Codificación del audio

Marta Ruiz Costa-jussà
Helenca Duxans Barrobés

PID_00188067

UOC
Universitat Oberta de Catalunya

Índice

Introducción

La codificación permite obtener una representación más compacta de las señales de audio. Las ventajas que proporciona son las siguientes:

- En transmisión: **se reduce el ancho de banda** necesario para transmitir el audio. Por lo tanto, se puede aumentar la velocidad de transmisión o multiplexar diferentes flujos de audio en un mismo canal.

- En almacenamiento: **se reduce el número de bits** necesarios para representar la misma información. Por lo tanto, se consigue almacenar el mismo audio de modo que ocupa menos. Esto permite almacenar más contenido en un mismo soporte físico.

Objetivos

Este módulo presenta el proceso de codificación de la señal de audio para almacenarlo o transmitirlo digitalmente, y los procesos necesarios para recuperar el audio que ha sido codificado previamente. Así, los objetivos principales de este módulo son los siguientes:

1. Identificar los módulos comunes a todos los codificadores y las diferencias entre cada uno de estos módulos.

2. Saber de qué manera se puede cuantificar una señal discreta para obtener una señal digital.

3. Identificar las ventajas y los inconvenientes de los diferentes tipos de cuantificación, así como el error de cuantificación asociado a cada tipología.

4. Clasificar los codificadores según la estrategia utilizada para representar la señal discreta de manera comprimida.

5. Conocer los principales formatos de ficheros de audio.

1. Introducción al audio digital

La señal de audio analógico es continua en tiempo y en amplitud. Los codificadores digitalizan las señales para almacenar o transmitir.

Codificador de audio

Figura 1. Esquema funcional de un codificador de audio genérico

El proceso de digitalización de una señal consta de tres fases, como se muestra en la figura 1. En primer lugar, la señal analógica **se discretiza en tiempo** por medio de un convertidor A/D, que muestrea la señal de entrada a una frecuencia fija, denominada *frecuencia de muestreo*. A continuación, cada muestra **se discretiza en amplitud**, utilizando, como mínimo, un cuantificador para representar todos los valores de las muestras posibles con un número finito de símbolos. Finalmente, los símbolos se transforman en bits para transmitirlos o almacenarlos.

Ved también

Recordad que la conversión A/D y D/A la hemos explicado en el apartado "Conversión A/D y D/A. Entorno analógico y entorno digital" del módulo 1.

En la figura 2 se muestra una señal sinusoidal analógica (línea roja), la señal discreta obtenida una vez se ha muestreado (secuencia de barras azules), los símbolos que corresponden a cada muestra (valores de 0 a 15) y la transformación de estos símbolos a bits.

Proceso de digitalización.

