 dB.

6) Umbral de enmascaramiento individual. Para poder calcular el umbral de enmascaramiento global (paso 6), el modelo I debe calcular primero los umbrales de enmascaramiento que cada componente tonal o no tonal genera sobre la señal de audio (llamados "Umbrales de enmascaramiento individuales"). Debe tenerse en cuenta que antes de esto se realiza un proceso conocido como "Decimation of maskers" (disminución en la cantidad de componentes enmascarantes). Este proceso consiste en escoger únicamente las componentes tonales y no - tonales que verdaderamente enmascaran el sonido (cuya magnitud y distancia en Barks debe ser apropiada), desechando el resto de componentes computadas en el paso anterior.
Después de realizada esta elección, el modelo I calcula el efecto de enmascaramiento que cada componente enmascaradora (tonal o no - tonal) tiene sobre las líneas de frecuencia adyacentes a ella. Este análisis sólo es necesario hacerlo para las líneas de frecuencia que se encuentran entre -3 y +8 Barks a partir de la componente enmascaradora.
O sea, el análisis abarca todas las líneas de frecuencia que se encuentren tres bandas críticas a la izquierda (hacia las bajas frecuencias), y ocho bandas críticas a la derecha (hacia las altas frecuencias) de la componente enmascaradora. Esto se debe a que el efecto de enmascaramiento de la componente tonal o no - tonal que está siendo analizada (por más intensidad que ésta tenga) es demasiado tenue por fuera de este rango.
Como el modelo II nunca separa las componentes no - tonales y tonales, sino que calcula el índice de tonalidad (en función de la frecuencia) que presenta cada componente enmascaradora, entonces no es necesario hacer el cálculo de los umbrales de enmascaramiento individuales.

7) Umbral de enmascaramiento global. Ambos modelos psicoacústicos incluyen un umbral de enmascaramiento absoluto, el cual ha sido determinado empíricamente: el mínimo umbral auditivo en un ambiente silencioso. Se debe recordar que éste es la intensidad del sonido más débil que se puede escuchar cuando no hay más sonidos presentes.
Usando el modelo I, este umbral absoluto se combina con los umbrales individuales calculados en el paso anterior para determinar el umbral de enmascaramiento global sobre toda la banda de audio.

[12] Ver apéndice A – 3.4.

El modelo II no calcula el umbral de enmascaramiento global, sino que trabaja todos los datos dentro de cada subbanda, de acuerdo con el índice de tonalidad que tenga cada componente enmascaradora en esa subbanda.

8) Pre - Eco[3,8]. Los efectos de pre - ecos son muy comunes cuando se trabaja con esquemas perceptuales de codificación de audio que usan alta resolución en frecuencia.
Para entender el origen de los pre - ecos, consideremos el diagrama simplificado del decodificador de un sistema de codificación perceptual de la figura 3.2.

Las líneas de frecuencia reconstruidas son combinadas por el filtro síntesis, que consiste en una matriz de modulación y una ventana de síntesis. El error de cuantización introducido por el codificador puede verse como una señal agregada a las líneas de frecuencia originales, con un largo en el tiempo que es igual al largo de la ventana de síntesis. Por esto, los errores de la reconstrucción se esparcen por todo el largo de la ventana. Si la señal de audio presenta un incremento abrupto de energía, el error de cuantización también se incrementa.

Si ese pico de energía ocurre dentro de la ventana de síntesis, el error se esparcirá dentro de la ventana de síntesis completa, precediendo en el tiempo la causa real de su existencia. Si dicha señal pre - ruido se extiende más allá del período de pre - enmascaramiento del oído humano, se vuelve audible y se llama pre - eco.
La capa tres incorpora varios pasos para reducir el pre - eco. Primero, el modelo psicoacústico de la capa tres contiene modificaciones que detectan las condiciones de pre - eco. Segundo, la capa tres puede pedir prestados *codebits* de la reserva de bits para reducir el ruido de cuantización cuando las condiciones de pre - eco se presentan. Por último el codificador puede cambiar a un tamaño de bloque MDCT más pequeño para reducir el tiempo de ventana efectivo.

9) Umbral de enmascaramiento mínimo. Ambos modelos psicoacústicos seleccionan el mínimo umbral de enmascaramiento en cada subbanda.

Con el modelo I, para encontrar el umbral de enmascaramiento mínimo en cada subbanda, simplemente se extrae el mínimo valor del espectro global incluido entre las dos frecuencias límites de cada subbanda, o sea, el valor extraído del umbral global debe ser el valor mínimo de enmascaramiento en la subbanda. Este método se comporta bien para las subbandas más bajas donde la subbanda es estrecha con respecto a las bandas críticas, pero se vuelve inadecuado para las subbandas altas porque una banda crítica en esta frecuencia se distribuye sobre varias subbandas. Esta imprecisión se incrementa todavía más, debido a que el modelo I concentra todas las componentes no tonales, dentro de cada banda crítica, en un único valor para una sola frecuencia.

El modelo II selecciona el mínimo de todos los umbrales de enmascaramiento en cada subbanda sólo para regiones de frecuencia donde el ancho de la subbanda es amplio comparado con el ancho de la banda crítica. Si el ancho de la subbanda es estrecho en comparación con el ancho de la banda crítica, el modelo realiza un promedio entre todos los umbrales de enmascaramiento en esa subbanda. El modelo II es más preciso para las subbandas altas, ya que éste no concentra las componentes de ruido.

10) Relaciones señal a máscara. Los dos modelos computan la relación señal a máscara, SMR, como la relación entre la energía de la señal en la subbanda (para la Capa 3, un grupo de bandas) y el mínimo umbral de enmascaramiento para esa subbanda. El modelo

psicoacústico pasa este valor a la sección "Repartición de ruido" (para las Capas 1 y 2, "Repartición de bits") para uso posterior. En la Capa 3, el valor que se entrega no es la SMR, sino un valor equivalente llamado "Distorsión permitida" o "Ruido permitido". Este valor determina cuál es la cantidad máxima de ruido de cuantización que se permite en el bloque "Repartición de ruido".

3.3.3. Banco de filtros híbridos conmutados. [1,2,4,5]

El banco de filtros usado en MPEG capa 3 pertenece a la clase de bancos de filtros híbridos. Estos son construidos poniendo en cascada dos bancos de filtros diferentes, primero un banco de filtros polifásico (igual que en las capas 1 y 2) y segundo, un banco de filtros MDCT.

3.3.3.1. Filtro pasaaltos.

El estándar ISO/IEC 11172-3 proporciona respuesta en frecuencia hasta el nivel de 0 Hz. Sin embargo, para ciertas aplicaciones, se puede incluir un filtro pasaaltos a la entrada del codificador, con su frecuencia de corte ubicada entre 2 y 10 Hz. La aplicación de tal filtro evita el innecesario requerimiento de una alta tasa de bits para la subbanda más baja y aumenta la calidad total en el sonido.

3.3.3.2. Banco de filtros polifásicos.

Como ya vimos, el oído tiene una limitada selectividad en frecuencia que varía desde menos de 100 Hz para las frecuencias más bajas hasta un poco más de 4 kHz. Para las frecuencias más altas. El ancho de banda que proporcionan los filtros polifásicos 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. Debido a esto, 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.

3.3.3.3. Transformada discreta del coseno modificada.

La Capa 3 subdivide cada una de las 32 bandas (salidas del banco de filtros) mediante una transformación DCT Modificada de seis o dieciocho puntos (líneas de frecuencia) y 50% de solapamiento, con el fin de compensar la falta de precisión del banco de filtros, logrando subdividir la salida espectral en frecuencias que proporcionen mejor resolución con respecto a las bandas críticas.
Usando dieciocho puntos, el número máximo de componentes en frecuencia es:
32 x 18 = 576, dando una resolución en frecuencia (ancho de banda) de: 24000/576 = 41,67 Hz. (f = 48 kHz.) en cada banda.
Usando 6 puntos, la resolución en frecuencia es menor, pero la temporal es mayor, y se aplica en aquellas zonas en las que cabe esperar efectos de pre - eco (transiciones bruscas de silencios a altos niveles energéticos, como por ejemplo justo antes de un sonido de percusión). En estos casos se produce un transitorio con elevados errores de cuantización, debido a la saturación del cuantizador.

Al realizar la decodificación, el error se distribuye por toda la trama, ocasionando que las partes de silencio ya no sean silencio, sino que presenten parte de la energía de las otras regiones de la trama. Esto obliga al uso de ventanas MDCT temporales más pequeñas que limitan el efecto de pre - eco a un número menor de muestras, en comparación con el uso de ventanas grandes logrando de esta manera, reducir la distorsión. El pre enmascaramiento temporal evita que la distorsión restante sea audible.

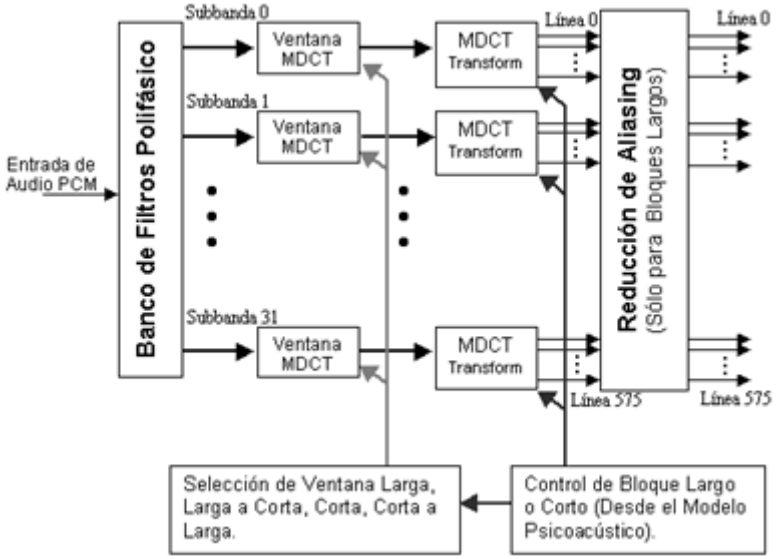


Figura 3.4 Diagrama de bloques de las operaciones de la MDCT

La Capa 3 tiene tres modos de bloque: dos modos donde las 32 salidas del banco de filtros pueden pasar a través de las ventanas y las transformadas MDCT, todas las salidas con la misma longitud de bloque. Y un modo de bloque mixto donde las dos bandas de frecuencia más baja usan bloques largos y las 30 bandas superiores usan bloques cortos. La decisión del modo de bloque a ser usado recae sobre el modelo psicoacústico: si la señal presenta muchos transitorios se debe usar bloque corto, correspondiente a tres ventanas cortas; pero si la señal es más estacionaria, se debe usar bloque largo, correspondiente a una ventana larga. El cambio entre modos no es instantáneo; un bloque largo con una ventana de datos especializada (ventana larga a corta o, ventana corta a larga) proporciona el mecanismo de transición entre modos. En la figura 3.5 se muestran los cuatro tipos de ventana que se usan durante el proceso MP3: (a) *NORMAL*, (b) transición de ventana larga a corta (*START*), (c) tres ventanas cortas (*SHORT*), y (d) transición de ventana corta a larga (*STOP*).

Si se ejecuta la MDCT sobre cualquiera de las ventanas largas (*NORMAL*, *START*, o *STOP*), se producirán 18 líneas de frecuencia debido al 50% de solapamiento. Cuando se usan las tres ventanas cortas se producirán 3 grupos, cada grupo con 6 líneas de frecuencia que pertenecen a diferentes intervalos de tiempo. El proceso de la transformación MDCT sobre cualquier tipo de bloque producirá, entonces, 576 líneas de frecuencia referidas como "Gránulo" (subdivisión de una trama).

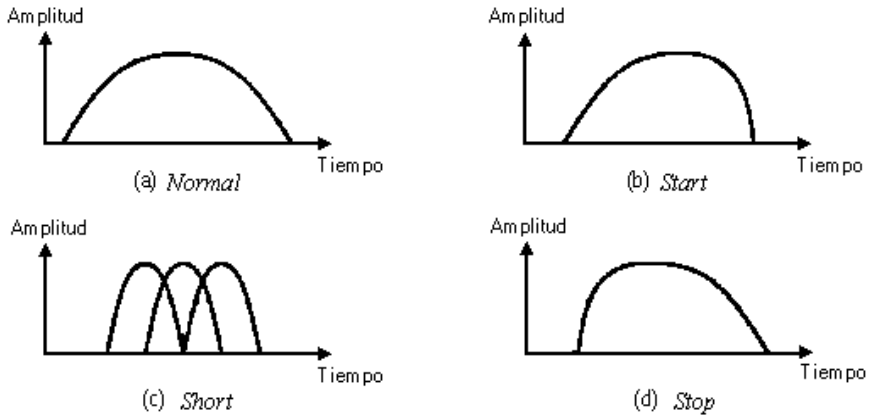


Figura 3.5 Ventanas de datos usadas durante el proceso MP3.

En resumen, el proceso que se ha mostrado es: 576 muestras PCM de entrada se convierten en 576 muestras subbanda. El solapamiento, antes de la MDCT, ocasiona que esta cantidad se duplique: en este punto son 1152 muestras subbanda, las cuales finalmente producen 576 coeficientes MDCT (líneas de frecuencia) de salida.

Antes de continuar, se realiza la reducción del aliasing introducido por el filtro análisis. Este proceso se realiza aquí, para obtener una reducción en la cantidad de información a ser codificada y transmitida.

Capítulo 4

Criterios de diseño

4.1 Elección de las técnicas y parámetros de diseño.

Como se adelantó en el punto **2.4** vamos a emplear dos técnicas de generación de modelos que son Cuantización Vectorial y Modelos de Mezclas Gaussianas. Estas técnicas se eligieron por ser las adecuadas para las aplicaciones independientes del texto que queremos desarrollar. Sin embargo llevamos a la práctica dos variaciones de las anteriores que, como se verá, brindan resultados comparables en lo que a performance del sistema refiere.

La primera modificación tiene como propósito acelerar la lenta convergencia del algoritmo EM. Para esto recurrimos a iniciar el algoritmo EM mediante VQ[13] y obtenemos un modelo de partida mucho mejor que cualquier modelo aleatorio de los que parte GMM. Logramos así muchos menos pasos de iteración y un modelo más ajustado a los datos. Denotaremos este modelo como HÍBRIDO.

Del proceso del cálculo de los coeficientes MP3CEP[14] es de suponer que estos están no correlacionados, sin embargo una rápida inspección de la matriz de correlación (figura 4.1) revela cierto grado de correlación entre los primeros coeficientes. Muy a pesar de esto, tal cual se explicó en el apartado **2.4.4**, el uso de GMM diagonal reduce considerablemente las operaciones y es comparable al GMM completo con un menor número de regiones, de manera que se considerará esta opción.

[13] VQ según se lo referencia en el punto 4.1.1
[14] El calculo de los MP3CEP se explica en el punto 4.2

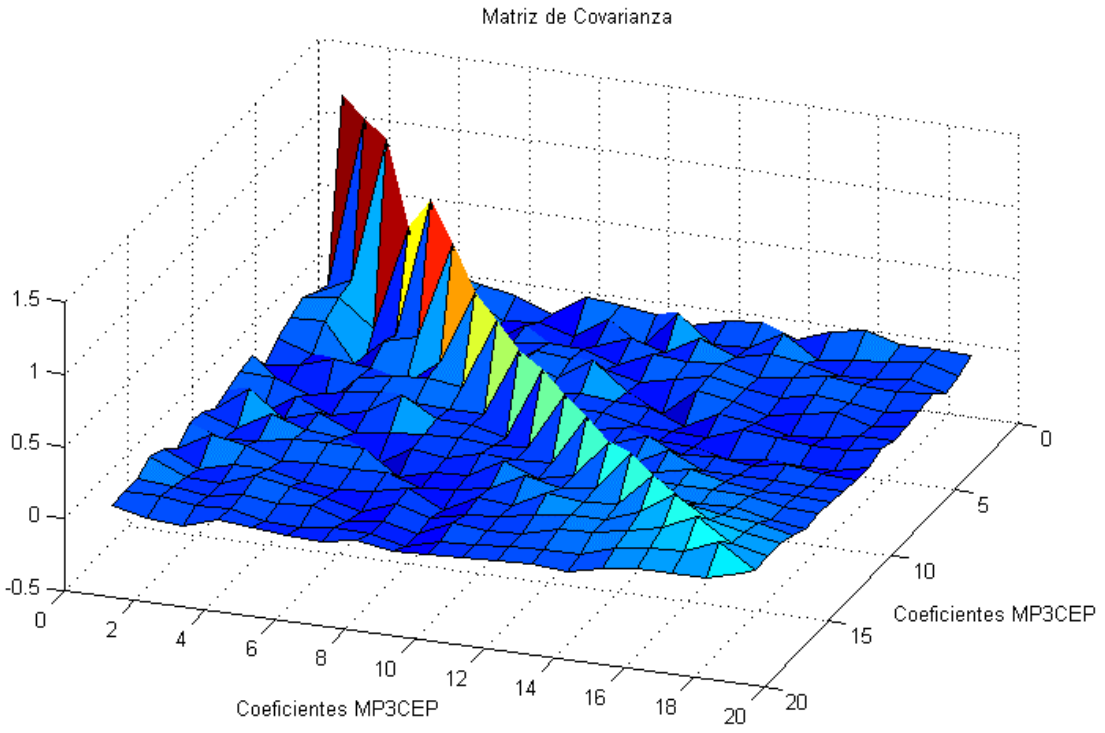


Figura 4.1

4.1.1 Modificaciones adicionales sobre el modelo VQ

A la hora de realizar los modelos, VQ nos da buenos resultados y de manera rápida pues para definir las regiones solo trabaja con su centroide. Con el fin de compatibilizar estos modelos con los hallados para GMM, tomaremos nota de la covarianza y los pesos de cada una de las regiones.

El peso para una región dada se estimó como el cociente entre los vectores de características que cayeron en esa región sobre el total de vectores. Por su parte tomamos como matriz de covarianza a la covarianza de los elementos que se encuentran en esa región. De esta manera queda definido el modelo de cada persona por los centroides, una matriz de covarianza para cada región y los pesos de estas.

Es de destacar que el algoritmo de Lloyd permanece intacto, es decir, la convergencia se alcanza exclusivamente por medio de los centroides. De esta forma estas modificaciones no insumen mayor tiempo pues se realizan al final el algoritmo de Lloyd y mejoran las bondades del modelo VQ original.

De ahora en más nos referiremos por modelo VQ, a aquel realizado mediante el algoritmo de Lloyd y que además brinda una estimación de la matriz de covarianza y el peso de cada región.

4.2 Extracción de características sobre MP3.

En toda la bibliografía consultada[17,18] se busca extraer los coeficientes MFCC a la salida del banco de filtros polifáse. El proceso para esto de manera abreviada es el siguiente: mirado desde el lado del decodificador, una vez desmembrada la trama, se deben decodificar las muestras con un decodificador Huffman y luego se deben reajustar los valores de escala. Una vez hecho esto, se tienen los 576 coeficientes MDCT sin aliasing (Figura 4.2). Continuando con la decodificación se debe pasar a través del bloque de aliasing y finalmente la IMDCT para alcanzar las 36 muestras por subbanda (salida del filtro polifáse).

Ubicados en este punto podemos disponer del largo de ventanas que deseemos, así también como del solapamiento entre estas. Dependiendo de la resolución deseada se pueden tomar las características en los valores de subbanda que se crean convenientes. Por ejemplo en **[18]** se toman 36 muestras por subbanda cada 18 para obtener ventanas de 26ms cada 13ms @ 44.1hHz.

En definitiva se pueden o bien tomar los 576 valores MDCT o bien se toman valores en subbanda. Dado nuestro objetivo escogimos los 576 valores MDCT.

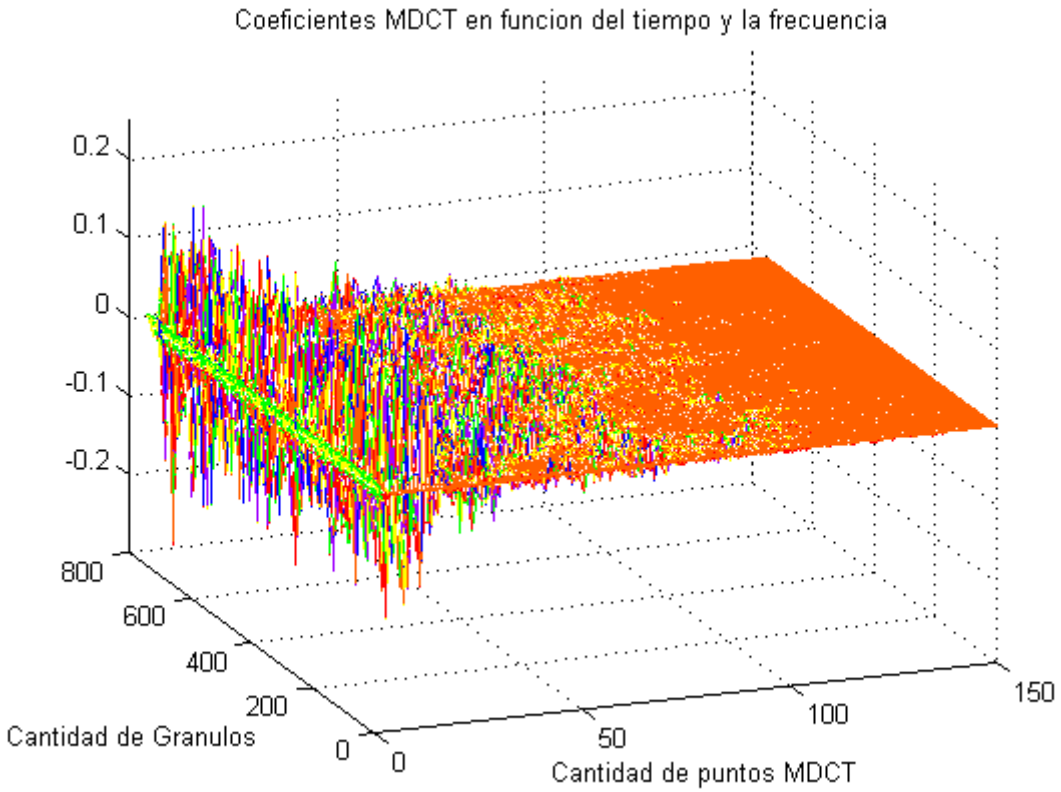


Figura 4.2 Se muestran 150 de los 576 coeficientes MDCT para los primeros 300 gránulos. El termino de continua de los coeficientes MDCT es una suerte de promedio de la señal en el tiempo cada 576 muestras.

Las ventajas de nuestra elección son un rápido acceso a los datos para calcular las características y además obtenemos coeficientes sin aliasing. Por otro lado no conocemos exactamente cual es el solapamiento ya que este varia en cada subbanda de acuerdo al tipo de ventana escogido por el modelo psicoacústico debido a los efectos de pre - eco.

El procedimiento utilizado para obtener los parámetros de características de la voz es el descrito en la sección **2.3.1.** con la salvedad que sustituiremos el espectro de la señal (FFT) por los coeficientes MDCT. Denominaremos a estos parámetros MP3CEP.

En la práctica se encontró para determinados archivos de entrenamiento, una distorsión en los coeficientes MDCT entre 50 y 60 Hz debido a ruido de la red, cables, etc. Para solventar este inconveniente se procedió a eliminar el aporte de la primer banda al momento del cálculo de los MP3CEP, en todos los archivos de entrenamiento.

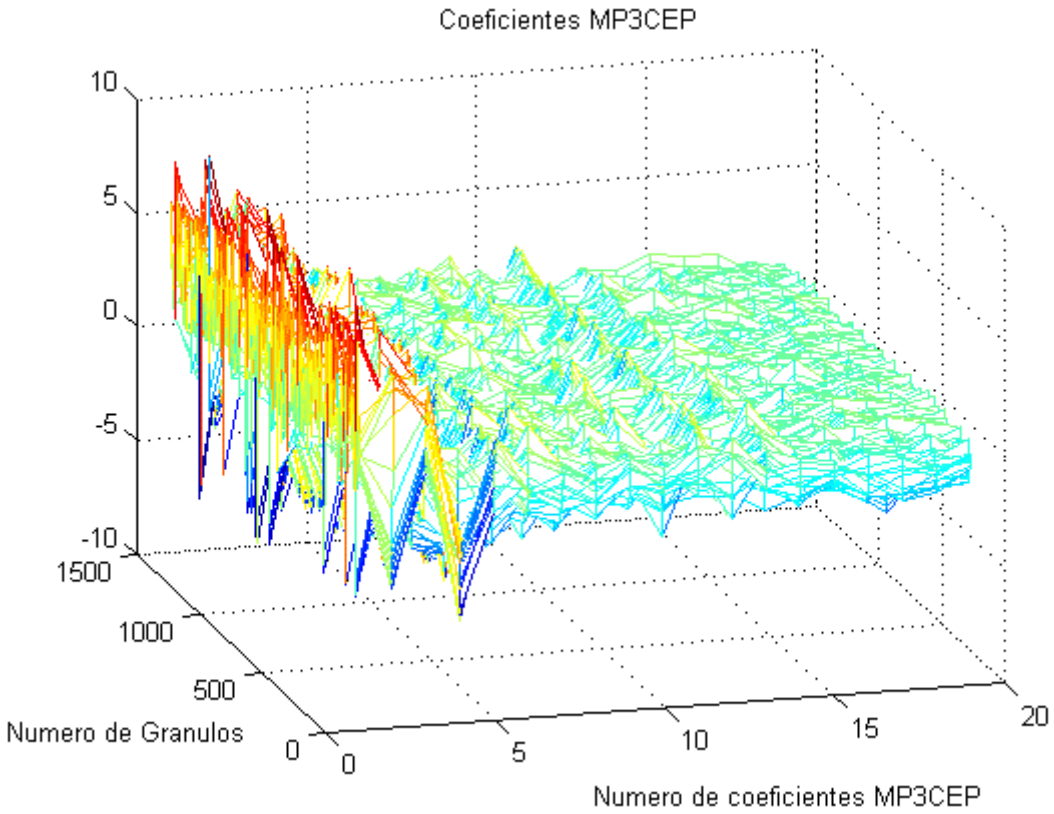


Figura 4.3. Se muestran los 19 coeficientes MP3CEP para 1500 gránulos. Se nota claramente la correlación de la señal de voz en el tiempo (numero de gránulos), mientras que se ven poco correlacionados los coeficientes MP3CEP como se había visto en la Fig 4.1

En resumen, trabajando a 44.1 kHz. mono se obtienen 76.6 vectores de características MP3CEP por segundo de 19 coeficientes cada uno. Estos vectores se obtienen de la salida de un banco de filtros solapados 110 Mels desde 71 Hz hasta 5.3KHz. En la Fig 4.3 se muestran los coeficientes MP3CEP.

4.3. Comparación de características

La comparación de características busca computar cierta medida de similitud entre un vector de características y un vector de algún modelo. Los modelos de los locutores son generados a partir de vectores de características extraídas como se explico en **4.2**.
Existen dos tipos de modelos, los modelos estocásticos y los modelos de vecindad. En el caso de los modelos estocásticos la comparación de características es probabilística y resulta

ser una medida de máxima verosimilitud o probabilidad condicional de las observaciones dado el modelo.

Para los modelos de vecindad la comparación de características resulta ser determinista. Se asume que las observaciones son una copia imperfecta de los componentes de una vecindad y se intenta modelar a estas observaciones minimizando alguna medida de distancia *d*.

El modelo de vecindad más simple para un set de n vectores de entrenamiento consiste en su centroide (μ), entonces el resultado de la comparación de características entre un vector de características x_i de un sujeto y el de algún modelo es: $d(\mu, x_i)$.
Se pueden definir muchos tipos de distancias entre μ y x_i, sin embargo haremos hincapié en dos de ellas, la distancia Mahalanobis y la Norma Dos.

Muchos tipos de distancia se pueden expresar como $d(x_i, \mu)= (x_i-\mu)^T W(x_i, \mu)$ siendo W una matriz de pesos. Si W es la identidad se tiene la Norma Dos y si es la inversa de la matriz de covarianza correspondiente a la vecindad de μ es la distancia Mahalanobis. La distancia Mahalanobis otorga menos peso a las componentes con mayor varianza y es equivalente a la Norma Dos sobre los vectores propios de la matriz de covarianza. El lugar geométrico de los puntos que distan la misma distancia Mahalanobis es una hiper elipse.

Para *Identificación*, tanto la distancia Mahalanobis como la Norma se usaron de la siguiente manera. Primero se computa cual es la región que dista menos al vector de características para cada modelo de locutor. Luego, al cabo de n vectores de características se decide quien es el locutor según quien haya acumulado la menor distancia.

4.4. Fases de entrenamiento y testeo de modelos.

Una vez decididos los métodos de generación de modelos y los coeficientes a usar para representar las características, se procedió a generar una base de datos. Para ello fueron grabados los noticieros de Telenoche 4 y Telemundo 12 (emitidos en frecuencia modulada) en 20 secciones durante un periodo de 4 meses, a fin de captar las variaciones de la voz. Fueron extraídos 34 locutores entre ellos reconocidos personajes de la política, periodistas y cronistas. Se generaron además tests de prueba y diseño para *Identificación* y *Verificación*.

4.4.1. Base de datos MP3 para locutores.

A modo de estudiar la variabilidad de la voz de los distintos locutores debido a diversos factores, se construyeron dos bases de datos de 34 usuarios cada una. En la primera de ellas (Base A), se entreno cada modelo con 30 segundos de audio tomados el mismo día bajo las mismas condiciones.

En la otra base de datos (base B), se entreno cada modelo con un promedio de 3 minutos por locutor, recogiendo la mayor variabilidad posible en las voces al incluir en las mismas muestras de diferentes días, estados de ánimo, condiciones de grabación, etc.

La base A, se traduce en regiones muy bien definidas con alta concentración de puntos, por el contrario en la base B habrá un esparcimiento mayor en las regiones debido a las variaciones de la voz tomadas.

Los modelos se van a generar mediante los cuatro métodos (GMM completo, GMM diagonal, VQ e HÍBRIDO) con 16 regiones cada uno y se hará una comparación detallada basándose en dos aspectos muy importantes: velocidad y performance.

4.4.2. Base de datos MP3 para el Modelo Mundial.

Este modelo tiene como objetivo caracterizar a un grupo de personas que serán representativas de todas las personas y sus características de voz. Por este motivo el modelo mundial debe ser entrenado por distintas voces y debe estar descrito con un mayor número de regiones. Para crear el modelo mundial se tienen dos opciones, a saber, que solo incluyan a los locutores que forman parte de la base de datos o que se cree a partir de las voces de un conjunto desconocido de personas.

La primera opción cuenta con la ventaja de que teniendo las voces de las bases ya se tiene todo el audio necesario para proceder a crear el modelo mundial, pero tiene la gran desventaja que se debe reconstruir la base mundial cada vez que se requiera agregar o quitar a un usuario de la misma. Esto último consume mucho tiempo y costo computacional, ya que las bases mundiales son del orden de 30 a 60 minutos (dependiendo de la cantidad de usuarios en la base de datos) lo que hace que sea un sistema lento y poco práctico a la hora de modificar el número de locutores.

La segunda opción, al no tener en cuenta los archivos MP3 de los locutores de la base de datos, se independiza de ella y no es necesario modificar al modelo mundial cada vez que se modifica la base de datos. Para eso es necesario crear una única vez, un modelo mundial con gran variedad de voces y de larga duración.

Para nuestro caso creamos el modelo mundial empleando la segunda opción, con una hora de duración y utilizamos 32 regiones. Se eligió esta cantidad porque hay más parámetros para poder ajustar mejor el modelo a los datos.

Los algoritmos con cuales fueron creados los modelos mundiales fueron VQ y el modelo diagonal, pues como se vera más adelante, estos son los más rápidos.

4.4.3. Base de datos MP3 para testeo del reconocimiento de locutores.

Esta base fue obviamente creada con voces de los mismos locutores a los que se les hallaron los modelos. La característica de estos archivos MP3 es que fueron grabados en días distintos a los usados en la generación de modelos por lo que se pueden dar distintos tipos de distorsiones en las voces de los locutores. El tiempo de estos tests asciende a los 45 minutos.

Capítulo 5

Marco experimental

En la sección **2.2** se discutieron en detalle las dos grandes ramas del reconocimiento de locutores, *Verificación* e *Identificación de locutores*. Como nuestro proyecto no estaba enfocado específicamente a ninguna de las dos aplicaciones, optamos por llevar a cabo las dos.

Si bien inicialmente nos dedicamos a la *Identificación*, de *grupo de cerrado*[15], por ser la más sencilla de las dos[16], una vez finalizada esta etapa proseguimos con *Verificación*. Para esto debimos aplicar nuevas ideas y conceptos. No obstante, lo aprendido, utilizado y diseñado en *Identificación* fue imprescindible para el correcto desarrollo de este nuevo emprendimiento.

Las pruebas para *Identificación* se realizaran con ambas bases así como diferentes tipos de métricas y modelos de generación. En lo que a *Verificación* refiere, tomaremos la mejor de las bases y la mejor métrica observadas en *Identificación*. Como la *Búsqueda* no es más que una sucesión de tests de *Verificación* y esta se puede pensar como un caso particular de *Identificación,* implementaremos la *Búsqueda* con los mejores resultados obtenido de los ensayos de *Identificación* y *Verificación*.

5.1. Tiempos en la generación de modelos.

Como ya hemos visto, se van a generar los modelos con 4 métodos distintos. Para calcular el tiempo de generación de los modelos se tomaron cinco juegos de tests de diferentes duraciones y se los promedió arrojando los resultados de la tabla 5.1. Según las pruebas concluimos que la generación de dos modelos distintos, con el mismo tiempo de entrenamiento, no va a tener la misma duración. Esto se debe principalmente a dos causas:

[15] Ver sección 1.2.
[16] La consideramos más sencilla por el hecho de que no se debe tratar con el cálculo de umbrales y el diseño del modelo mundial. En lo demás la dificultad es la misma.

1. Que los métodos se inician con modelos aleatorios que están más o menos cerca del modelo final al cual converge. Esto se ve reflejado en que se tenga una cantidad de pasos variable para generar el modelo final.

2. Debido a la variabilidad de la voz (de la misma persona) que haya en el archivo de entrenamiento. Si la información con la que se va a entrenar el modelo esta muy dispersa, se necesitará iterar más veces para finalizar, en cambio cuando la voz es muy parecida y no hay mucha variabilidad entonces se necesitaran menos pasos para converger a un modelo final.

Las condiciones en que se hicieron estas pruebas son: 16 regiones de 19 coeficientes MP3CEP.

Tiempo de entrenamiento	**Tiempo de generación de modelos**			
	VQ	**GMM diag.**	**HÍBRIDO**	**GMM**
30 segundos	3,5 seg.	8,8 seg.	1: 05 min.	1: 32 min.
1 min	7,5 seg.	12,5 seg.	1: 32 min.	2: 19 min.
2 min	11,9 seg.	18,0 seg.	2: 05 min.	3: 17 min.
3 min	23,1 seg.	36,8 seg.	3: 20 min.	4: 08 min.
4 min	36,5 seg.	46,1 seg.	3: 34 min.	5: 04 min.

Tabla 5.1. Tiempos promedio de demora para generar un modelo con los distintos métodos y tiempos de entrenamiento

Según la tabla 5.1, el algoritmo VQ supera en velocidad al resto de los métodos, aunque esa diferencia disminuye a medida que aumenta el tiempo de entrenamiento. También se evidencia la reducción en la cantidad de pasos necesarios para la convergencia del algoritmo GMM al inicializarlo con VQ (método HÍBRIDO).

Si bien VQ y GMM diagonal son bastante más rápidos no podremos concluir cual es el que brinda mayor performance en tanto no se realicen las pruebas de rendimiento.

5.2. Identificación.

Se crearon tests que consistían en grabaciones, de algunos segundos, de aquellos locutores que conocemos sus modelos (forman parte de la base de datos). Estas fueron adquiridas en días distintos a las grabaciones que se usaron para generar los modelos. Como en este caso se parte de la hipótesis que los tests van a ser siempre de un único usuario de la base de datos, el sistema invariablemente se decidirá por uno de ellos. Si la decisión es correcta se computara como un acierto y si no es correcta no se computara nada. Luego de haber probado con todos los tests se hallara el porcentaje de aciertos como:

$$Aciertos(\%) = 100 * \frac{Cantidad\_de\_Aciertos}{Cantidad\_de\_\mathrm{Pr}uebas}$$

Una vez generados los modelos fueron testeados con la Norma Dos, la distancia Mahalanobis, y la métrica propia de GMM (Logl). Paralelamente se hicieron las mismas pruebas para los modelos GMM diagonales, testeados solamente con el Logl. Además todos los testeos se realizaron con diferentes intervalos de tiempo desde 13 mseg, hasta 4 segundos, como se muestra en la tabla 5.2.

BASES A:

Duración	% de aciertos									
	VQ			GMM			HÍBRIDO			GMM diag.
	Norm.	Maha.	Logl	Norm.	Maha.	Logl	Norm.	Maha.	Logl	Logl
13 ms	16.6	18.2	19.5	15.6	16	17.7	16.7	17.2	18.4	19.7
250 ms	35.8	38.6	43.7	28.9	33.3	39.9	35.2	38.5	43.1	44.3
500 ms	43.7	44.5	51.1	34.4	37.9	47.5	43.0	44.5	51.3	52.9
1 seg	48.9	48.4	56.8	38.5	41.6	52.2	48.5	49.6	56.3	59.8
2 seg	54.3	52.9	60.7	41.8	45.1	55.5	53.3	51.7	60.3	64.4
3 seg	55.7	53.6	60.2	44.0	46.8	55.5	53.6	54.1	61.6	66.7
4 seg	57.3	55.6	62.5	43.4	47.0	56.3	54.6	54.6	62.3	68.2

Tabla 5.2. Se detallan los porcentajes de aciertos para los cuatro métodos de generación e modelos, testeados con diferentes métricas para las bases A

En primer lugar notamos que el mejor resultado se obtiene para el modelo GMM diagonal. Esto se debe a que la correlación que hay entre las distintas componentes de los coeficientes MP3CEP (los elementos no diagonales de la matriz de covarianza) es prácticamente nula. De considerarse no nula, como es el caso de los modelos completos, esto no contribuye a distinguir entre los modelos, es decir, no caracteriza adecuadamente al locutor.

Otra forma de comprender esto es la siguiente, consideremos al modelo diagonal como un modelo completo salvo que las componentes no diagonales de la matriz de covarianza son nulas. Ahora bien, al momento del ajuste de los parámetros el error introducido por los elementos no diagonales será nulo y habrá más margen para el ajuste de la diagonal. Como la condición de salida esta fija, el GMM diagonal es quien tiene un mejor ajuste de la diagonal.

Esto es causa del tipo de base, las bases A son pequeñas y no contienen mucha información sobre la variabilidad de la voz en los locutores, por lo que es perjudicial considerar más parámetros de estas bases. Esto es lo que se llama *sobre ajuste* a datos pobres.

Conviene usar modelos completos cuando hay una basta cantidad de datos (es decir una mayor variabilidad de la voz) como es el caso de las bases B

De estos resultados también podemos concluir que el tipo de modelo (completo) elegido no influye tanto en comparación al tipo de medida con la que se testea. Logl es la medida que mejores resultados da para todos los casos como también para las distintas duraciones de tests, mientras que Mahalanobis y la Norma se comportan de manera similar.

Se realizaron los mismos tests para las bases B obteniéndose los resultados descriptos en la tabla 5.3. En primer lugar se observa, como era de esperarse, el gran aumento en la performance del sistema en comparación con las bases A.