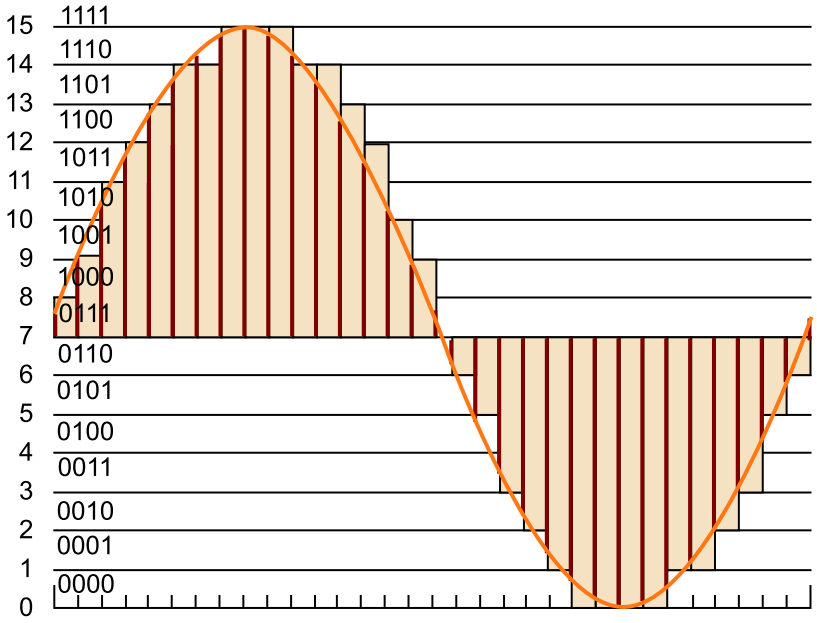


Figura 2. La señal sinusoidal analógica ($x(t)$ rojo) se discretiza en tiempo ($x[n]$ puntos señalados con las barras azules) y obtiene muestras, que se discretizan en amplitud mediante codificación 4-PCM (escalas en negro) y se obtiene la secuencia de 0 y 1 s.

La operación inversa a la codificación se denomina *descodificación* y teóricamente permite recuperar la señal original (a pesar de que en la práctica veremos que no somos capaces de recuperar exactamente la señal original).

La codificación tiene como objetivo representar una señal analógica de una manera **digital y compacta**, es decir, el objetivo que tiene es minimizar la cantidad de información necesaria para representar una señal. A la vez, intenta minimizar la pérdida de calidad de la señal que se obtiene cuando se descodifica.

Para evaluar la eficacia de un codificador se tienen en cuenta diferentes parámetros:

- La **fidelidad**, o cómo es de semejante para el oído humano el audio descodificado con el audio original.

- La **tasa de bits** (*bit rate*), o la cantidad de bits por segundo que se necesitan para representar la señal codificada.

- La **complejidad** de la codificación y la descodificación, que determina si se pueden llevar a cabo con software o es necesario disponer de un hardware dedicado.

- El **retardo** introducido para la codificación y la descodificación.

Para la edición de música profesional se necesitan codificadores con una alta fidelidad, aunque esto implique tener una tasa de bits elevada e, incluso, un dispositivo especial. En cambio, para la telefonía la calidad no es tan importante, siempre que la voz codificada sea inteligible: en el caso de la telefonía se busca reducir la tasa de bits para ahorrar ancho de banda en las transmisiones.

En este módulo veréis cómo se lleva a cabo la discretización en amplitud de las muestras (módulo central de la figura 1), proceso denominado *cuantificación*. También veréis las estrategias que utilizan los codificadores de audio actuales para mejorar la eficiencia de la codificación.

Transcodificación

Un **transcodificador** es un sistema que cambia la codificación aplicada a una señal. Por lo tanto, la entrada que tiene es una señal codificada (digital) y la salida, la señal recodificada según la nueva codificación elegida. Así, un transcodificador permite la conversión directa (de digital a digital) de una codificación a otra.

Para saber más

El audio digital nació a finales de los años cincuenta. Max Mathews y su equipo de los laboratorios Bell Telephone desarrollaron el denominado *teorema de muestreo* y fueron los primeros en generar sonidos simples mediante un ordenador. La limitación principal de entonces era la poca capacidad de las máquinas.

El primer sistema de grabación digital lo creó NHK en los años sesenta. La utilización de un cuantificador de 16 bits empezó con Stockham en 1976. El sonido digital llegó al público a principios de los ochenta gracias al disco compacto, creado por Sony y Philips. Más o menos, al mismo tiempo, surgían los primeros sintetizadores digitales. Más adelante se desarrollaron tipos de archivos como el MP3.

2. Cuantificación

El cuantificador es un componente imprescindible en los codificadores; por lo tanto, dedicaremos parte de este módulo a saber qué es un cuantificador y cuáles son sus características y su comportamiento.

La cuantificación consiste en transformar una señal continua en amplitud y discreta en tiempo en una señal digital (discreta en amplitud y tiempo). A la vez, se busca representar un número extenso de valores con un número más pequeño de valores.

Antes, definiremos algunos conceptos básicos sobre cuantificación: número de bits de cuantificación, número de niveles de cuantificación, margen dinámico, paso de cuantificación y error de cuantificación.

- El **número de bits (*b*)** nos indica la cantidad de estados de salida del cuantificador. Un cuantificador tiene más resolución si tiene más número de bits.

- Los **niveles de cuantificación (*N*)** son los valores nuevos que toma la señal cuantificada y vienen dados por el número de bits del cuantificador. Su expresión es $N = 2^b$.