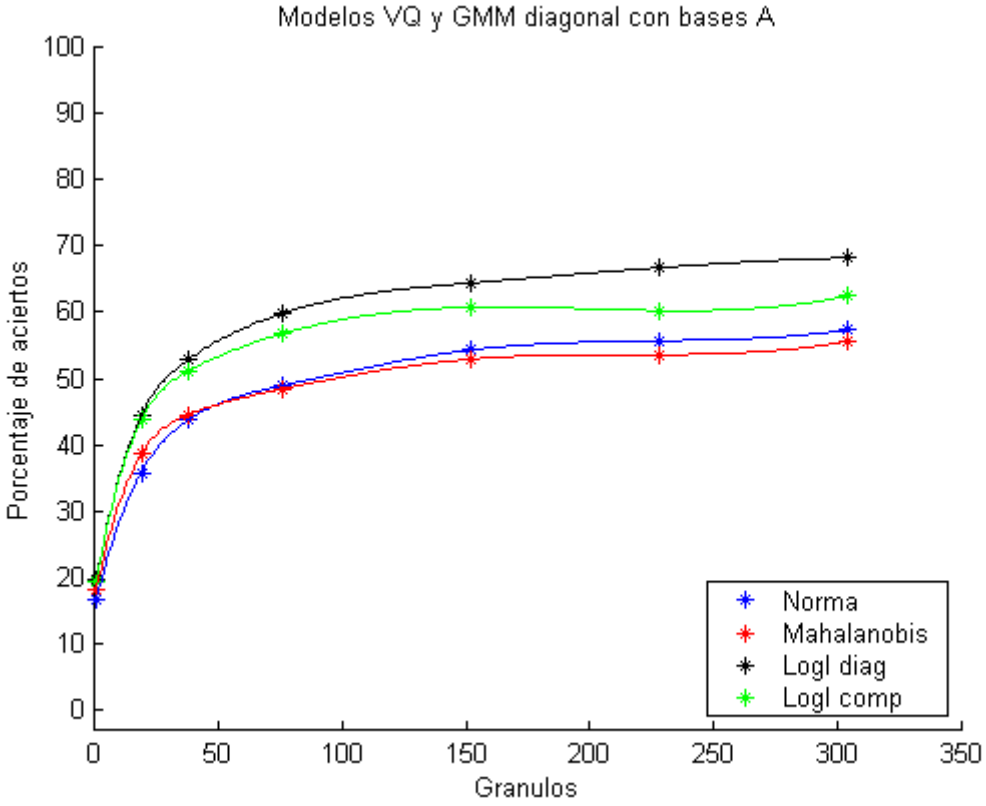


Figura 5.1. En la figura se muestra como los modelos generados a partir de la diagonal brindan una mayor performance frente al mejor de los modelos completos. Puede verse también la convergencia de estas métricas al ir incrementando el largo del test.

Con estas bases también se observa que hay una influencia mayor en la medida usada, más que en el tipo de modelo usado, donde la medida Logl tiene una amplia ventaja sobre las otras medidas. La mejor performance para Logl se obtiene con las bases generadas por el algoritmo HÍBRIDO. Esto era de esperarse debido a que el algoritmo HÍBRIDO ajusta más a los datos que el algoritmo EM para la misma condición de salida, por eso es de esperarse mejores resultados en el modelo HÍBRIDO para medidas estocásticas como Logl.

Como se observa en la figura 5.2 Al igual que en las bases A, aquí el modelo GMM diagonal tienen un buen desempeño. Sin embargo, en este caso las matrices de covarianza completas, juegan un papel más importante, debido a que las bases ya no son tan pobres y sus componentes no diagonales contribuyen al modelado de las mismas.

BASES B:

Duración	% de aciertos									
	VQ			GMM			HÍBRIDO			GMM diag.
	Norm.	Maha.	Logl	Norm.	Maha.	Logl	Norm.	Maha.	Logl	Logl
13 ms	22.0	25.7	31.2	21.2	23.2	32.3	22.0	23.9	31.7	27.7
250 ms	51.7	52.0	69.0	48.5	47.6	69.8	50.3	46.6	69.9	63.0
500 ms	62.0	58.5	77.3	58.0	55.9	78.6	59.8	53.7	78.7	72.3
1 seg	68.7	62.7	82.6	65.4	60.7	84.1	66.3	58.3	84.1	79.3
2 seg	73.6	65.3	86.1	70.5	65.3	87.6	71.6	60.7	87.7	84.2
3 seg	74.8	65.8	88.3	72.7	66.5	89.1	73.7	62.9	90.2	85.8
4 seg	74.5	65.9	88.3	72.8	67.3	88.6	74.2	63.0	89.5	86.9

Tabla 5.3. Se detallan los porcentajes de aciertos para los cuatro métodos de generación de modelos, testeados con diferentes métricas para las bases B

La performance para la base B es del orden de un 20% mayor para todas las combinaciones de modelos y métricas. De este modo optamos por trabajar para *Identificación* con las bases B. En cuanto a la métrica elegimos el logl por tener resultados notoriamente más amplios que cualquier otra métrica. Al no haber una diferencia tan notoria en la performance entre los distintos tipos de métodos para la base B, se optó por el GMM diagonal debido a su buen desempeño a la hora de generar modelos y al menor costo computacional en lo que a testeo se refiere.

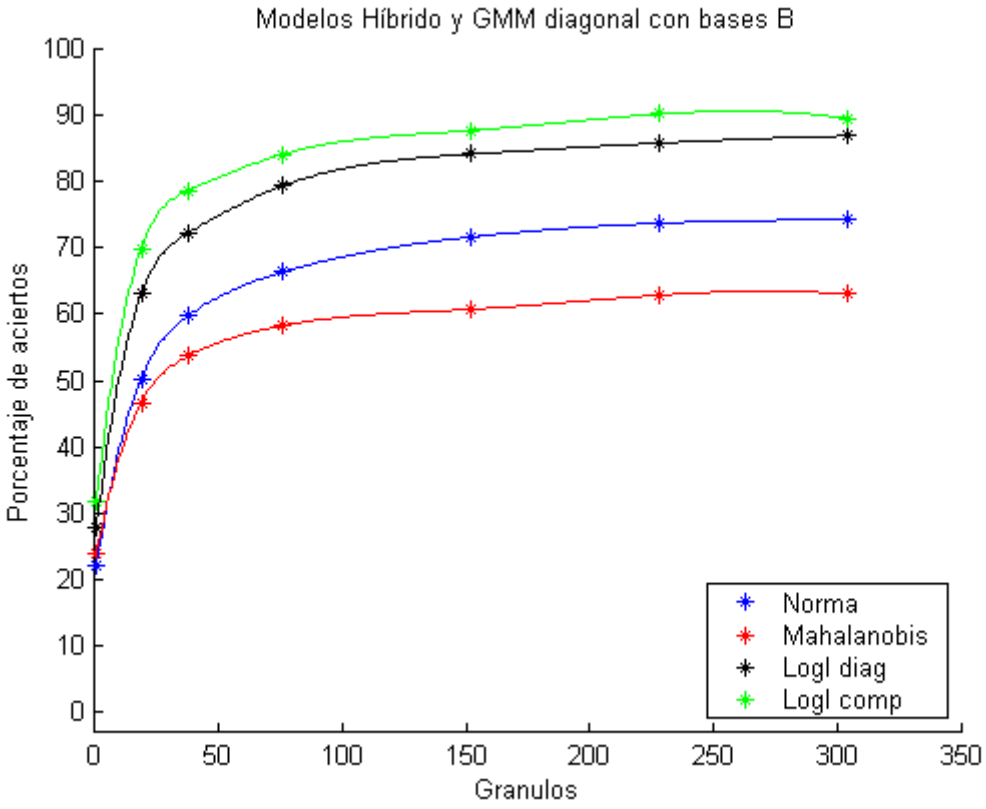


Figura 5.2 Comparación de las métricas sobre los modelos HÍBRIDOS Vs el modelo Diagonal.

5.3. Verificación

Se usaron los mismos tests que en *Identificación*. A partir de estos tests se llevaron a cabo una serie de experimentos variando el umbral, en forma arbitraria en el rango [-1...1][17] y observamos el porcentaje de falsas aceptaciones (FA) y de falsos rechazos (FR) para cada umbral. El umbral óptimo se eligió tal que el porcentaje de FA sea igual al porcentaje de FR.

Debido a los resultados que obtuvimos en *Identificación*, aquí solamente trabajamos con la medida Logl, por tener mejores resultados. En cuanto al modelo mundial este fue creado con dos métodos VQ y GMM diagonal, debido al tiempo que estos toman y a su ya reconocida capacidad.

Aquí solo consideramos las bases B, los cuatro tipos de modelos de los locutores y las distintas duraciones de tests para efectuar las pruebas.

La relación que se obtuvo entre las distintas bases así como las métricas fue igual a la descrita para *Identificación*. Por esto consideraremos las bases B, únicamente con la métrica del logl. En la tabla 5.4 se muestran los resultados para las bases B

BASES B

Duración	**GMM diagonal**		**GMM completo**		**VQ**		**HIBRIDO**	
	Umbral	% EER	Umbral	% EER	Umbral	% EER	Umbral	% EER
13 ms	-0.37	26.4	0.11	21.7	0.10	21.0	0.12	21.2
250 ms	-0.39	13.6	0.19	8.2	0.12	8.5	0.18	8.2
500 ms	-0.40	11.0	0.20	6.2	0.18	6.1	0.21	6.2
1 seg	-0.41	8.9	0.27	4.6	0.27	4.3	0.29	4.3
2 seg	-0.45	7.9	0.19	4.2	0.16	4.1	0.17	4.2
3 seg	-0.48	7.6	0.21	4.0	0.28	3.5	0.19	4.0
4 seg	-0.49	7.2	0.12	4.3	0.22	3.3	0.19	3.8

Tabla 5.4 umbrales óptimos con sus respectivos porcentajes EER para cada modelo

A diferencia que en *Identificación*, el modelo que mejor rendimiento tiene es el VQ. La performance va mejorando a medida que los tests van aumentando su duración, inclusive hasta cuatro segundos. Si bien esto es así, si se observa la figura 5.3b notamos que para tiempos mayores 500 ms (38 gránulos) el error ya no decae tanto y para una aplicación de búsqueda rápida, es razonable tomarlo en esta cantidad para evitar intervalos de búsqueda demasiado largos.

Si se requiere una aplicación cítrica en tiempos de ejecución como puede ser la búsqueda, el modelo GMM diagonal se vuelve una opción valedera.

[17] Vimos en la sección 1.4.4. que el umbral teórico es $\theta = 0$.

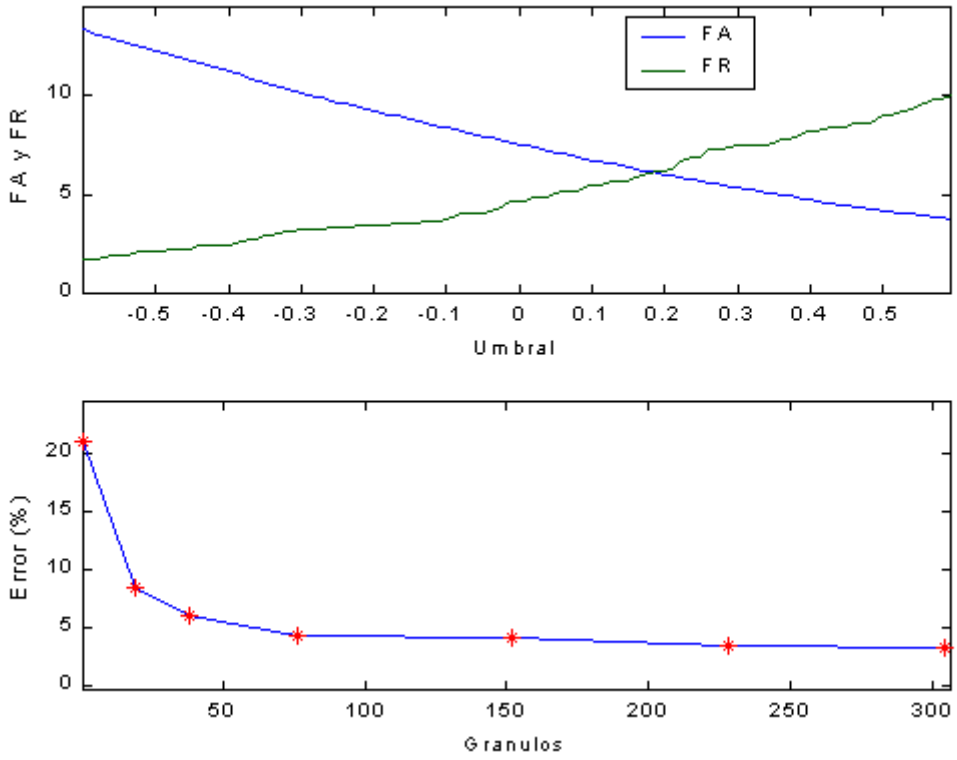


Figura 5.3 A) Curvas de FA y FR para el modelo VQ (t = 500 ms). B) Porcentaje de error eligiendo como umbral la intersección de ambas curvas.

5.4. Búsqueda.

Para esta aplicación en particular no se fijara el umbral a pesar de ser esta un caso particular de *Verificación*, ya que la única diferencia sustancial es que se debe verificar de a pequeños tramos en un archivo más grande. La razón por la cual no se fija, es por que los resultados dependen de la duración del archivo en el que se desee buscar, así como también del largo del discurso del locutor.

Supongamos que se tiene el valor de 10% tanto para FA como FR. Supongamos además que el archivo en el cual se va a llevar acabo la búsqueda es de 100 minutos y el locutor que en cuestión habla durante 5 minutos. Con estos valores el resultado de la búsqueda podría ser de unos 9:30 minutos de FA y de 30 segundos de FR. Este resultado podría ser muy molesto ya que se extraen 14 minutos de los cuales 4:30 minutos esta el locutor hablando. Si nos guiamos por el ejemplo anterior el sistema no parece tener muy buenos atributos, sin embargo si el locutor hablase durante 40 minutos se extraerán unos 6 minutos de FA y se perderán unos 4 minutos por FR para extraer un total de 42 minutos.

Para la Búsqueda de locutores en archivos MP3, nos basamos en los r características esultados obtenidos a partir de *Verificación* de locutores, ya que es una aplicación directa. Además se utilizó la base B por ser la que dio mejor performance. El método de generación de modelos a usar es VQ pues es el que brinda mejor relación aciertos/tiempo.

El funcionamiento de este sistema se basa en ventanas deslizantes de tamaño fijo, donde cada ventana toma un tramo del archivo, verifica, y decide si lo acepta o rechaza. Luego la ventana se desliza un 100% y toma la siguiente parte del archivo para así continuar sucesivamente con el proceso hasta que el archivo termine. Cabe destacar que lo que adquiere la ventana, es lo que llamamos test, y el largo de la ventana es la duración del test.

Al variar el tamaño de la ventana, varía también la performance según la tabla 5.4, pero no solo esto es importante, el tiempo que demora en procesar la búsqueda es clave. La medida Logl con los 3 métodos, trabaja con la matriz de covarianza completa y tiene más cálculos que procesar para poder decidir, en cambio en el modelo GMM diagonal estos cálculos se reducen, lo que se ve reflejado en el tiempo de parametrización. Estos son los resultados observados:

Duración del archivo	**Tiempo de adquisición**	**Tiempo de parametrización**	
		GMM diagonal	**Otros**
30 seg	3.3 seg	0.12 seg	1.0 seg
7 min	39.3 seg	1.6 seg	8.9 seg
30 min	172.4 seg	6,50 seg	37.6 seg

Tabla 5.5. Tiempos para GMM diagonal Vs los modelos completos.

Los tiempos de parametrización con GMM diagonal son considerablemente menores, lo cual acelera el tiempo de búsqueda. Cabe destacar que el tiempo total de búsqueda es la suma de tiempo de parametrización más el tiempo de adquisición de los MP3CEP. Este tiempo de adquisición, es independiente del modelo que se use y se observa que el tiempo total de GMM diagonal es aproximadamente un 85% del tiempo total que demoran los otros métodos (con los modelos completos).

Uno de los problemas que presenta la *Búsqueda* es el tiempo que lleva analizar un archivo de gran tamaño, con el fin de mejorar los tiempos totales de búsqueda, la misma se hará analizando uno de cada N ventanas. De acuerdo a los resultados de la sección anterior podemos fijar el largo de la ventana en medio segundo y dependiendo del tiempo mínimo que hable cada locutor en el archivo, se elegirá un N adecuado. Por ejemplo si suponemos que cada locutor habla como mínimo cinco segundos entonces N debe ser menor que diez, para no perderlo.

Notamos de la Tabla 5.5 que el tiempo total de búsqueda completa (con N = 1), se reduce al 10% de la duración del archivo. Este es el tiempo máximo que puede durar el proceso de búsqueda, pero a su vez es el más efectivo pues analiza todos los tramos. Generalizando para cualquier valor de N, el tiempo de búsqueda se reduce en N, no obstante la efectividad se ve algo diezmada.

Otro de los problemas de la *Búsqueda* es la gran cantidad de FA que se pueden obtener, dependiendo del largo del archivo. Estas FA se presentaran como espurios en el transcurso del archivo, de modo que paliamos este problema implementando un filtro de medianas.

Como era de esperarse la performance del sistema varía entre los locutores dependiendo de su condición de panelista o cronista. Por ejemplo si el locutor buscado es un panelista, el cual generalmente habla en las mismas condiciones (puede variar la forma de hablar), los resultados son muy buenos y hasta a veces mejores que los desplegados en la tabla 5.4. Sin embargo, al probar con periodistas que siempre hablan en diferentes condiciones y en

muchas ocasiones con ruido ambiente, los resultados son parecidos a los de la tabla 5.4, aunque a veces debido al mucho ruido de fondo, la performance decrece. En la tabla 5.6 se muestra un ejemplo de esto donde Fernando y María Inés son panelistas en tanto que Roberto y Elsa son cronistas.

Fernando		María Inés		Roberto		Elsa	
Real	Hallado	Real	Hallado	Real	Hallado	Real	Hallado
0:0 - 0:22	0:0 - 0:22	1:42 - 2:07	1:41 - 2:07	13:04 - 13:13	13:06 - 13:13	2:07 - 2:44	2:09 - 2:43
6:55 - 7:51	6:53 - 7:52	9:37 - 10:07	9:37 - 10:07	13:13 - 13:24	FR		
11:22 – 11:41	11:20 - 11:39	12:48 - 13:04	12:47 - 13:04	13:24 - 13:47	13:24 - 13:47		
14:54 – 15:11	14:51 - 15:10	15:57 - 16:18	15:55 - 16:16	18:57 - 19:15	FR		
17:15 – 17:36	17:12 - 17:35	18:34 - 18:54	18:31 - 18:54	FA	19:59 - 20:05		

Tabla 5.6 Búsqueda para 72 gránulos, N = 2 y EER = 0.17

5.5. Consideraciones adicionales sobre el bitrate.

Para determinar la influencia del bitrate de los MP3 en los resultados obtenidos, efectuamos pruebas de *Identificación* variando el bitrate de los tests para los modelos VQ anteriormente obtenidos (@ 64Kbps). Los resultados obtenidos se ven en la tabla 5.7. Concluimos que los cambios son insignificantes.

BASE B:

Duración	**Modelo VQ (% de aciertos)**								
	64 kbps			**32 kbps**			**160 kbps**		
	Norm.	Maha.	Logl	Norm.	Maha.	Logl	Norm.	Maha.	Logl
13 ms	22.0	25.7	31.2	22.0	25.0	30.7	21.8	25.5	31.1
250 ms	51.7	52.0	69.0	53.3	51.3	70.1	51.6	53.1	68.7
500 ms	62.0	58.5	77.3	63.3	58.7	78.2	61.3	60.4	76.9
1 seg	68.7	62.7	82.6	69.3	62.9	82.6	69.1	63.5	82.3
2 seg	73.6	65.3	86.1	75.1	67.8	87.8	73.1	66.0	85.8
3 seg	74.8	65.8	88.3	76.2	66.6	88.8	74.9	65.9	88.3
4 seg	74.5	65.9	88.3	74.4	65.7	89.8	73.9	66.1	88.4

Tabla 5.7.