- El **margen dinámico (*MD*)** de un cuantificador nos indica el mínimo y máximo de la señal que se debe cuantificar ($-x_{max}$, x_{max}).

Rango dinámico

El mínimo y máximo de la señal que se ha de cuantificar se determina, entre otros rangos, a partir del rango de valores analógicos que tiene la señal o a partir del rango de valores analógicos que nos interesa para nuestro procesamiento digital.

- El **paso de cuantificación** (Δ) se define como la diferencia que hay entre dos niveles de cuantificación consecutivos:

$$\Delta = \frac{MD}{2^b}$$

- El **error de cuantificación** se define como la distancia entre la señal original y la señal cuantificada; por lo tanto, que:

$$e[n] = \quad y[n] - x[n] = Q(x[n]) - x[n]$$

En la figura 3 vemos una representación gráfica del error de cuantificación:

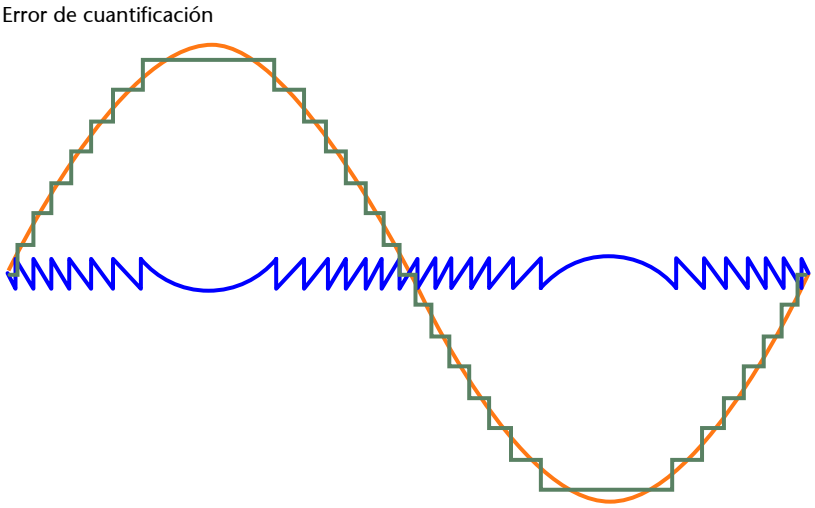


Figura 3. Representación de una señal analógica (rojo), el resultado de la cuantificación de esta señal (verde) y el error de cuantificación (azul). La señal analógica (en rojo) se discretiza en tiempo con una frecuencia de muestreo muy alta y al cuantificarse en amplitud (escalas verdes) surge un error de cuantificación (en azul).

> Si tenemos una señal continua cuya amplitud oscila entre 0 y 3, una cuantificación posible es que a todos los valores entre 0 y 1 les demos un valor de 0,5, a los valores entre 1 y 2 un valor de 1,5 y a los valores entre 2 y 3 un valor de 2,5. Esto es una cuantificación uniforme. Por lo tanto, los niveles de cuantificación son 0,5, 1,5 y 2,5.

En los próximos apartados veremos diferentes estrategias para cuantificar una señal.

2.1. Cuantificación uniforme

Sabemos que para cada valor de la señal $x[n]$ tenemos un valor asociado $y[n]$, donde $y[n] = Q(x[n])$. La particularidad de la cuantificación uniforme consiste en considerar exclusivamente **niveles de cuantificación (L) distribuidos de la misma manera**. Es el tipo de cuantificación más sencilla que se utiliza.

La figura 4 muestra un ejemplo de cuantificador uniforme de 3 bits. El número de niveles de un cuantificador viene dado por 2^b, donde b es el número de bits. En el caso particular de tener 3 bits, hay ocho niveles de cuantificación, en los que cada nivel tiene asignado un código binario entre 000 y 111:

Cuantificador uniforme

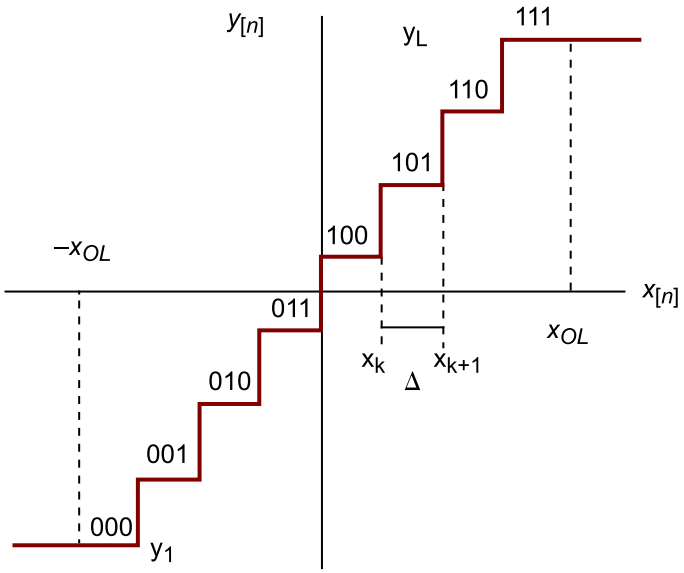


Figura 4. Ejemplo de cuantificador uniforme de ocho niveles (3 bits) (, $0 < L < 7$)

Así, observamos que el cuantificador uniforme de la figura 4 contiene los intervalos de cuantificación siguientes:

$$x_1 = -\infty, x_2, \ldots, x_L, x_{L+1} = \infty$$

y estos niveles de cuantificación:

$$y_{k+1} - y_k = \Delta, \qquad k = 1, \ldots, \quad L-1$$

$$x_{k+1} - x_k = \Delta, \qquad x_{k+1}, x_k \quad \text{finitos}$$

Para definir los intervalos de cuantificación hacemos lo siguiente:

- Definimos el margen dinámico del cuantificador.
- Definimos el número de bits que queremos utilizar (b), que nos da el número de niveles de cuantificación: $N = 2^b$.
- Obtenemos el paso de cuantificación:

$$\Delta = \frac{MD}{2^b} = \frac{(x_{max} - x_{min})}{2^b}$$

- Definimos cada intervalo de cuantificación (q) como la diferencia entre el valor más alto y más bajo de la entrada a los que se asigna el mismo estado de salida.

A partir de los niveles de cuantificación de un cuantificador uniforme podemos definir la relación señal-ruido (SNR) que tienen.

Referencia bibliográfica

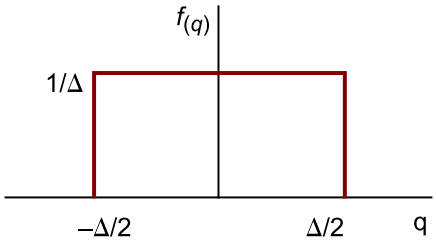
A. Moreno (2003). "Cuantificación".

Recordamos que la SNR, como indica su nombre, es el cociente entre el nivel de señal (σ_x^2) y el nivel de ruido (σ_q^2).

A continuación, estudiamos el nivel de ruido que tiene un cuantificador uniforme. El nivel de ruido se obtiene a partir de la varianza, que se define de la manera siguiente:

$$\sigma_q^2 = \int_{-\infty}^{\infty} q^2 f_q(q) dq$$

donde $f_q(q)$ es la función densidad de probabilidad del error de cuantificación. Consideramos que el error de cuantificación está distribuido uniformemente. Entonces, tenemos que la función densidad de probabilidad del error de cuantificación es gráficamente la siguiente:



Y, analíticamente, la función densidad de probabilidad es esta:

$$f_q(q) = \begin{cases} \frac{1}{\Delta} & |q| \leq \frac{\Delta}{2} \\ 0 & el \quad resto \end{cases}$$

Así pues, concretamente, la varianza es la siguiente:

$$\sigma_q^2 = \int_{-\infty}^{\infty} q^2 f_q(q) dq = \int_{-\Delta/2}^{\Delta/2} q^2 \frac{1}{\Delta} dq = \frac{\Delta^2}{12}$$