Esto se debe a que para codificar la voz es suficiente con 32 kbps, el uso de un bitrate mayor es innecesario puesto que todo el bitrate se reparte entre las primeras bandas hasta los 5.3 kHz.

5.6. MP3CEP Vs MFCC

Los coeficientes MFCC han sido concebidos para obtener buena performance mientras que a los MP3CEP los diseñamos primando la velocidad. La figura 5.4 ilustra el tiempo de parametrización para ambos esquemas.

Cabe notar que estos tiempos están sumamente ligados a la eficiencia de los códigos con los cuales fueron implementados, sin embargo es de destacar que dichos códigos fueron optimizados sucesivamente.

Claramente se puede notar que el tiempo consumido por los MP3CEP es del orden de cinco veces menor que por los MFCC. Esto es por consecuencia de dos razones. En primer lugar el mayor tiempo de descompresión, pues los MFCC requieren descomprimir todo el archivo. En segundo lugar un mayor tiempo de parametrización, ahora con la señal en el tiempo hay que nuevamente entramar y transformar a frecuencia.

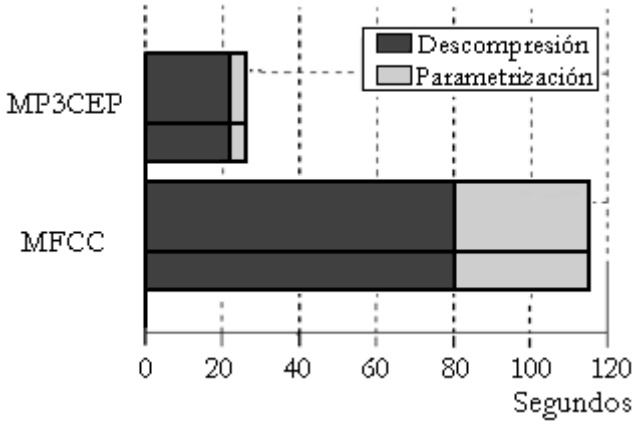


Figura 5.4. Comparación del tiempo de procesamiento (descomprimir y parametrizar) las características MP3CEP y MFCC para un archivo MP3 de 3:36 segundos

En la tabla 5.8 muestra se puede notar que los coeficientes MFCC tienen un mayor rendimiento que los MP3CEP. Sin embargo esta diferencia es insignificante para la norma y logl.

Duración	**MP3CEP** (% de aciertos)			**MFCC** (% de aciertos)		
	Norm.	**Maha.**	**Logl**	**Norm.**	**Maha.**	**Logl**
13 ms	22.0	25.7	31.2	23.5	28.2	34.4
250 ms	51.7	52.0	69.0	51.3	55.4	69.5
500 ms	62.0	58.5	77.3	61.9	62.7	77.8
1 seg	68.7	62.7	82.6	69.3	67.1	82.9
2 seg	73.6	65.3	86.1	73.4	69.8	86.9
3 seg	74.8	65.8	88.3	75.3	70.3	88.6
4 seg	74.5	65.9	88.3	77.0	70.0	89.5

Tabla 5.8. Comparación de rendimientos para la base B con el modelo de generación VQ y todas las métricas.

La mayor diferencias para logl se da para tests de 4 segundos donde los MFCC alcanzan su mayor performance en 89.5%. Esta diferencia es de apenas de 1.2% lo cual es un muy buen resultado en particular si se toma en cuenta que los MP3CEP usan un 13% menos vectores que los MFCC.

En resumen podemos decir que los coeficientes MP3CEP son tan buenos como los MFCC, sin embargo partiendo de la señal comprimida no tiene sentido usar los MFCC pues el tiempo que esto insume es mucho mayor.

5.7. Problemas encontrados.

Dentro de la gran gama de problemas enfrentados el más significativo fue el de la obtención de señales tanto para entrenamiento como para testeo. Esto se debió a los distintos tipos de ruidos, a saber, mala sintonización, interferencia de la red, etc.

Esto se solucionó descartando aquellas señales con altos niveles de ruido, viéndonos obligados a adquirir una mayor cantidad de señales puesto que es inviable siquiera pensar en filtros dedicados a cada tipo de ruido.

La interferencia de la red afecta a las componentes hasta 70 Hz, de manera que a la hora de calcular los coeficientes Mel se anulo la salida del primer filtro Mel, es decir forzamos a uno su valor para mantener inalterado a los MP3CEP.

Otro tipo de problemas muy distinto al anterior fue la disponibilidad de información. Si bien la información obtenida sobre reconocimiento de locutores fue basta y rica en conceptos, lamentablemente no podemos decir lo mismo de la información sobre MP3. Esto nos privó de implementar nuestro propio decodificador, viéndonos obligados a recurrir a un código ajeno.

Capítulo 6

Conclusiones

Después de los resultados obtenidos podemos decir que finalmente se llego al objetivo buscado y que es posible reconocer locutores trabajando con MP3 sin tener que descomprimir todo el archivo. Resaltamos los siguientes resultados:

En primer lugar se puede concluir que los coeficientes MP3CEP describen de buena manera las características de la voz con una performance similar a la de los coeficientes MFCC. Como este proyecto supone que la señal de partida es una señal MP3, calcular los coeficientes MP3CEP resulta ser cinco veces más rápido que calcular MFCC.

En lo que refiere a los modelados de las características de la señal de voz, podemos concluir que, como era de esperarse, es conveniente alimentar al sistema considerando una mayor variabilidad de señales. Este es el caso de las bases A y B, donde las bases B obtuvieron un 20% más de aciertos.

Respecto a generar los modelos con métodos estocásticos o métodos de vecindad el resultado es similar, siempre y cuando se tomen en cuenta las consideraciones hechas para VQ. Esto lo vemos en la paridad entre los modelos VQ, GMM e HIBRIDO. Sin embargo el tiempo de generación de VQ es considerablemente menor que el resto. En cuanto al método GMM diagonal, podemos decir que un 15% más rápido.

Por último si comparamos los MP3CEP con los MFCC obtenidos a partir de los archivos WAV originales tenemos que:

- Se ahorra 90 % de espacio de almacenamiento.
- Se demora un 33% menos en parametrizar los MP3CEP.
- La performance es prácticamente la misma.

Apéndice A-1

Linear Prediction Coefficients[15]

La voz usualmente se modela mediante una serie de pulsos de la laringe, que pasan a través de una función de transferencia que contiene únicamente polos y que representa el efecto del tracto vocal sobre la señal de voz[15]. Los LPC (Linear Prediction Coefficients), representan la ubicación específica de los polos de la función de transferencia. Mas precisamente, los LPC son el resultado de predecir cada muestra de voz como una combinación lineal de un cierto número de muestras anteriores (ec. A-1.1).

$$s(n) = -\sum_{k=1}^{p} a_k .s(n-k) + G.u(n) \qquad \text{(A-1.1)}$$

donde u(n), la entrada al filtro, será un tren de impulsos periódicos o una fuente de ruido aleatorio[15]. El tren de impulsos producirá señales sonoras mientras que la fuente de ruido aleatorio producirá señales no sonoras a la salida del filtro. De esta manera el filtro representa un modelo del tracto bucal, como se indica en la figura A-1.1.

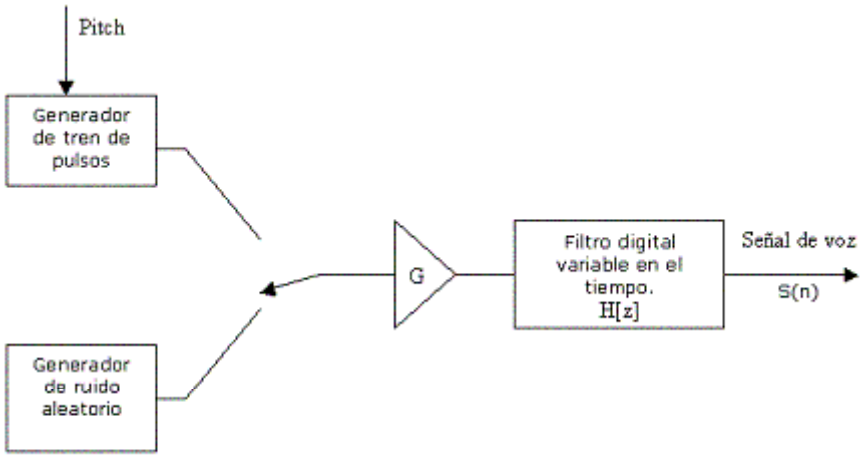


Figura A-1.1. Representación del tracto bucal mediante un FIR.

La función de transferencia del filtro se obtiene haciendo la transformada Z a la ecuación (A-1.1):

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k . z^{-k}} \qquad \text{(A-1.2)}$$

Donde ***G***, la ganancia del filtro, depende de la naturaleza de la señal. Entonces, dada la señal s(n), el problema consiste en determinar los LPC $\boldsymbol{a_k}$ y la ganancia ***G***.

Los LPC son típicamente calculados para intervalos de tiempo cortos, sobre los cuales se asume invarianza temporal. La duración de estos intervalos es de entre 10ms y 30ms y usualmente se enventanan mediante una ventana de Hamming o alguna otra función de enventanado similar.
Serán los coeficientes LPC los que se usen como parámetros de reconocimiento. Su determinación se realiza minimizando el error que se comete cuando se intenta realizar la aproximación de la señal. Sea $\tilde{s}$ la señal predicha a partir de la señal s original, entonces:

$$\tilde{s}(n) = -\sum_{k=1}^{p} a_k . s(n-k) \qquad \text{(A-1.3)}$$

El error entre la señal real y la señal predicha es:

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^{p} a_k . s(n-k) \qquad \text{(A-1.4)}$$

Mediante el método de mínimos cuadrados, los coeficientes LPC se calculan minimizando el error cuadrático medio con respecto a cada uno de los coeficientes. Sea E el error cuadrático total:

$$E = \sum_{n} e^2(n) = \sum_{n} (s(n) + \sum_{k=1}^{p} a_k . s(n-k))^2 \qquad \text{(A-1.5)}$$

Se minimiza con respecto a $\boldsymbol{a_k}$:

$$\sum_{k=1}^{p} a_k . \sum_{n} s(n-k).s(n-1) = -\sum_{n} s(n).s(n-i) \ , \ 1 \le i \le p \qquad \text{(A-1.6)}$$

De las dos relaciones anteriores (A-1.5) y (A-1.6) se deduce la ecuación:

$$R(i) = \sum_{n=0}^{N-1-i} s(n).s(n+i), \quad i \ge 0 \qquad \text{(A-1.7)}$$

A continuación se realiza un análisis de autocorrelación. La función de autocorrelación proporciona una medida de la correlación de la señal con una copia desfasada en el tiempo de si misma. Se define ***p*** como el orden de análisis que es a su vez la cantidad de coeficientes de autocorrelación. Valores típicos de ***p*** pueden ser entre diez y quince. Podemos identificar los coeficientes de autocorrelación en las ecuaciones que minimizan

los errores en la estimación de la señal predicha. Para resolver este conjunto de ecuaciones de manera eficiente se recurre al algoritmo de *Levinson-Durbin*:

$$
\begin{aligned}
&E_0 = R(0) \\
&k_j = -\frac{R(i) + \sum_{j=1}^{j-1} a_{j,j-1}.R(i-j)}{E_{j-1}} \\
&a_{i,j} = k_j \\
&a_{j,i} = a_{j,i-1} + k_j.a_{i-j,i-1} \qquad 1 \le j \le i-1 \\
&E_j = (1-k_j^{\,2}).E_{j-1} \\
&Solución\ final: a_j = a_{j,p} \quad 1 \le j \le p
\end{aligned}
\qquad \text{(A-1.8)}
$$

Una vez hallados los coeficientes a_k se dispone, para la ventana de análisis, de la función de transferencia del modelo del tracto vocal en ese instante. Es decir se conoce la forma con la que la cavidad bucal se comporta y junto con la señal de excitación se puede obtener el sonido emitido en ese momento.

Para comprobar este hecho podemos comparar el espectro LPC adquirido, con el espectro de la señal obtenida mediante la transformada discreta de Fourier, de una porción de señal correspondiente a una vocal durante 35 milisegundos (figura A-1.2). Se observa que el espectro LPC (figura A-1.4), en cierta forma envuelve al espectro de la señal (figura A-1.3). Es decir que coinciden en las resonancias, que son las que caracterizan al contenido frecuencial de la señal vocal[15].

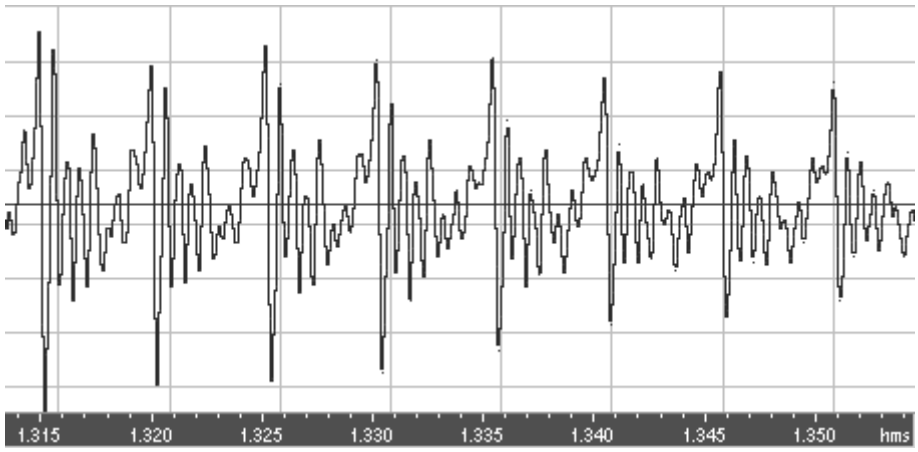


Figura A-1.2. Señal de voz en el tiempo.

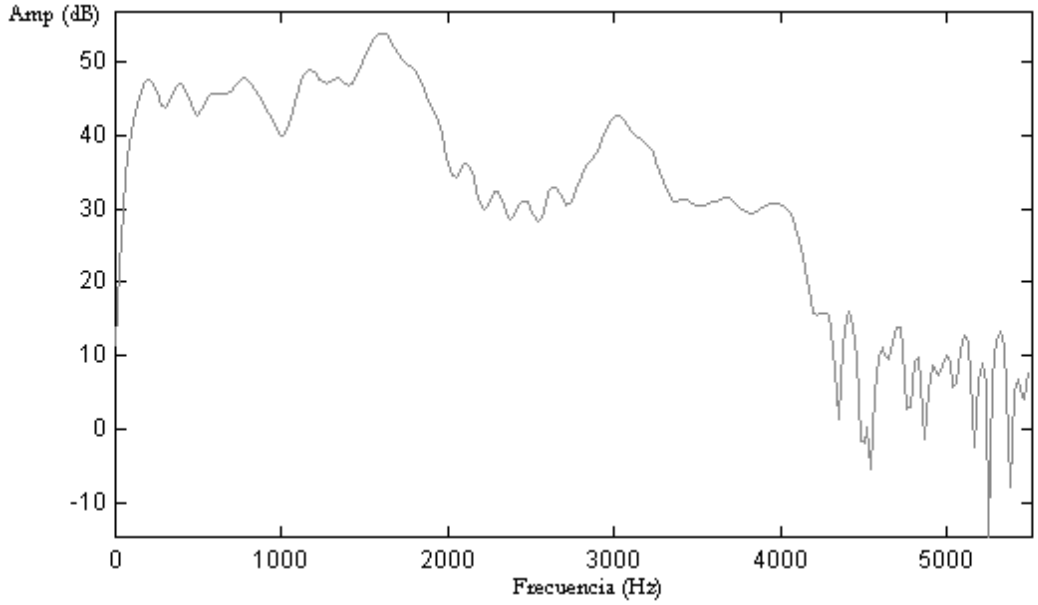


Figura A-1.3. Señal de voz en frecuencia.

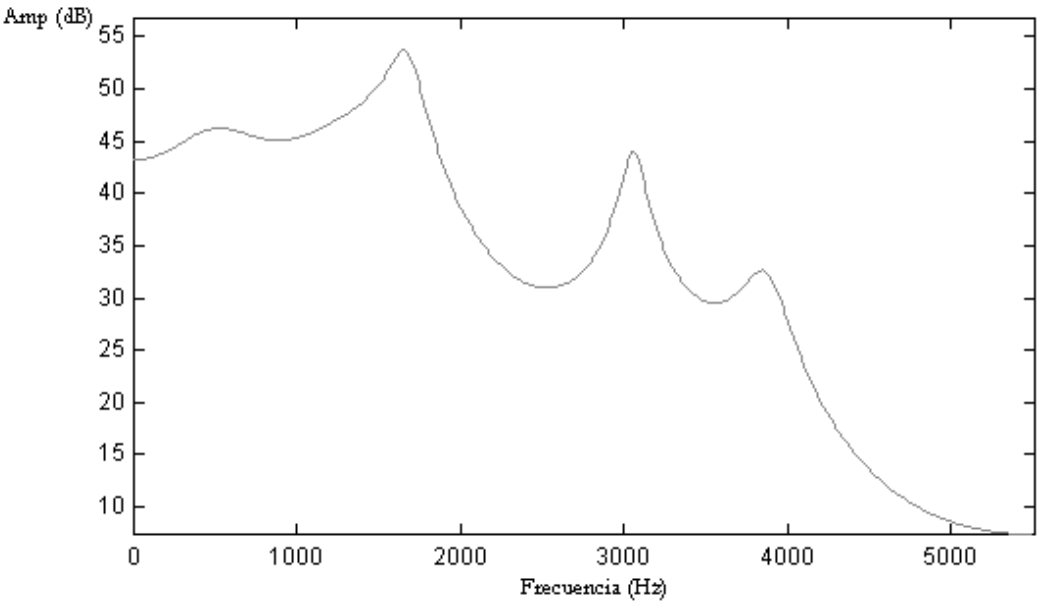


Figura A-1.4. Modelo LPC en frecuencia.

En este ejemplo la magnitud del espectro LPC sigue a la envolvente espectral. Se puede observar que el modelo LPC se ajusta mejor los picos que los valles debido a la mayor contribución de los picos al criterio del error cuadrático medio. A los picos se le llaman frecuencias formantes y a los valles regiones antiformantes.

Al aumentar el orden $\boldsymbol{p}$ del modelo LPC, éste se ajusta mejor al espectro de la voz. Al tender $\boldsymbol{p}$ a infinito el ajuste será exacto.

Apéndice A-2.

Algoritmo EM

A-2.1 Elección del modelo inicial.

Lo primero que se debe hacer es inicializar el modelo, i.e asignar valores de partida a los parámetros del modelo (vector de pesos, matriz de medias, y matrices de covarianza).

Típicamente se toman todos los componentes equiprobables por lo que $w_i = 1/K \quad \forall$i.
En cuanto a los vectores de medias, simplemente usamos algunos vectores de los datos de entrada escogidos aleatoriamente. Es posible que se deban probar distintas inicializaciones para obtener buenos resultados.

Por último para fijar valores razonables para las matrices de covarianza, calculamos la covarianza del conjunto completo de los vectores de entrada.

A-2.2 Descripción del algoritmo.

Para un conjunto de vectores característicos independientes $X=\{\vec{x}_1, \vec{x}_2, ..., \vec{x}_T\}$, se hacen estimaciones de máxima verosimilitud para hallar el modelo λ^* tal que :

$$\lambda^* = \arg\max_\lambda P(\lambda / X) = \arg\max_\lambda \{\frac{P(X/\lambda)P(\lambda)}{P(X)}\} = \arg\max_\lambda P(X/\lambda) \quad \text{(A-2.1)}$$

esta igualdad se obtiene usando el *Teorema de Bayes* y suponiendo que hay equiprobabilidad entre todos los locutores.