Sabemos que:

$$\Delta = \frac{(x_{max} - x_{min})}{2^b}$$

Por lo tanto:

$$\sigma_q^2 = \frac{1}{12}\left(\frac{(x_{max} - x_{min})}{2^b}\right)^2$$

Y, finalmente, la SNR en decibelios es esta:

$$\mathrm{SNR}_{\mathrm{dB}} = 10log\frac{\sigma_x^2}{\sigma_q^2} = 10b\,\log(4) + 10log3 + 10\log\left(\frac{\sigma_x^2}{x_{max}^2}\right)$$

$$= 6b + 10log3 - 20\log(\frac{x_{max}}{\sigma_x})$$

Observando el resultado vemos que la SNR mejora 6 dB por cada bit que añadimos al cuantificador, independientemente del tipo de señal que se tenga que cuantificar. Ahora bien, la SNR también depende de la proporción entre el valor máximo de la señal y la varianza que tiene. Cuanto mayor es el cociente, peor es la SNR.

Observad que es difícil decidir el rango de un cuantificador porque un cuantificador debe ser válido para diferentes tipos de señales: voz (señales sordas y sonoras, señales pronunciadas por diferentes locutores), música, etc.

2.2. Cuantificación no uniforme

La cuantificación no uniforme asigna **niveles de cuantificación que no están distribuidos uniformemente**. La principal ventaja de este tipo de cuantificación es que se puede adaptar a la señal. Así, si una señal contiene más información en un margen de amplitud concreto, se asignan más niveles de cuantificación en este margen.

Para hacer un cuantificador no uniforme se deben buscar los intervalos x_k y los niveles de cuantificación óptimos, de modo que se minimice la variancia del error de cuantificación.

Una manera de elaborar una cuantificación no uniforme es hacer una compresión de la señal, después una cuantificación uniforme y finalmente una expansión de la señal, como muestra la figura 5. Esta manera de hacer una cuantificación no uniforme se denomina **cuantificación logarítmica**.

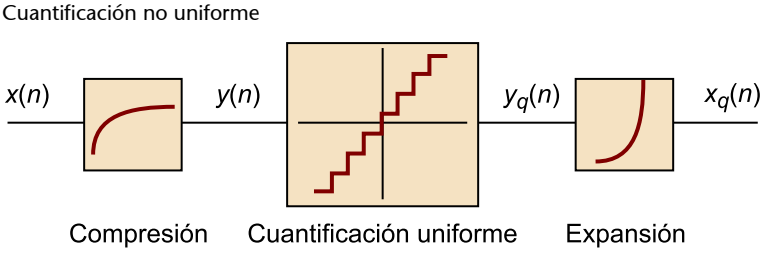


Figura 5. Pasos que se deben seguir en una cuantificación escalar o logarítmica: compresión, cuantificación uniforme y expansión de la señal

Por lo tanto, la cuantificación escalar o logarítmica consiste en añadir un **compresor logarítmico** antes de elaborar una etapa de cuantificación uniforme convencional. La utilidad del compresor logarítmico es muy clara para señales de audio. Sabemos que una señal de audio puede tener un rango de amplitudes muy extenso (superior a 60 dB) pero no todas las amplitudes son igualmente probables. Interesa minimizar el error de cuantificación (es decir, aumentar la

Referencia bibliográfica

Asunción Moreno (2003). "Cuantificación".

resolución del cuantificador) donde las amplitudes de la señal son más probables (por ejemplo, en señales de voz telefónicas, los valores de las amplitudes pequeñas son los más probables).

En términos generales, podemos decir que hay dos estándares para hacer un cuantificador logarítmico: la ley-A (europeo) y la ley- μ (americano y japonés). La diferencia entre una y otra es el tipo de compresión y expansión. Ahora bien, el objetivo es el mismo, esto es, amplificar los valores con amplitudes más pequeñas antes de hacer la cuantificación. La figura 6 muestra la compresión y expansión que se realiza de la señal. Con la señal de salida de esta compresión y expansión se efectúa la cuantificación uniforme.