El algoritmo EM encuentra a λ^* a través de sucesivas iteraciones tal que:

$P(X/\lambda^{(k)}) > P(X/\lambda^{(k-1)})$ en la k-ésima iteración. A partir de un modelo inicial, se estima un nuevo modelo que se parece mas al conjunto de datos X.

Se usa el modelo original para calcular en cada iteración EM las probabilidades a posteriori para la componente k:

$$c_{kt} = P(k / \vec{x}_t, \lambda) = \frac{w_k \, b_k(\vec{x}_t)}{\sum_{i=1}^{M} w_i b_i(\vec{x}_t)} \quad k = 1...M \qquad \text{(A-2.2)}$$

La ecuaciones EM son:

$$\hat{w}_k = \frac{1}{T}\sum_{t=1}^{T} c_{kt} \qquad \vec{\hat{\mu}}_k = \frac{\sum_{t=1}^{T} c_{kt}\vec{x}_t}{\sum_{t=1}^{T} c_{kt}} \qquad \hat{r}_{kj} = \frac{\sum_{t=1}^{T} c_{kt}(x_{tj} - \mu_{kj})^2}{\sum_{t=1}^{T} c_{kt}} \qquad \text{(A-2.3)}$$

Este algoritmo garantiza que converge hacia un máximo local (que puede no ser un máximo absoluto). Si el máximo local no es el absoluto se deberá proceder nuevamente pero cambiando el modelo inicial de partida.

En la practica se toman dos condiciones de parada que se pueden cumplir una o ambas a la vez. La primera consiste en imponer un número máximo de iteraciones y la segunda en acotar la diferencia entre Logls de pasos consecutivos:

$$\log P(X / \lambda^{(k)}) - \log P(X / \lambda^{(k-1)}) < \varepsilon \text{ (A-2.4)}$$

Cabe recordar que las ecuaciones EM esta formuladas en forma genérica para el modelo completo, si se usara el modelo simplificado donde las matrices de covarianza se consideran diagonales, las operaciones se verán reducidas en gran forma.

A medida que se avanza en el proceso iterativo, se debe tener en cuenta que los elementos de la matriz de covarianza pueden llegar a ser muy pequeños y esto produce singularidades en el algoritmo. Para eso se debe establecer un limite inferior $\sigma_{\min}$ y chequear tras cada iteración si algún elemento de la matriz es menor que $\sigma_{\min}$ y sustituirlo

por $\sigma_{\min}$ si lo es , es decir:

$$\begin{aligned} &si \quad r_{kj} < \sigma_{\min} \Rightarrow r_{kj} = \sigma_{\min} \\ &si \quad r_{kj} > \sigma_{\min} \Rightarrow r_{kj} = r_{kj} \end{aligned} \qquad \text{(A-2.5)}$$

Apéndice A-3

El modelo psicoacústico[3]

A-3.1 Introducción[6]

La psicoacústica estudia la relación entre las propiedades físicas del sonido y la interpretación que el cerebro hace de ellas.

Los objetivos generales de la psicoacústica pueden resumirse en determinar:

- La característica de respuesta de nuestro sistema auditivo, es decir, cómo se relaciona la magnitud de la sensación producida por el estímulo con la magnitud física del estímulo.
- El umbral (absoluto) de la sensación.
- El umbral diferencial de determinado parámetro del estímulo (mínima variación y mínima diferencia perceptibles).
- La resolución o capacidad de resolución del sistema para separar estímulos simultáneos o la forma en que estímulos simultáneos provocan una sensación compuesta.
- La variación en el tiempo de la sensación del estímulo.

En el campo de la psicoacústica se ha hecho un gran progreso hacia caracterizar la percepción auditiva humana y particularmente las capacidades del análisis tiempo - frecuencia del oído interno. Aunque la aplicación de las reglas perceptivas a la codificación de señales no es una idea nueva, la mayoría de los codificadores de audio actuales alcanzan la compresión usando el hecho de que existe información "irrelevante" en la señal, que no es perceptible ni siquiera por personas entrenadas.

Podemos identificar esa información incorporando varios principios de la psicoacústica, como umbrales absolutos de audición, análisis de bandas críticas, enmascaramiento simultáneo y enmascaramiento temporal.

Antes de pasar a estudiar estos conceptos en detalle debemos definir el concepto de nivel de presión sonora (SPL), el cual mide las variaciones de presión de aire en el oído.

Puesto que el SPL se mide en dB es necesario tomar un nivel de referencia, el cual está establecido en Pref = 20 μPa. Entonces el SPL queda definido como:

$$SPL = 20 \log_{10} (P / Pref) \qquad \text{(A-3.1)}$$

El nivel de presión sonora de los sonidos audibles varía entre 0 dB y 120 dB. Los sonidos de más de 120 dB pueden causar daños auditivos inmediatos e irreversibles, además de ser bastante dolorosos para la mayoría de las personas.

A-3.2 Umbral de audición[3]

El umbral de audición representa la sensibilidad del aparato auditivo, es decir, el valor mínimo de presión sonora que debe tener un tono para que sea apenas detectado por un oyente en un ambiente silencioso. El umbral de audición se expresa típicamente en términos de SPL en dB. Para estudiar la dependencia con la frecuencia de este umbral realicemos el siguiente experimento. Tenemos a una persona en una habitación aislada de otros sonidos, hacemos sonar un tono de 1kHz a un nivel mínimo de sonido, y aumentamos el nivel hasta que sea apenas audible. Luego variamos su frecuencia y modificamos su amplitud de modo que siga siendo apenas audible. Finalmente representamos esto en una gráfica:

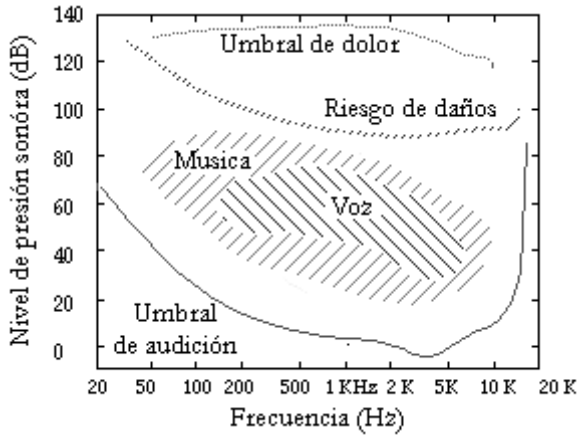


Figura A-3.1. Umbrales de audición

A modo de ejemplo, un tono de 1 kHz y 20 dB SPL será audible pues está por encima del umbral de audición (Figura A-3.1) mientras que un tono de 50 Hz e igual nivel será inaudible pues está por debajo del umbral. El umbral puede ser aproximado por la función no lineal:

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 \text{ (dB SPL)} \qquad \text{(A-3.2)}$$

Esta expresión es representativa de un oyente joven con audición normal. Cuando hablamos de compresión de señales, $T_q(f)$ podría ser interpretado como el nivel de energía máximo permitido para las distorsiones de la codificación introducidas en el dominio de la frecuencia puesto que no serán oídas.

Consideraciones a tener en cuenta:

- Primero, los umbrales captados en la figura A-3.1 se asocian a los estímulos de tonos puros, mientras que el ruido de cuantización en codificadores perceptivos es generalmente complejo y no tonal.
- En segundo lugar, es importante notar que al momento del diseño de los algoritmos no se tiene conocimiento previo de los niveles reales de reproducción. Se asume que el control de volumen en un decodificador típico será fijado de tal forma que la menor señal de salida posible será presentada cerca del 0 dB SPL. Esta hipótesis asumida es conservadora para ambientes ruidosos, y por lo tanto esta práctica se encuentra comúnmente en algoritmos que utilizan el umbral de audición absoluto (SPL)

A-3.3 Bandas críticas[6]

El sonido propagado a través del oído externo y medio llega hasta la cóclea, donde las oscilaciones en los fluidos hacen vibrar a la membrana basilar y a todas las estructuras que ésta soporta.

La membrana basilar es una estructura cuyo espesor y rigidez no es constante: cerca de la ventana oval, la membrana es gruesa y rígida, pero a medida que se acerca hacia el vértice de la cóclea se vuelve más delgada y flexible.

La rigidez decae casi exponencialmente con la distancia a la ventana oval; esta variación de la rigidez en función de la posición afecta la velocidad de propagación de las ondas sonoras a lo largo de ella, y es responsable en gran medida de un fenómeno muy importante: la selectividad en frecuencia del oído interno.

Las ondas de presión generadas en la perilinfa a través de la ventana oval tienden a desplazarse a lo largo de la escala vestibular. Debido a que el fluido es incompresible la membrana basilar se deforma, y la ubicación y amplitud de dicha deformación varía en el tiempo a medida que la onda de presión avanza a lo largo de la cóclea.

Para comprender el modo de propagación de las ondas de presión, supóngase que se excita el sistema auditivo con una señal sinusoidal de una frecuencia dada. La membrana basilar vibrará sinusoidalmente, donde la amplitud de la envolvente irá en aumento a medida que se aleja de la ventana oval, hasta llegar a un punto en el cual la deformación de la membrana basilar será máxima. Resulta así que existe una localización del pico de resonancia de la membrana basilar en función de la frecuencia. Esto confiere al oído interno una cualidad analítica que es de fundamental importancia en la discriminación tonal del sonido, especialmente para los sonidos de frecuencias superiores a los 1000 Hz.

A partir de esa región, la onda no puede propagarse eficientemente, de modo que la amplitud de la vibración se atenúa muy rápidamente a medida que se acerca al helicotrema. En la Figura A-3.2 se observa la onda en la membrana basilar en un instante de tiempo.

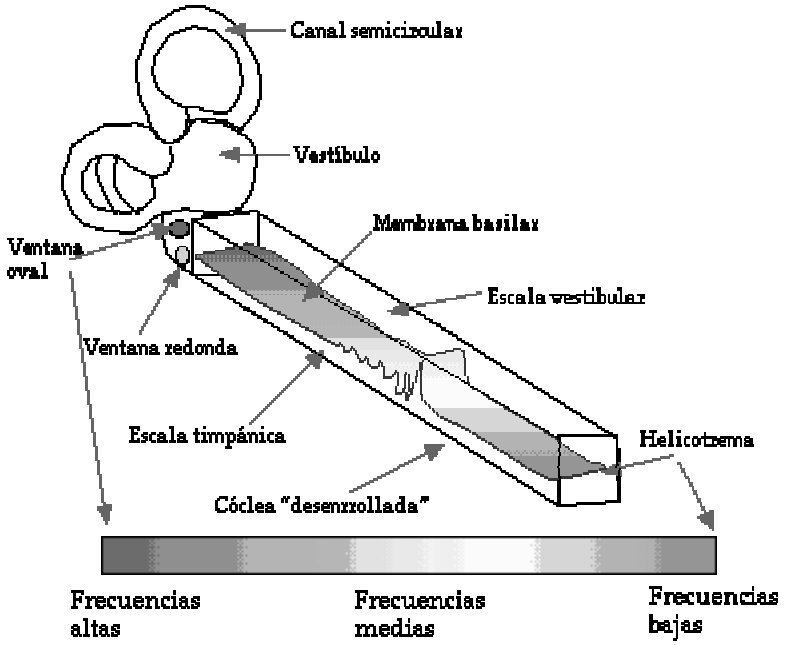


Figura A-3.2.

En este modo de propagación, las ondas de presión son ondas viajeras, en las cuales (a diferencia de las ondas estacionarias) no existen nodos. En la Figura A-3.3 se observa la amplitud de oscilación de la membrana basilar en dos instantes de tiempo, junto con la envolvente de la onda viajera, en función de la distancia al estribo.

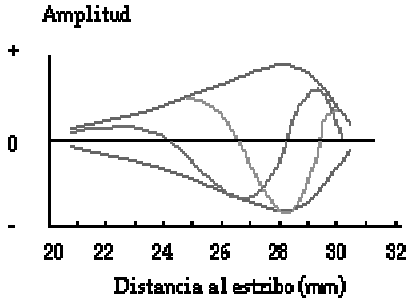


Figura A-3.3.

La ubicación del máximo de la envolvente de la onda viajera depende de la frecuencia de la señal sonora, como puede observarse en la Figura A-3.4 mientras menor es la frecuencia del tono, mayor es la distancia que viaja la onda a lo largo de la membrana antes de ser atenuada, y viceversa. De esta forma, la membrana basilar dispersa las distintas componentes de una señal de espectro complejo en posiciones bien definidas respecto a la ventana oval.

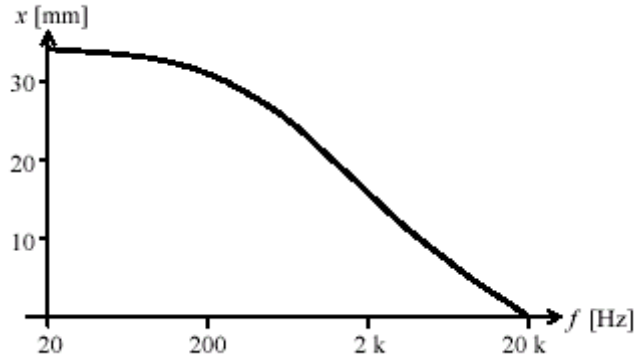


Figura A-3.4a Transformación frecuencia/posición (de resonancia) a lo largo de la membrana basilar.

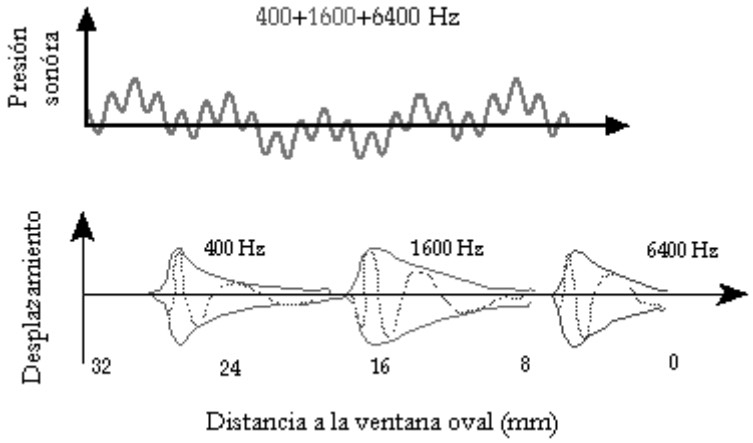


Figura A-3.4b Representación esquemática de la envolvente de una onda viajera que ocurre en respuesta a una señal compuesta por la suma de tres tonos.

Como resultado de la transformación de frecuencia - posición, la cóclea se puede ver como un banco de filtros pasabanda, con bandas superpuestas. Las respuestas de la magnitud son asimétricas y no lineales, más aún, tienen un ancho de banda no uniforme, que aumentan con la frecuencia. El ancho de banda crítica es una función de la frecuencia que cuantiza los filtros pasabanda de la cóclea. Esto conduce al concepto de bandas críticas (*BBCC*).

Por debajo de los 500 Hz[3], el ancho de banda crítico es aproximadamente constante igual a 100 Hz, mientras que por encima de los 500 Hz crece en proporción a la frecuencia. El ancho de una banda crítica (BC) centrada en una frecuencia superior a 500 Hz es de alrededor del 20% de su frecuencia central. La ecuación (A-3.3), permite calcular el ancho de banda crítico, correspondiente a la frecuencia *f*, con un error inferior al 10%:

$$BW_c(f) = 25 + 75(1 + 1.4(f/1000)^2)^{0.69} \text{ (Hz)} \qquad \text{(A-3.3)}$$

Aunque la función es continua, es útil construir sistemas prácticos para tratar el oído como sistema discreto de filtros pasabanda. Basándose en los valores obtenidos mediante la

ecuación A-3.3, es posible subdividir el rango de frecuencias audibles en intervalos adyacentes de una banda crítica de ancho, que no se solapan entre sí. Esta subdivisión se representa en la figura A-3.5. En el rango audible de 20 Hz a 20 kHz se encuentran 25 bandas críticas adyacentes, numeradas en forma consecutiva en la figura.

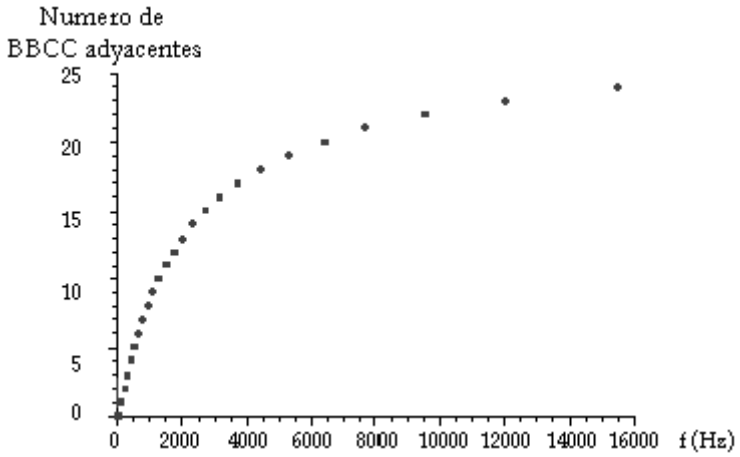


Figura A-3.5. Cantidad de bandas críticas en función de la frecuencia.

En la Tabla A-3.1. se muestran los valores que definen las primeras 24 bandas críticas, los cuales se han convertido en un estándar para describir la distribución de las bandas en función de la frecuencia.

Nº de banda crítica	1	2	3	4	5	6	7	8	9	10	11	12
Frec. Central (Hz)	50	150	250	350	450	570	700	840	1000	1170	1370	1600
Frec. Superior (Hz)	100	200	300	400	510	630	770	920	1080	1270	1480	1720
Ancho de la BC(Hz)	100	100	100	100	110	120	140	150	160	190	210	240

Nº de banda crítica	13	14	15	16	17	18	19	20	21	22	23	24
Frec. Central (Hz)	1850	2150	2500	2900	3400	4000	4800	5800	7000	8500	10500	13500
Frec. Superior (Hz)	2000	2320	2700	3150	3700	4400	5300	6400	7700	9500	12000	15500
Ancho de la BC(Hz)	280	320	380	450	550	700	900	1100	1300	1800	2500	3500

Tabla A-3.1.

Ahora supóngase que se subdivide de manera continua el rango de frecuencias audibles en intervalos solapados entre sí de una BC de ancho[6]. Se desea obtener para cada frecuencia f_0, un valor que represente el número (no necesariamente entero) de bandas críticas adyacentes y no solapadas contenidas en el intervalo de 0 a f_0 Hz. Los valores así obtenidos constituyen la denominada tasa de BBCC. La tasa de BBCC y el ancho de las mismas están relacionados a través de la ecuación A-3.4.

$$Z(f) = \int_0^f \frac{1}{\Delta f_{BC}(x)} dx \quad \text{(A-3.4)}$$

Para los valores de tasa de BBCC, se ha definido como unidad el "Bark": un intervalo de frecuencia de 1 Bark es, por definición, un intervalo de una BC de ancho en cualquier punto del rango de frecuencias audibles. La relación entre la tasa de BBCC y la frecuencia puede ser expresada mediante la ecuación (A-3.5) la cual permite calcular la tasa de BBCC (en Barks), *Z(f)*, correspondiente a la frecuencia en Hz, *f*, con un error inferior a ± 0,2 Barks:

$$Z(f) = 13\arctan(0.00076 f) + 3.5\arctan((f/7500)^2) \text{ (Barks)} \qquad \text{(A-3.5)}$$

Las BBCC y su escala asociada están relacionadas estrechamente con diversos fenómenos fisiológicos y psicoacústicos. Por una parte, los intervalos de una BC de ancho corresponden a distancias iguales a lo largo de la membrana basilar, medidas en sentido longitudinal (desde la ventana oval hacia el helicotrema). Cada BC representa una distancia de 1,3 mm.