Ley-A y ley-μ de compresión y expansión de la señal de voz

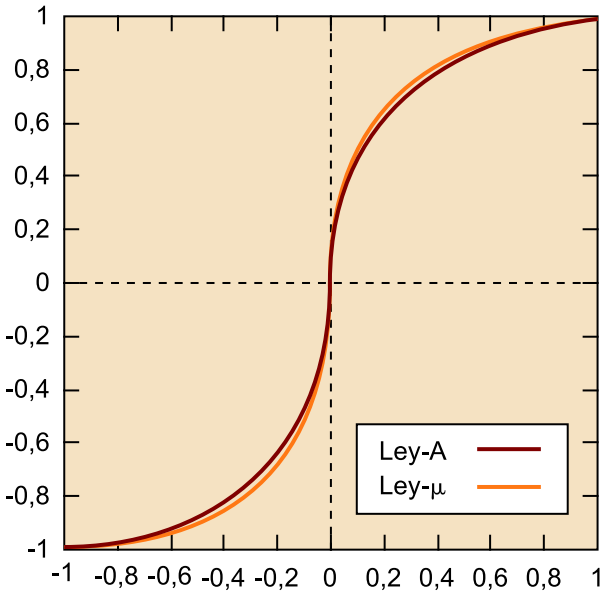


Figura 6. Ley-A y ley-μ de compresión y expansión de la señal de voz

Esta cuantificación se utiliza en telefonía.

La varianza del error de cuantificación en un cuantificador logarítmico incorpora la curva de compresión $c(x)$:

$$\sigma_q^2 = \int_{-\infty}^{\infty} q^2 c(x) f_q(q) dq = \int_{-\Delta/2}^{\Delta/2} q^2 c(x) \frac{1}{\Delta} dq$$

2.3. Cuantificación vectorial

La cuantificación vectorial (VQ) **cuantifica los datos en bloques** de *N* muestras. Así, a diferencia de la cuantificación escalar (uniforme o no uniforme), la cuantificación vectorial tiene en cuenta los valores de la señal de manera conjunta. Para empezar a elaborar la cuantificación se crean bloques de *N* muestras. Cada bloque de *N* muestras se trata como un vector de dimensión *N*.

Cada vector de dimensión *N*, $\overline{x} = [x_1, x_2, \ldots, x_N N,]$ se codifica mediante un vector también de la misma dimensión $\overline{y} = [y_1, y_2, \ldots, y_N]$. Cada vector $\overline{x}$ se codifica en el vector $\overline{y}$ respecto al que tiene una distancia más pequeña. Una medida de distancia entre dos vectores que se puede utilizar es la distancia euclidiana.

Distancia euclidiana

La distancia euclidiana entre dos vectores viene dada por la distancia entre cada una de las dimensiones del vector; así, en un espacio bidimensional, tenemos que:

$$\text{dist. euclidiana} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Los vectores $\overline{y}$ forman lo que se denomina **biblioteca de códigos** o ***codebook***. Conocida la biblioteca de códigos, solo se debe guardar el índice del vector y por ello se puede hacer una codificación más elevada. El número de vectores (*M*) que contiene la biblioteca de códigos se denomina *medida de la biblioteca*. La biblioteca la denominamos *Y*:

$$Y = \vec{y}_i, \qquad i = 1 \ldots M$$

Esta cuantificación es más eficiente que las cuantificaciones escalares pero es más compleja computacionalmente. Sobre todo es complicado diseñar la biblioteca de códigos. Uno de los criterios que se utiliza para diseñar la biblioteca de códigos es minimizar el error cuadrático medio entre los vectores *x* e *y*, que se define de la manera siguiente:

$$d(x, y) = \frac{1}{N} \quad (x - y)^T (x - y) = \frac{1}{N} \sum_{k=1}^{N} (x_k - y_k)$$

Este error cuadrático medio es una manera de medir el error de cuantificación.

Por otro lado, a menudo se utilizan bibliotecas de códigos marcadas por estándares a los que, a pesar de no ser óptimos para la señal que se ha de cuantificar, no hay que adjuntar la biblioteca de códigos. Estas bibliotecas se han diseñado a partir de señales de entrenamiento y no a partir de la señal que se ha de cuantificar.