Puesto que los receptores auditivos están distribuidos de manera equidistante a lo largo de la membrana, cada BC corresponde por lo tanto a un número constante de receptores; en consecuencia, un número Z_0 de bandas críticas, que representa un intervalo de Z_0 Barks, equivale a una distancia de 1.3 Z_0 mm. Por ende, la función de tasa de bandas críticas en términos de la frecuencia puede interpretarse como una función que indica la relación entre la frecuencia de la señal sonora y su posición asociada en la membrana basilar.

Para ilustrar lo expuesto anteriormente en la figura A-3.6 se observan los patrones de excitación producidos por bandas de ruido estrechas, centradas en distintas frecuencias y con la misma intensidad total (60 dB SPL), expresados en función de la tasa de BCs.

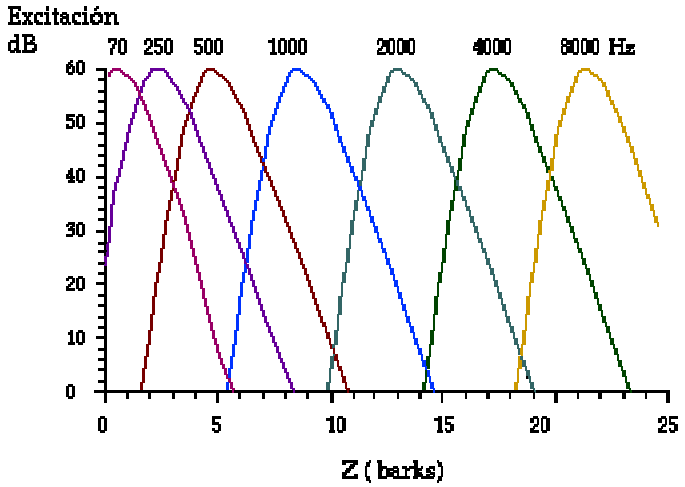


Figura A-3.6. Patrones de excitación de bandas estrechas de ruido centradas en distintas frecuencias, en función de z.

A-3.4 Emascaramiento[3]

A-3.4.1 Enmascaramiento simultáneo o en frecuencia

El enmascaramiento se refiere al proceso donde un sonido se hace inaudible debido a la presencia de otro sonido. El enmascaramiento simultáneo puede ocurrir siempre que dos o más estímulos se presenten simultáneamente en el sistema auditivo. Desde un punto de vista de la frecuencia, las formas relativas de los espectros de las señales enmascaradoras y enmascaradas determinan en qué medida la presencia de cierta energía espectral enmascarará la presencia de la otra energía espectral.

Una explicación simplificada de este fenómeno es que la presencia de un tono o ruido enmascarador fuerte, crea una excitación lo suficientemente fuerte en la membrana basilar, sobre determinada banda crítica que bloquea la correcta detección de una o más señales débiles. Aunque los espectros de señales de audio pueden tener panoramas de enmascaramiento simultáneo complejos, para nuestros propósitos de modelar distorsiones en la codificación es conveniente distinguir solamente tres tipos de enmascaramiento simultáneo, ellos son:

- Ruido enmascara tono (NMT)
- Tono enmascara ruido (TMN)
- Ruido enmascara ruido (NMN)

NMT: En la figura A-3.7. un ruido de banda angosta (por ejemplo, 1 Bark de ancho de banda) enmascara a un tono dentro de la misma banda crítica, donde la intensidad del tono enmascarado está por debajo del umbral relacionado con la intensidad y la frecuencia central de la banda de ruido. La diferencia más pequeña entre la intensidad del ruido y la intensidad del tono ocurre cuando la frecuencia del tono enmascarado es cercana a la frecuencia central de la banda de ruido. En este caso la relación señal a mascara (SMR) es mínima.

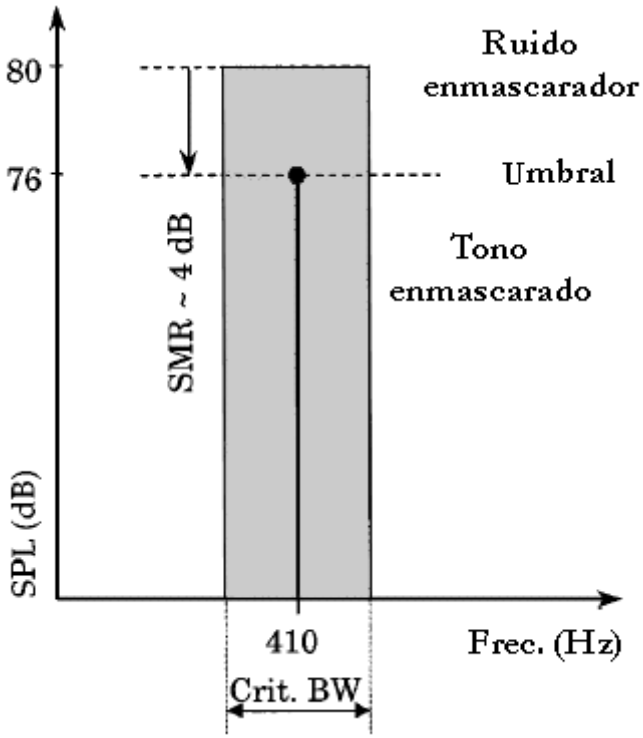


Figura A-3.7.

En la figura A-3.7. una banda crítica de ruido centrada en 410 Hz con una intensidad de 80 dB SPL enmascara un tono de 410 Hz, y la SMR en el umbral de detección es de 4 dB. Si la frecuencia del tono cambia, la energía de la mascara disminuye, es decir, aumenta SMR.

TMN: Un tono puro puede enmascarar ruido de cualquier ancho de banda o forma siempre y cuando el espectro de este esté por debajo de un umbral de enmascaramiento. El mismo se halla directamente relacionado con la intensidad y la frecuencia del tono enmascarador. La SMR mínima ocurre cuando la frecuencia del tono enmascarador está cerca de la frecuencia central del ruido de prueba. El valor mínimo de SMR está entre 21 y 28 dB.

En la figura A-3.8. un ruido de banda angosta centrada en 1 kHz es enmascarado por un tono de 1 kHz de 80 dB SPL de intensidad. La SMR que resulta del umbral de detección es de 24 dB. Al igual que en NMT, el poder de enmascaramiento decrece si la frecuencia central de la banda del ruido se corre sobre o debajo de la frecuencia del tono enmascarador.

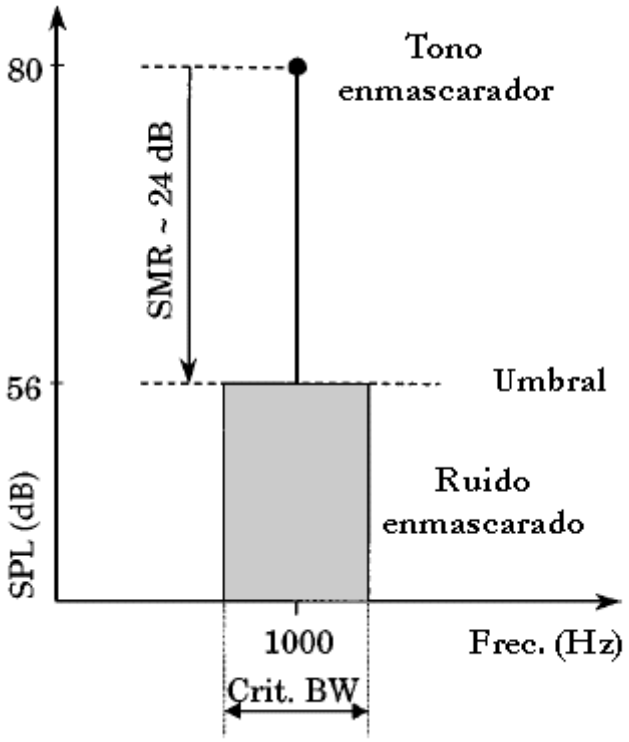


Figura A-3.8.

NMN: Este efecto, cuando un ruido de banda angosta enmascara a otro ruido de banda angosta, es más difícil de caracterizar que NMT o TMN debido a la influencia de las relaciones de la fase entre la señal enmascaradora y la enmascarada. Esencialmente, diversas fases relativas entre los componentes de cada uno pueden conducir a diversos umbrales. Los resultados a partir de un estudio de los umbrales de detección produjeron una SMR de casi 26 dB para NMN.

Enmascaramiento por dispersión: Según lo referido anteriormente, los efectos de enmascaramiento simultáneos caracterizados por los modelos NMT, TMN, y NMN no están limitados dentro de una sola banda crítica. Es decir, un enmascarador centrado en una banda crítica tiene cierto efecto en los umbrales de detección en otras bandas críticas. Este efecto, también es conocido como enmascaramiento por dispersión, y usualmente se modela por una función triangular que tiene pendiente de +25 y –10 dB por Bark. Analíticamente podemos expresar la función de dispersión como:

$$SF_{dB}(x) = 15{,}81 + 7{,}5(x + 0{,}474) - 17{,}5\sqrt{1 + (x + 0{,}474)^2} \text{ dB} \quad \text{(A-3.6)}$$

Donde x tiene unidades de Barks y SF se expresa en dB.

Después que el análisis de bandas críticas fue hecho y que el enmascaramiento por dispersión fue considerado, los umbrales de enmascaramiento en los codificadores perceptivos son establecidos por las siguientes relaciones en dB:

$$TH_N = E_T - 1.45 - B \quad \text{(A-3.7)}$$
$$TH_T = E_N - K \quad \text{(A-3.8)}$$

Donde:

- $\boldsymbol{TH_N}$ **y** $\boldsymbol{TH_T}$ son los umbrales de enmascaramiento del ruido y del tono, respectivamente.
- $\boldsymbol{E_N}$**y** $\boldsymbol{E_T}$ son los niveles de energía enmascaradora de las bandas de ruido y de los tonos respectivamente.
- $\boldsymbol{B}$ es el número de bandas críticas.

Dependiendo del algoritmo, el parámetro $\boldsymbol{K}$ se ha fijado típicamente entre 3 y 5 dB.

TH_N y TH_T se refieren solamente a las contribuciones individuales del ruido o el tono como enmascarantes. Después de que se hayan identificado, se combinan para formar un umbral de enmascaramiento global. Este umbral abarca una estimación del nivel en el cual el ruido de cuantización es más sensible. Por lo tanto, el umbral de enmascaramiento global se refiere a veces como el nivel de distorsión apenas notable (JND).

Es costumbre en la codificación perceptiva primeramente clasificar señales que enmascaran, como ruido o tono. Luego, computar los umbrales apropiados, para entonces usar esa información para modelar el espectro del ruido para que este quede debajo del JND. Observamos que el umbral absoluto (T_q) de la audición también está considerado al formar los espectros del ruido, y que el Máximo(JND, T_q) es utilizado como el umbral de distorsión.

Las nociones del ancho de banda crítica y del enmascaramiento simultáneo dan lugar a una cierta terminología que ilustraremos en la figura A-3.9. donde consideramos el caso de un solo tono enmascarador en el centro de una banda crítica.

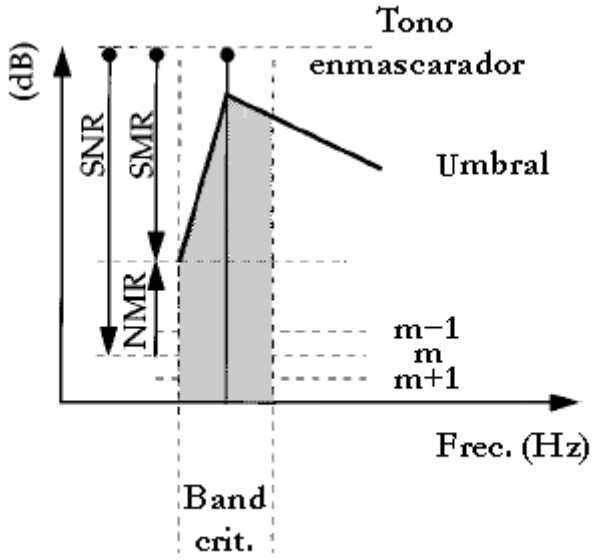


Figura A-3.9.

El tono enmascarador genera una excitación a lo largo de la membrana basilar, la cual es modelada por la función de dispersión y el umbral de enmascaramiento correspondiente. Para la banda bajo consideración, el umbral de enmascaramiento mínimo esta denotado por la función (A-3.6). Si se asume que la señal enmascaradora es cuantizada con un cuantizador uniforme de m bits, este puede introducir ruido hasta el nivel m.

SMR y NMR (relación ruido a mascara) denotan las distancias del umbral de enmascaramiento mínimo al tono enmascarador y al nivel de ruido, respectivamente. La relación señal a ruido (SNR), es la suma de estas dos.

A-3.4.2 Emascaramiento no simultáneo o temporal

Como se muestra en la figura A-3.10. los fenómenos del enmascaramiento se extienden en el tiempo más allá de la persistencia del estímulo. Es decir para una señal enmascaradora de duración finita, este tipo de enmascaramiento ocurre antes de que aparezca esta señal, así como también persiste una vez que desaparece. Básicamente, lo que sucede es que los umbrales absolutos de la audición para sonidos enmascarados son aumentados antes, durante, y después de la ocurrencia del estimulo enmascarador.

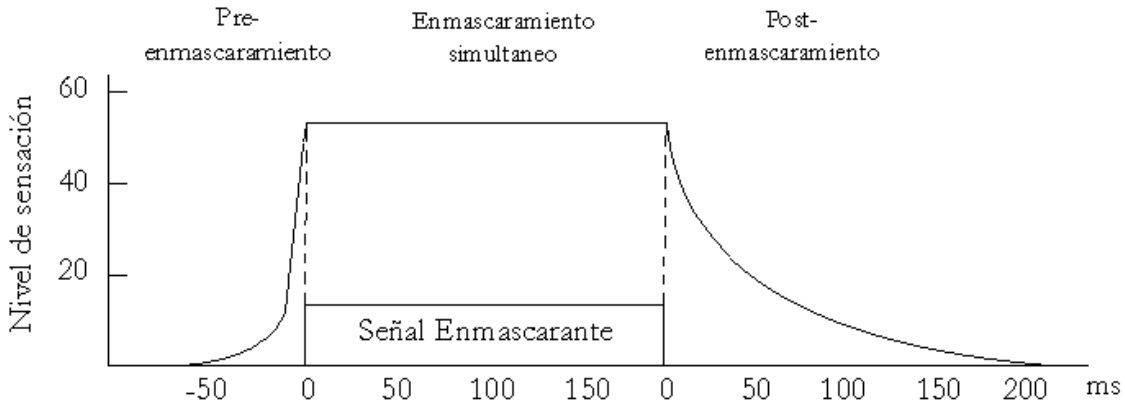


Figura A-3.10. Representación esquemática de enmascaramiento simultáneo.

El pre - enmascaramiento se extiende hasta 50 ms antes de la aparición de la señal enmascaradora, sin embargo es realmente notorio hasta 2 ms antes. En cuanto al post – enmascaramiento, este se podrá extender hasta 300 ms, dependiendo de la intensidad y duración de la señal enmascaradora. Consideramos aquí las características más importantes del enmascaramiento no simultáneo que se deben aplicar a los modelos perceptuales del codificador de audio.

Para un estímulo enmascarador y un tono de prueba de la misma frecuencia, se ha demostrado por medio de estudios experimentales que el post - enmascaramiento depende de manera bastante predecible de la frecuencia, la duración y la magnitud del estímulo. El post - enmascaramiento también exhibe un comportamiento dependiente de la frecuencia similar al del enmascaramiento simultáneo, que puede ser observado cuando se varía la relación de la frecuencia de la señal enmascaradora y la enmascarada.

En la figura A-3.10. notamos que el pre - enmascaramiento decae mucho más rápido que el post – enmascaramiento. Por ejemplo, un estudio demostró que solamente 2 ms antes del inicio de la señal enmascaradora, el umbral de enmascaramiento era de 25 dB por debajo del umbral de enmascaramiento simultáneo. A pesar de resultados inconsistentes encontrados a través de estudios, se acepta generalmente que la cantidad del pre - enmascaramiento medido depende del entrenamiento que hayan tenido los sujetos a los que se les realiza el experimento. Para los propósitos de la codificación perceptiva, las señales de audio (por ejemplo: el inicio de un instrumento musical de percusión) crean regiones en el tiempo de pre y post - enmascaramiento durante el cual un oyente no

percibirá señales debajo de los umbrales de audición, estos elevados debido a la señal enmascaradora.

Apéndice A-4

MP3

A-4.1 Banco de filtros polifáse[1].

El banco de filtros divide la señal de audio en 32 subbandas de frecuencias igualmente espaciadas. Los filtros son relativamente simples y ofrecen una buena resolución temporal con una razonable resolución en frecuencia.

Para el diseño de estos filtros se deben de tener en cuenta algunas consideraciones. Primero, la igualdad del ancho de las subbandas no representa adecuadamente las bandas críticas del oído. Muchos efectos psicoacústicos son consistentes si tomamos un escalado de frecuencias como el de las BBCC. Por ejemplo, la audición de una señal en presencia de una señal enmascaradora es diferente para señales que están dentro de una BC que para señales que están más allá de una BC. Para bajas frecuencias una subbanda abarca varias BBCC. En estas condiciones el número de bits de cuantización no puede ser explícitamente fijado por el ruido enmascarador disponible por cada una de las BBCC. En cambio, la BC con el menor enmascaramiento de ruido nos da el número de bits de cuantización necesitados para toda la subbanda.

En segundo lugar, el banco de filtros polifáse y su inversa son transformaciones con pérdidas. Incluso sin cuantización, la transformación inversa no puede recuperar perfectamente la señal original. Sin embargo, el error introducido por el banco de filtros polifáse es pequeño e inaudible.

Finalmente, las bandas de dos filtros adyacentes tienen un gran solapamiento de frecuencia, esto es, un tono puede generar una salida en dos subbandas adyacentes.

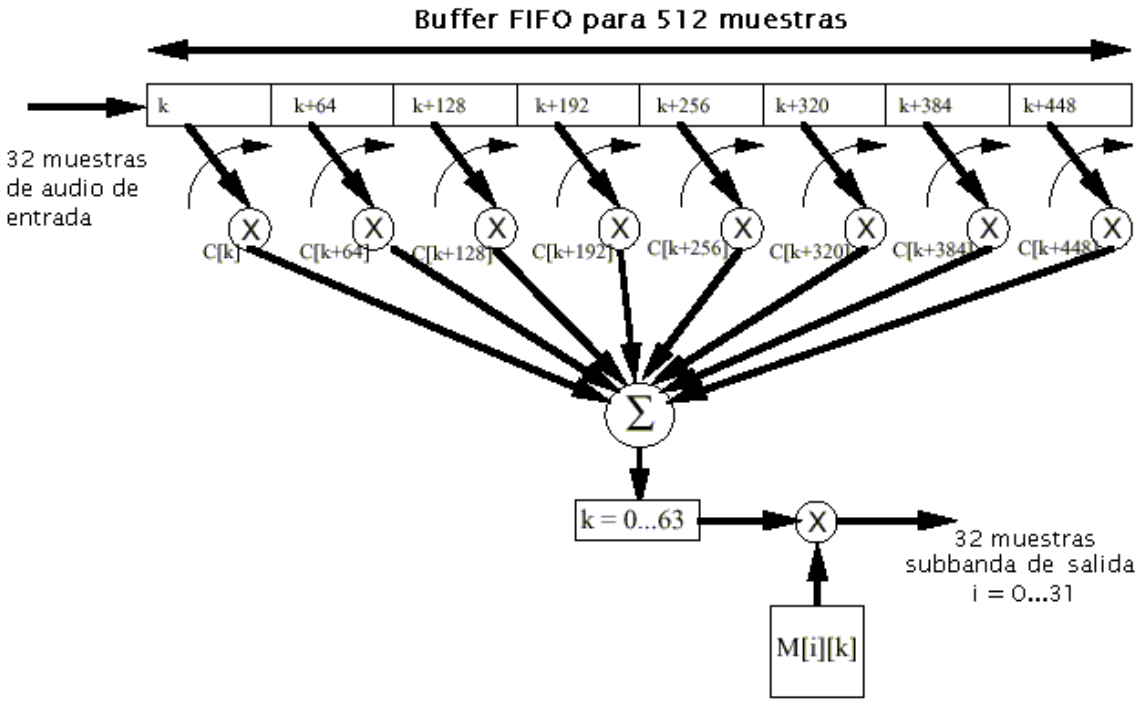


Figura A-4.1 Detalle de un banco de filtros polifásicos.

Fuente: PAN, Davis. A tutorial on MPEG/Audio compression. En: IEEE Multimedia Journal. Vol. 2 No. 2 (Summer 1995).

La salida del filtro mostrado en la figura A-4.1. es:

$$s_t(i) = \sum_{k=0}^{63} \sum_{j=0}^{7} M[i][k] \times (C[k+64j] \times x[k+64j]) \qquad \text{(A-4.1)}$$

donde: i es el índice de la subbanda, los márgenes van desde 0 hasta 31.
$s_t(i)$ es la muestra de salida del filtro para la subbanda i en un tiempo t.
t es un entero múltiplo de los 32 intervalos de muestras de audio.
$C[n]$ es uno de los 512 coeficientes de la ventana de análisis definida en el estándar.
$x[n]$ es la muestra de audio de entrada extraída de un buffer de 512 muestras.
$M[i][k] = \cos\left[(2i+1) \times (k-16) \times \pi/64\right]$ es la matriz de coeficientes de análisis.

La ecuación A-4.1. esta parcialmente optimizada para reducir el número de operaciones. Debido a que la función, que está entre paréntesis, es independiente del valor de i, y *M*[i][k] es independiente de j, las 32 salidas de los filtros necesitan solo 512 + 32 x 64 = 2560 productos y 64 x 7+32 x 63 = 2464 sumas, o aproximadamente 80 productos y sumas por cada salida. Sin embargo se puede mejorar sustancialmente el número de operaciones mediante una transformada discreta del coseno rápida(FDCT), o la FFT.
Podemos modificar la ecuación 5.1. utilizando la típica ecuación de convolución:

$$S_t[i] = \sum_{n=0}^{511} x[t-n] \times H_i[n] \qquad \text{(A-4.2)}$$

donde: x[t] es una muestra de audio en un tiempo t.

$$H_i[n] = h[n]\cos\left[(2i+1) \times (k-16) \times \pi/64\right] \qquad \text{(A-4.3)}$$

con h[n] = -C[n], si la parte entera de (n/64) es impar, o h[n] = C[n] en el caso contrario, para n = 0 hasta 511.

De esta forma cada subbanda del banco de filtros tiene su propio filtro pasa banda correspondiente a la respuesta al impulso $H_i[n]$. A pesar de que esta forma es muy conveniente para el análisis matemático, es claramente, una solución ineficiente para ser implementada. Una implementación directa de esta ecuación requiere 32 x 512 = 16384 multiplicaciones y 32 x 511 = 16352 sumas para obtener las 32 salidas de los filtros. Podemos observar que en la ecuación A-4.3. se modula la señal h[n] mediante el producto con un coseno, de esta forma obtenemos un desplazamiento en frecuencia de la señal h[n], es por este motivo que se denominan filtros polifáse.

A pesar de que el banco de filtros polifáse tiene pérdidas, los errores que de éste se obtienen son pequeños.

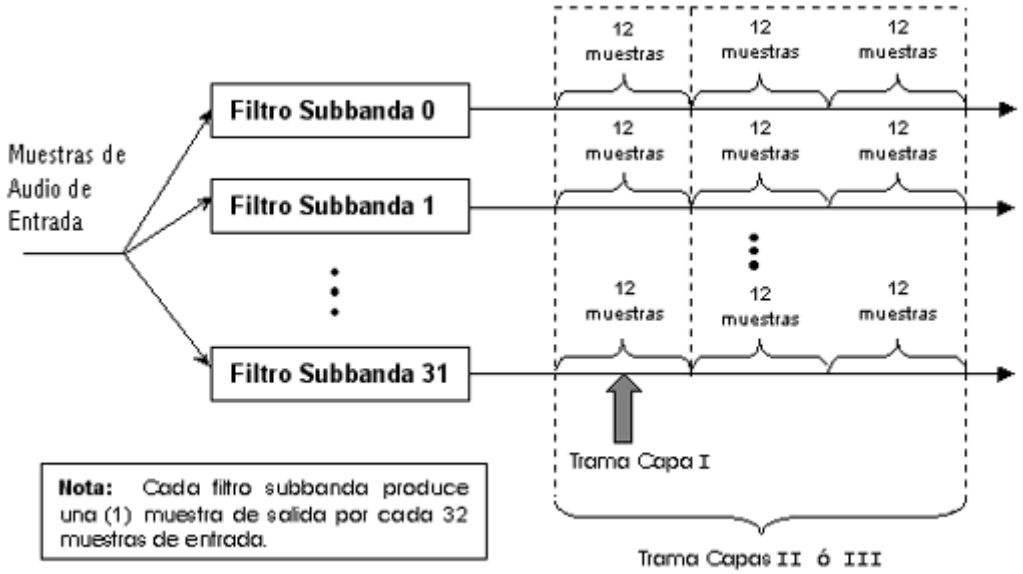


Figura A-4.2 División en subbandas de las muestras de audio.

A la salida del banco de filtros polifásico, las muestras de audio se dividen por subbandas de la manera mostrada en la figura A-4.2 Como se ve, cada subbanda aporta 12 muestras para un total de 384 muestras de audio, en la Capa 1, mientras que para la Capa 3, cada subbanda aporta 36 muestras de audio para un total de 1152 muestras subbanda por trama.

A-4.2 Repartición de Bits

El modelo psicoacústico calcula el umbral de enmascaramiento, para cada subbanda, por debajo del cual el ruido de cuantización es imperceptible para el oído humano. En el proceso de repartición de bits, estos son asignados a las subbandas de acuerdo al nivel de enmascaramiento que resultó de aplicar el modelo psicoacústico. El objetivo de la repartición de bits es minimizar el máximo de las relaciones mascara - ruido, el máximo tomado sobre todos los canales y todas las subbandas. Para las capas 1 y 2, este proceso empieza calculando la relación ruido - mascara (NMR) dada por la siguiente ecuación:

NMRdB = SNRdB - SMRdB (todos los valores en decibeles)

La SNR puede hallarse por distintos métodos pero el estándar MPEG/audio provee tablas que dan estimaciones de la misma, resultante de la cuantización con un número dado de niveles.

Una vez que la unidad que reparte los bits tiene relaciones ruido - mascara para todas las subbandas, busca la subbanda con NMR más baja y le asigna *codebits*. Cuando a una subbanda se le pueden asignar más *codebits*, la unidad de reparto de bits busca la nueva estimación para la SNR y recalcula la NMR de la subbanda. El proceso se repite hasta que no se puedan asignar más *codebits*.

A-4.3 Repartición de ruido (Noise Allocation). [1], [4]

Mientras las capas 1 y 2 usan repartición de bits el codificador de la capa 3 usa repartición de ruido. La repartición de bits únicamente aproxima la cantidad de ruido causado por la cuantización, mientras que la repartición de ruido verdaderamente calcula el ruido. La repartición se hace en un ciclo de iteración que consiste de un ciclo interno y uno externo.

Ciclo interno (rate control loop).

El ciclo interno realiza la cuantización no uniforme y escoge un determinado paso de cuantización, cuantiza los valores espectrales, y a estos datos cuantizados se les aplica codificación de Huffman. Si el número de bits resultante de la codificación excede el número de bits disponible para codificar un bloque de datos dado, de acuerdo con el *bit - rate* escogido, se realiza una corrección ajustando la ganancia global. De este modo se logra tener un paso de cuantización más grande, dando así valores cuantizados más pequeños, entonces el ciclo comienza otra vez con un nuevo intervalo de cuantización, ejecutando la cuantización y la codificación de Huffman otra vez. El ciclo termina cuando los valores cuantizados que han sido codificados con Huffman usan menor o igual número de bits que la máxima cantidad de bits permitida.

Ciclo externo (distortion control loop).

Para moldear el ruido de cuantización de acuerdo al umbral de enmascaramiento (colorearlo para que se adecue a los contornos de frecuencia variable), factores de escalas son aplicados a cada banda de factor de escala. que se adecue a los contornos de frecuencia variable del umbral de enmascaramiento. El sistema comienza con un factor por defecto de 1.0 para cada banda. Si el ruido de cuantización en una determinada banda excede el umbral de enmascaramiento (ruido permitido) obtenido del modelo psicoacústico, el factor de escala para esta banda se ajusta para reducir el ruido de cuantización. Ahora el ciclo externo se encarga de verificar si el factor de escala para cada subbanda tiene más distorsión que la permitida (ruido en la señal codificada), comparando cada banda del factor de escala (*scalefactor band*) con los datos previamente calculados en el análisis psicoacústico. Si cualquiera de las bandas del factor de escala tiene más ruido que el máximo permitido, el ciclo amplifica esa banda de factor de escala, decrementa el tamaño del paso del cuantizador para las mismas y ejecuta ambos ciclos (el interno y el externo) de nuevo. El ciclo externo termina cuando una de las siguientes condiciones se cumple:

- Ninguna de las bandas del factor de escala tiene una distorsión mayor a la permitida.
- La próxima iteración amplificaría cualquiera de las bandas por encima del valor máximo permitido.
- Todas las bandas han sido amplificadas al menos una vez.

Ya que el ciclo consume mucho tiempo, una aplicación en tiempo real debe tener en cuenta una cuarta condición, que detenga el ciclo evitando que la codificación se ejecute fuera de tiempo.

Dado que lograr ruido de cuantización menor requiere un número de pasos de cuantización mayor y por tanto una *bit - rate* más alta, el ciclo interno debe repetirse cada vez que se usen nuevos factores de escala En otras palabras, el ciclo interno se anida dentro del ciclo externo. El ciclo externo es ejecutado hasta que el ruido real (calculado a partir de la resta entre los valores espectrales originales y los valores espectrales cuantizados) está por debajo del umbral de enmascaramiento para cada factor de escala (i.e. banda crítica).

A-4.4 Otras mejoras de la capa 3. [4]

El último bloque en el proceso de codificación es el encargado de producir un flujo de bits MP3 válido. Este bloque almacena el audio codificado y algunos datos adicionales en tramas, donde cada trama contiene información de 1152 muestras de audio. Una trama es un bloque de datos con su propio encabezado e información de audio junto con el chequeo de errores y los datos auxiliares, estos dos últimos opcionales. El encabezado describe, entre otros, cuál capa, tasa de bits y frecuencia de muestreo se están usando para el audio codificado. Los datos codificados con Huffman y su información secundaria están localizados en la parte de los datos de audio, donde la información secundaria dice qué tipo de bloque, tablas de Huffman y factores de ganancia deben ser usados.

A-4.4.1 Cuantización no uniforme[7]

El cuantizador eleva su entrada a la ¾ potencia antes de cuantizar, de esta manera se busca tener una mayor consistencia de los valores de SNR sobre el rango de cuantización. El decodificador realiza el proceso inverso. La ecuación completa para el cuantizador es:

$$x_q[i] = Rnd\left[\left(\frac{x[i]}{2^{\Delta/4}}\right)^{3/4} - 0.0946\right] \qquad \text{(A-4.4)}$$

Donde: Δ es el paso de cuantización
$x[i]$, es la señal a cuantizar.

El máximo valor de cuantización permitido tiene como fin limitar el tamaño de las tablas usadas para la búsqueda del decodificador.

A-4.4.2 Codificación de Huffman (codificación entrópica).

El MP3 también emplea la clásica técnica del algoritmo de Huffman. Actúa al final de la compresión para codificar la información; por lo tanto, no es un algoritmo de compresión, sino más bien un método de codificación.

Esta técnica crea códigos de longitud variable sobre un número total de bits, donde los símbolos con más alta probabilidad tienen códigos más cortos.
Los códigos de Huffman tienen la propiedad de poseer un único prefijo y por lo tanto, pueden ser decodificados correctamente a pesar de su longitud variable; el proceso de la decodificación es muy rápido, a través de una tabla de correspondencias. Este tipo de

codificación permite ahorrar, en promedio, aproximadamente un 20% en espacio de almacenamiento.
Las técnicas que se han mostrado son el complemento ideal para la codificación psicoacústica: durante gran polifonía, muchos sonidos están enmascarados o disminuidos, logrando que la codificación psicoacústica sea muy eficiente; y debido a que hay poca información idéntica, entonces el algoritmo de Huffman presenta poca eficiencia. Pero durante los sonidos "puros" hay muy pocos efectos de enmascaramiento, y es aquí donde la codificación de Huffman se vuelve muy eficiente debido a que los sonidos puros, cuando se digitalizan, contienen gran cantidad de bytes redundantes, que entonces serán reemplazados por códigos más cortos.

A-4.4.3 Reserva de Bits.

En el caso de las Capas 1 ó 2, las tramas son elementos totalmente independientes, así que se puede extraer cualquier fragmento de datos del archivo MPEG y decodificarlo correctamente. Sin embargo, en el caso de la Capa 3, las tramas no son siempre totalmente independientes.

Debido al posible uso del *bit reservoir*, que es una especie de búffer, las tramas son a menudo dependientes unas de otras. En el peor caso, se pueden necesitar hasta nueve tramas, antes de poder realizar la decodificación de una sola. La figura A-4.3 muestra con un ejemplo este concepto.

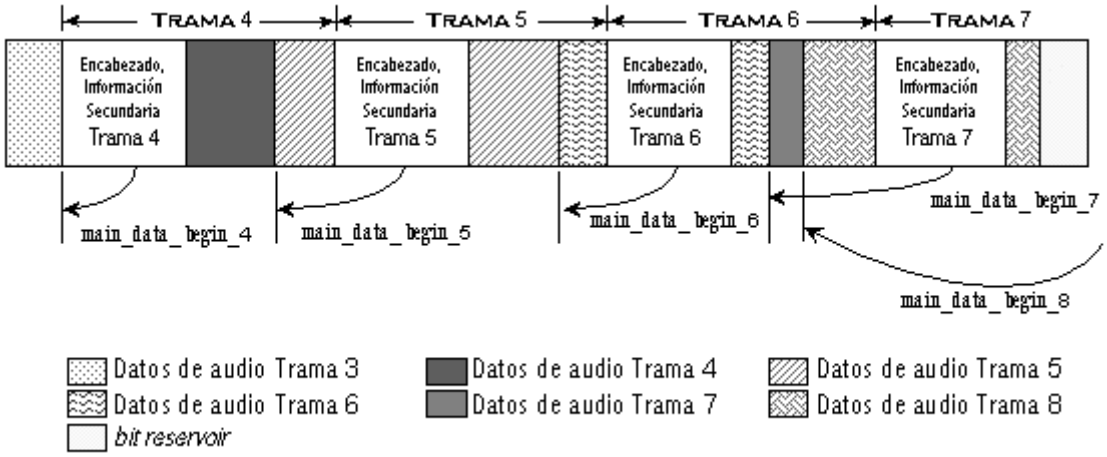


Figura A-4.3 Ejemplo de uso de la reserva de bits (*bit reservoir*)

El *main_data_begin* es un puntero (de nueve bits) de ajuste negativo, incluido dentro de la información secundaria, que apunta a la posición del byte de inicio de la información de audio dentro de cada trama. Por ejemplo, *main_data_begin_4* es igual a cero, indicando que los datos de audio empiezan inmediatamente después de la información secundaria. Para indicar que el audio de la trama 5 se inicia en la trama 4, se especifica *main_data_begin_5* como un ajuste negativo que indica el desplazamiento en bytes hacia la izquierda para encontrar el primer dato de audio de la trama 5.

En el ejemplo se ve como cada trama permite el uso del *bit reservoir*. En el caso de la trama 7, el proceso empieza codificando la información de audio de su propia trama, como los datos requieren muy pocos bits, y la trama 6 tenía espacio disponible, entonces todos los

datos de audio de la trama 7 se incluyen en la trama 6, pero la trama 6 sigue con espacio para *bit reservoir*, que se usa para datos de la trama 8; por lo que gracias al *bit reservoir*, la trama 6 incluye los datos de audio de tres tramas: las tramas 6, 7 y 8. El audio de la trama 8 se reparte entre las tramas 6 y 7; sin embargo, éste no alcanza a ocupar todo el espacio disponible en la trama 7, así que el *bit reservoir* de la trama 7 se usa para la trama 9, y así sucesivamente, teniendo en cuenta que los datos de audio de una determinada trama no pueden estar desplazados más de nueve tramas.

Este caso puede ocurrir en una señal de audio MPEG-1 estéreo, si la frecuencia de muestreo es 48 kHz. y la tasa de transferencia deseada es 32 Kbps. En este caso, cada trama consume 768 bits, donde 304 bits (32 bits para el encabezado, 16 bits para el chequeo de errores, 256 bits para la información secundaria) son fijos. Por lo tanto, quedan 464 bits disponibles para los datos codificados con Huffman, y debido a que el valor de *main_data_begin* puede apuntar máximo 511 bytes (4088 bits) hacia atrás, entonces es posible que *main_data_begin* apunte sobre más de ocho tramas (no se cuenta ninguno de los bits usados para el encabezado y la información secundaria de ninguna trama).

También es importante mencionar que el *bit reservoir* sólo puede originarse de tramas que ya han sido codificadas; para este búffer no es posible usar tramas para las que todavía no se haya hecho la repartición de los bits disponibles (repartición de ruido).

El formato que tiene cada trama se muestra en la figura A-4.4, en la cual se puede ver el encabezado de trama que posee 32 bits (cuatro bytes) de longitud; los primeros 12 bits siempre se ponen en '1', se llaman "*FRAME SYNC*", y se usan para sincronización de la trama.

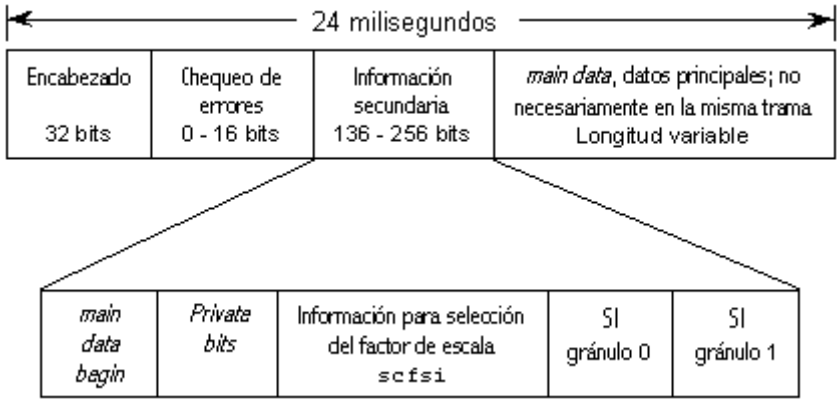


Figura A-4.4 Formato de la trama MP3.

Las tramas pueden tener opcionalmente un CRC para chequeo de errores. Su longitud es de 16 bits, y si existe, se pone después del encabezado. Volviendo a calcular el CRC se puede comprobar si la trama ha sido alterada durante la transmisión del flujo de bits de audio MP3. A continuación sigue la información secundaria (*Side Information*) que indica cómo se realizó la codificación, y por lo tanto, cómo debe realizarse la decodificación. En el último bloque sí vienen los datos de audio (*main data*), repartidos entre dos gránulos.

Dentro de la información secundaria, que usa 136 bits en modo monofónico y 256 bits en los otros modos, se incluye el *main_data_begin*, que es el puntero ya visto. Los bits privados están a disposición del usuario. Después viene la información que indica cuál combinación de factores de escala se está usando (scfsi, *scalefactor selection information*). Los últimos dos sub

- bloques corresponden a la información secundaria (*Side Info*, SI) para cada uno de los dos gránulos (sub - tramas) en los que se divide una trama.

El último bloque, *main data*, es el que lleva la información de audio; las muestras MDCT codificadas con Huffman, repartidas entre dos gránulos. Cada gránulo contiene información de 576 muestras de audio (exactamente la mitad de la información total de la trama). Además, en este mismo bloque se incluyen los factores de escala de la trama y la información auxiliar, siendo esta última opcional.

Apéndice A-5.

Formato de las tramas MP3[5]

El audio en un flujo MPEG-1 está organizado de tal manera que cada fragmento del audio codificado (trama) sea decodificable por sí mismo, con una posible excepción para la Capa 3. La trama está constituida por las muestras de audio y por la información secundaria. Esta última sirve de control, además de proporcionar información del archivo. Para la Capa 3, la trama está constituida por**:** *1 trama = 1152 muestras de audio + información de la trama.*

A-5.1. Encabezado de tramas.

No existe encabezado principal de archivo en el formato de audio MPEG. En éste el encabezado es individual para cada trama (fragmento de archivo). Cuando se quiere leer información de un archivo MP3, usualmente es suficiente 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 *bitrates* variables, donde cada trama posee su propia *bitrate*. 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 (con ayuda del búffer *bit reservoir*) en las partes donde se necesite. El encabezado de la trama tiene la siguiente presentación, figura A-5.1, con las posiciones para cada uno de los 32 bits:

Figura A-5.1 Trama MP3.

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

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

C: *Layer*. Dos bits usados para descripción de la capa. Para identificar cuál esquema (léase capa) fue usado durante la codificación del audio (figura A-5.2).

00	Reservado
01	Capa III
10	Capa II
11	Capa I

Figura A-5.2 Valores posibles para los bits C.

D: *protection_bit*. Un bit de protección. Si está en '0' indica que la trama está protegida por un código de redundancia cíclica para detección de errores. En la mayoría de los archivos MP3 D = 1.

E: *bitrate_index*. Cuatro bits para proporcionar el índice de la tasa de bits, de acuerdo con la tabla de la figura A-5.3.

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).			

Figura A-5.3 Codificación para la tasa de bits (*bits E*).

F: *sampling_frequency*. Dos bits que indican la tasa de muestreo (figura A-5.4).

00	44.1 KHz
01	48 KHz
10	32 KHz
11	Reservado

Figura A-5.4 Codificación para la tasa de muestreo (*bit F*).

G: *padding_bit*. Un bit usado para relleno. Únicamente se usa para frecuencias de 44.1 kHz. Si se usan tramas de 417 bytes de largo no se logra la tasa de transferencia de 128 Kbps. Para solucionar esto este bit se pone a uno y se agrega un byte extra al final de esas tramas para así obtener 128 Kbps.

H: *private_bit*. Un bit para uso privado. Generalmente no se usa.

I: *mode*. Dos bits que indican el modo de canal, tal y como se muestra en la figura A-5.5.

00	*Stereo*
01	*Joint Stereo*
10	*Dual Channel* (2 canales monofónicos independientes)
11	*Single Channel* (1 canal monofónico)

Figura A-5.5 Codificación para el modo de canal (*bits I*).

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 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 3, 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 fórmula:

$$M_i = \frac{\sqrt{2}}{2}\left(L_i + R_i\right) \quad y \quad Si = \frac{\sqrt{2}}{2}\left(L_i - R_i\right) \quad \text{(A-5.1)}$$

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 así**:**

$$L_i = \frac{M_i + S_i}{\sqrt{2}} \quad y \quad R_i = \frac{M_i - S_i}{\sqrt{2}} \qquad \text{(A-5.2)}$$

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**:**

Código para la Capa III		
Código	*Intensity stereo*	*M/S stereo*
00	no	no
01	sí	no
10	no	sí
11	sí	sí

Figura A-5.6 Codificación de la extensión al modo (*bits J*).

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

L: *original/copy*. Un 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 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.

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

Figura A-5.7 Codificación de la información de énfasis (*bits M*).

A-5.2. 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**:**

$$CRC-16 = x^{16} + x^{15} + x^2 + 1 \qquad \text{(A-5.3)}$$

A-5.3. Información secundaria.

Ésta consta de 17 bytes para el modo monofónico, y de 32 bytes en cualquier otro modo. La información que contiene, consiste de cuatro partes**:** el puntero *main_data_begin*, información secundaria para ambos gránulos (*scfsi* y *private_bits*), información secundaria para el gránulo 0, e información secundaria del gránulo 1.

main_data_begin (9)	private_bits (5,3)	scfsi (4,8)	SI gránulo 0 (59,118)	SI gránulo 1 (59,118)

Figura A-5.8 Formato de la información secundaria.

main_data_begin: el campo *main_data* no está necesariamente localizado justo después de la información secundaria. *main_data_begin* es un puntero que usa 9 bits, indicando la localización donde está el primer byte del *main_data* de la trama actual. La localización está

especificada como un desplazamiento negativo en bytes desde el encabezado actual (bytes a la izquierda, antes del primer bit del encabezado).

La información secundaria (SI) común a ambos gránulos se muestra a continuación**:**

private_bits: El número de *private_bits* para la información secundaria depende del número de canales (5 para mono y 3 para estéreo). El número de bits reservados para *private_bits* es definido por el usuario.

scfsi**:** La variable *scfsi* (información para selección del factor de escala) determina si los factores de escala se envían para cada gránulo, o si son comunes para ambos gránulos, por canal. Se transmiten cuatro bits por canal, cada bit perteneciente a un grupo de bandas del factor de escala diferente. Un '0' para un grupo específico de bandas del factor de escala, indica que los factores de escala para ese grupo en particular, se transmiten para cada gránulo. Un '1' indica que se usan los mismos factores de escala para ambos grupos; por lo tanto, sólo se transmiten los factores de escala correspondientes al grupo de bandas del primer gránulo.

Después de la información secundaria para ambos gránulos, sigue la información secundaria para cada gránulo**:**

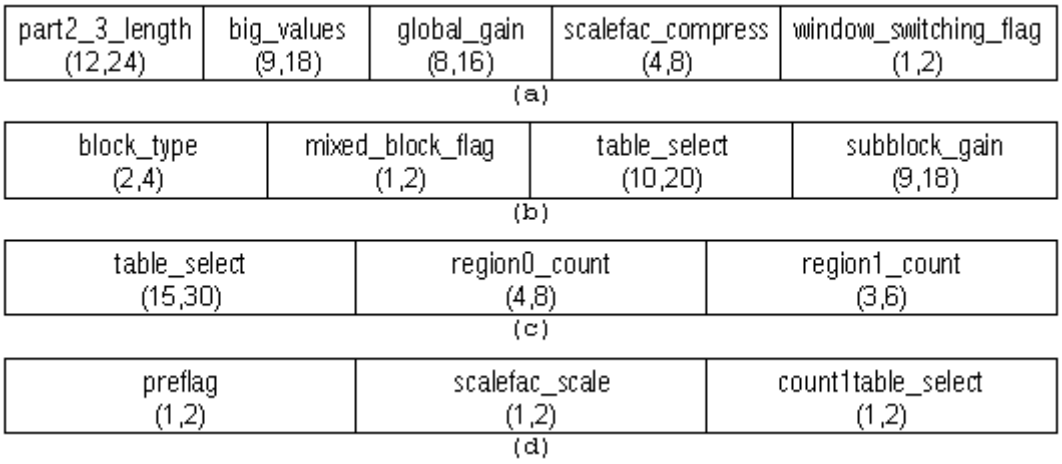


Figura A-5.9 Información secundaria para cada gránulo.

En el caso de bloques largos, la información secundaria para cada gránulo es**:**

part2_3_length: Denota el número de bits que son usados en *main_data* para los factores de escala y los datos codificados con Huffman. Se usan 12 bits en modo mono y 24 en los otros modos. Como la cantidad de bits usados para la información secundaria es constante, *part2_3_length* puede usarse para calcular el comienzo del próximo gránulo.

big_values: Después de la cuantización, las 576 muestras MDCT cuantizadas están organizadas en un orden determinado (de menor a mayor frecuencia). Luego, estos valores se dividen en tres particiones consecutivas: *rzero*, *count1* y *big_values*. La primera partición, *rzero*, se localiza en las altas frecuencias y consiste en pares de ceros. La partición de la mitad, *count1*, consiste de cuádruplos cuyo valor es -1, 0 ó 1. La última partición, *big_values*, se localiza en las bajas frecuencias extendiéndose hasta el nivel de frecuencia de 0 Hz. y se compone de pares de valores restringidos a una amplitud máxima absoluta de 8206 (8191+15, el cual es el máximo valor cuantizado permitido). El campo *big_values* indica la

cantidad de pares cuantizados que pertenecen a esta partición. Nueve bits se usan para *big_values* en modo mono y 18 en los otros modos.

global_gain: Contiene información acerca del intervalo usado en el cuantizador, donde la cuantización se hace logarítmicamente. La variable *global_gain* usa 8 bits en modo mono y 16 bits para los otros modos.

scalefac_compress: Es una variable de 4 bits (en modo mono), transmitida para cada gránulo, la cual determina el número de bits usados para la transmisión de los factores de escala. Cada gránulo se divide en 12 ó 21 bandas del factor de escala dependiendo del tipo de ventana que se esté usando. Estas bandas del factor de escala se dividen de nuevo en dos grupos (0-10 y 11-20 para ventanas largas; 0-5 y 6-11 en el caso de ventanas cortas). La variable *scalefac_compress* se usa como índice a una tabla proporcionada en el estándar ISO 11172-3, la cual retorna dos variables llamadas "slen1" y "slen2", que indican la cantidad de bits usados para los factores de escala del primer y segundo grupo de bandas, respectivamente.

window_switching_flag: Un bit por canal que señala si una ventana diferente del tipo *NORMAL* se está usando. Este valor determina los siguientes 22 bits en la información secundaria: si está en '1', se añaden los bits de la figura A-5.9(b); si está en '0', se añaden los bits de la figura A-5.9(c).

table_select: Habilita el uso de 32 diferentes tablas para el código de Huffman, dependiendo de las estadísticas de la señal. Se usan 15 bits por canal (5 bits por región) para indicar cuáles de las 32 tablas han sido seleccionadas.

region0_count: Para mejorar el desempeño en la codificación, la partición *big_values* se subdivide en tres regiones llamadas region0, region1 y region2. Cada región se codifica con una de las 32 tablas de Huffman (seleccionada con *table_select*). La variable *region0_count* especifica el límite entre region0 y region1. Esta variable de 4 bits (en modo mono) especifica la cantidad de bandas del factor de escala incluidas en esta región, pero disminuidas en 1.
region0_count = bandas del factor de escala en region0 - 1

***region1_count*:** especifica el límite entre region1 y region2. Esta variable de 3 bits por canal indica las bandas del factor de escala incluidas en region1, disminuidas en 1.
region1_count = bandas del factor de escala en region1 - 1

preflag: Un bit por canal, indicando que se usó preénfasis (o sea, amplificación adicional en las altas frecuencias). Este valor apunta a una tabla en el estándar ISO 11172-3, cuyos 21 valores son sumados a los factores de escala. Para bloques cortos, no se usa preénfasis.

scalefac_scale: Los factores de escala están cuantizados de manera logarítmica con un intervalo de 2 ó $(2)^{1/2}$, dependiendo del valor de *scalefac_scale*, que usa 1 bit por canal.

count1table_select: Esta variable, que usa 1 bit por canal, indica cuál de dos posibles tablas de Huffman fue usada para codificar la partición count1.
En el caso de bloques cortos, la información secundaria sólo cambia en las variables mostradas en la figura A-5.9(c), las cuales son reemplazadas por aquellas de la figura A-5.9(b). Las otras variables mostradas en la figura A-5.9 no cambian.

block_type: Indica el tipo de ventana que se usa en un gránulo particular. La variable *block_type* consume 2 bits por canal.

mixed_block_flag: Esta variable, que consume 1 bit por canal, indica que se usan diferentes tipos de ventana en las bajas y en las altas frecuencias. Si esta variable está en '1', las dos subbandas más bajas usan ventana *NORMAL*, y las 30 subbandas restantes usan el tipo de ventana especificado por *block_type*.

table_select: En este caso, *table_select* usa 10 bits por canal, debido a que, para bloques cortos, la partición *big_values* sólo se subdivide en dos regiones.

subblock_gain: Habilita una ganancia por un factor de 4 para un sub - bloque particular. Esta variable usa 3 bits por canal.

A-5.4. Datos principales.

En esta parte del flujo de bits de la Capa 3, están incluidos los campos mostrados en la figura A-5.10**:**

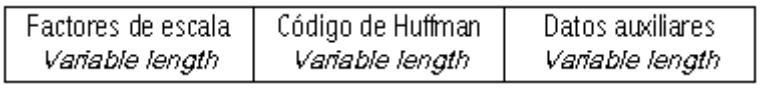


Figura A-5.10 Campos incluidos en los datos principales.

Factores de escala: Éstos se usan para colorear el ruido de cuantización. Los factores de escala se transmiten para cada grupo de líneas de frecuencia (bandas del factor de escala) de cada gránulo, dependiendo del valor de scfsi para ese grupo particular de líneas de frecuencia. La cantidad de factores de escala realmente transmitidos, también depende de *block_type*, *window_switching_flag* y *mixed_block_type*. Los factores de escala consumen entre 0 y 74 bits.

Código de Huffman: Las líneas de frecuencia de cada gránulo se dividen en tres particiones (*rzero*, *count1* y *big_values*). La partición *rzero* no se codifica, ya que sólo contiene valores iguales a cero. La partición *count1* contiene cuádruplos de valores iguales a -1, 0 ó 1, que se codifican usando una de 2 posibles tablas de Huffman, la cual ha sido especificada por *coun1table_select*. Para cada valor diferente de cero, se agrega un bit que indica el signo ('0' si es positivo). La partición *big_values* fue subdividida en tres regiones, las cuales se codifican separadamente, usando una de 32 posibles tablas de Huffman (numeradas de 0 a 31, pero en realidad son 30, ya que las tablas 4 y 14 no existen), o sea, una tabla por región. Dentro de la partición *big_values*, los pares de líneas de frecuencia con valor absoluto menor que 15, se codifican directamente. Para cada valor absoluto mayor o igual a 15, se agregan 1 ó 2 campos extras llamados *"linbitsx"* o *"linbitsy"* dependiendo de cuál es el valor del par *(x,y)* que es mayor o igual a 15. Este campo extra usa de 0 a 13 bits, dependiendo del parámetro *"linbits"*, el cual se calcula con base en el valor máximo de la región, como se muestra en la siguiente fórmula**:**

Linbits = $\log_2$ (máximo valor cuantizado - 14) $\Rightarrow$ Se redondea por exceso.

De nuevo, para cada valor diferente de cero, se agrega bit de signo ('0' si es positivo).
Por ejemplo: Asúmase, primero que la tabla de Huffman ya ha sido seleccionada, y también**:**

Par de valores cuantizados (x,y) = (0,15)
Máximo valor cuantizado de la región = 1039
Código de Huffman para el par (0,15) = '01101'
Valor adicional para 'y' = *linbitsy* = 15-15 = 0

linbits = $\log_2$ (1039 - 14) $\cong$ 10,0014 $\Rightarrow$ *linbits* = 11
linbitsy = 15 - 15 = 0 = '00000000000'
Codificación del par (0,15) = *Codificación del par* (0,15) + *linbitsy*
Codificación del par (0,15) = '01101"00000000000'
bits necesarios para codificar el par (0,15) = 16 *bits*

(x,y)	linbitsx	linbitsy	signx	signy
5 bits	0 bits	11 bits	0 bits	1 bit
Flujo de bits '01101' '00000000000' '0'				

Figura A-5.11 Ejemplo ilustrativo.

En el caso de que 'x' también sea mayor que 14, se debe buscar el código de Huffman para el par (15,15), y además también se debe codificar un valor adicional llamado "*linbitsx*", que indica la diferencia entre 15 (máximo valor de las tablas) y el valor verdadero de 'x'.

Adicionalmente, por cada valor diferente de cero se debe agregar un bit de signo ('0' si es positivo, '1' si es negativo). En el ejemplo, la cantidad total de bits que se necesita para codificar el par es 17 bits, ya que se debe agregar un bit para indicar que 'y' es diferente de cero.

Datos auxiliares: Éstos son opcionales, y la cantidad de bits repartidos para este campo, se define por el usuario.

Listado de Acrónimos.

BC: Banda Crítica
CRC: Cyclic Redundancy Code
DCT: Discrete Cosine Transformation
FA: Falsa aceptación
FDCT: Fast DCT
FFT: Fast Fourier Transformation
FR: Falso rechazo
GMM: Gaussian Mixture Models
ISO: International Standards Organization
JND: Just Noticeable Distortion
MDCT: Modified DCT
MFCC: Mel-Frecuency Cepstral Coefficients
MP3: MPEG-1 Layer 3
MP3CEP: MP3 Cepstrum
MPEG: Motion Pictures Experts Group
MS: Middle Sided
NMN: Noise Masking Noise
NMR: Noise to Mask Ratio
NMT: Noise Masking Tone
PCM: Pulse Code Modulation
SMR: Signal to Mask Ratio
SMR: Signal to Mask Ratio
SPL: Sound Pressure Level
TMN: Tone masking noise

Clave de citas.

[1] Davis Pan, "A Tutorial on MPEG/Audio Compresión".
www.cs.columbia.edu/~coms6181 /~coms6181 /slides/6R/mpegaud.pdf

[2] Marcos Faúndez Zanuy, "Estándares de codificación de audio MPEG".
www.elsnet.org/expertcvs/0481.html

[3] T. Painter, A. Spanias, "Perceptual Coding Of Digital Audio ".
www.eas.asu.edu/~spanias/audiopaper1.pdf

[4] K Brandenburg and H. Popp, "An introduction to MPEG Layer-3".
citeseer.ist.psu.edu/brandenburg00introduction.html

[5] http://www.multiweb.cz/twoinches/MP3inside.htm.

[6] http://www3.labc.usb.ve/EC4514/AUDIO/PSICOACUSTICA/Psicoacustica.html.

[7] Ramapriya Rangachar "Analysis and improvement of the MPEG-1 Audio Layer III algorithm at low bit-rates".
etd.adm.unipi.it/theses/available/ etd-10062003-114626/unrestricted/

[8] Minh N. Do, "An Automatic Speaker Recognition System".
lcavwww.epfl.ch/~minhdo/asr_project/

[9] Daniel Neiberg, "Text Independent Speaker Verification Using Adapted Gaussian Mixture Models".
www.speech.kth.se/~neiberg/publications.html

[10] Douglas A. Reynolds, PhD, Larry P. Heck, PhD. - Automatic Speaker Recognition, Recent Progress, Current Applications, and Future Trends.
www.ll.mit.edu/IST/pubs/aaas00-dar-pres.pdf

[11] Qin Jin, Alex Waibel - Application of LDA to Speaker Recognition.
isl.ira.uka.de/publications/ICSLP2000-qin2.pdf

[12] Capítulo 4 - Cuantificación. Transparencias del curso de Codificación de Imágenes y Video. IIE (2004) - Facultad de Ingeniería.

[13] Reynolds D., Rose R., (1995), "Robust Text-Independent Speaker Identifcation Using Gaussian Mixture Speaker Models", IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 1, pp. 72-83.
www.aclclp.org.tw/clclp/v9n2/v9n2a5.pdf

[14] Manuel Carracedo Sánchez, Alfredo Jiménez Martín, Pierre Jean Riviere, Reconocimiento de locutores basado en mezclas gaussianas".
www.gaps.ssr.upm.es/TDV/trabajos.html

[15] Rabiner & Schafer, "Digital Processing of Speech Signals", Prentice Hall, 1978.

[16] A.V. Oppenheim & R.W.Schafer, "Discrete Time Signal Processing", Prentice Hall, 1989.

[17] Silvia Pfeiffer and Thomas Vincent (2001), Formalisation of MPEG-1 compressed domain audio features.

[18] David Pye "Content-Based Methods for the Management of Digital Music".