El algoritmo de Lloyd (denominado también *k-means*) se utiliza para diseñar bibliotecas de códigos y se basa en la estrategia siguiente:

1. Inicialización. Se selecciona una biblioteca inicial formada por M vectores ($\boldsymbol{y}$) de la señal que queremos codificar. Cada uno de estos vectores de la biblioteca es el representante de un grupo de vectores $\{\boldsymbol{x}\}$ y se denomina *centroide*. La elección de la biblioteca inicial tiene influencia en la efectividad final; en algunos casos se puede hacer de manera aleatoria o a partir de otros métodos.

2. Clasificación. Se clasifica el conjunto de vectores de entrenamiento en M subgrupos, formados por los vectores $\boldsymbol{x}$, de manera que $x \epsilon \text{Grupo}_i \Leftrightarrow \forall\ j$ se verifica que $d(x, y_i) \leq d(x, y_j)$.

3. Actualización del diccionario. Se calcula para cada grupo de vectores el nuevo centroide $\boldsymbol{y}$. Este nuevo centroide cumple que es el vector que minimiza el error cuadrático medio del grupo.

4. Final o regreso al paso 2. Se acaba si se cumple cualquiera de las condiciones siguientes:

- El error cuadrático medio total no disminuye de manera significativa respecto a la iteración anterior.
- Se han hecho N iteraciones del algoritmo (N es un valor prefijado).
- El error cuadrático medio está por debajo de un umbral preestablecido, o en caso contrario se vuelve al paso 2.

El error de cuantificación en este caso es la suma de las distancias de los vectores x a los centroides.

Queremos aplicar el algoritmo de Lloyd para hacer la cuantificación de una señal triangular en Matlab. En primer lugar, diseñamos la señal triangular muestreada mediante las órdenes siguientes:

```
x=repmat([0:0.1:1 0.9:-0.1:0],1,1);
y=[1:1:21];
z=[x;y];
```

En segundo lugar, aplicamos el algoritmo de Lloyd a *z*, que es la representación de la señal en vectores de dos dimensiones, para crear una biblioteca de códigos de medida 5:

```
[U, v, sumd, D]=kmeans(z,5);
```

U es la matriz que define a qué grupo pertenece cada vector de la señal; *v* son los centroides; *D* son las distancias de cada punto de la señal al centroide y *sumd* es la suma de distancias dentro de cada centroide.

Con las órdenes siguientes visualizamos la señal y la cuantificación:

```
plot(z(:,1),z(:,2),'v');
hold on;
plot(v(:,1),v(:,2),'sr');
```

Aplicación

La señal de audio normalmente utiliza cuantificadores de 8, 16 o 20 bits. Esta cuantificación implica que una señal sinusoidal (un tono puro) tiene una relación de señal a error de cuantificación (SQNR) máxima, aproximadamente de 50, 100 y 123 dB, respectivamente. La calidad de CD se suele cuantificar con 16 bits (por canal si es sonido en estéreo), puesto que en la práctica los aparatos de música no reproducen más de 90 dB.

La cuantificación es el proceso que permite transformar una señal discreta a una señal digital. Por ejemplo, la señal discreta [0,2 0,35 0,7] a señal digital [0 0 1].

La cuantificación uniforme distribuye los valores de la señal original en *L* niveles separados uniformemente.

Como a menudo la información de una señal se concentra en un rango de valores determinado, es conveniente utilizar un cuantificador no uniforme que presente más granularidad donde la señal concentra más información. Dos estándares de cuantificación no uniforme son la ley-A y la ley-μ.

La cuantificación vectorial cuantifica bloques de *N* muestras a la vez, es decir, vectores de longitud *N*. Así, se debe diseñar la biblioteca de códigos que contiene los vectores representativos de los vectores de la señal original.

Referencia bibliográfica

Asunción Moreno (2003). "Cuantificación".

Rafael Molina. "Cuantificación escalar".

3. Cuantificación inversa

La operación inversa a la cuantificación se denomina *cuantificación inversa*. La entrada de un cuantificador inverso, representado normalmente por Q^{-1}, es una secuencia de niveles correspondientes a una señal cuantificada (señal digital), y la salida es la secuencia de muestras de la señal reconstruida (señal discreta). Podéis ver la figura 7.

Cuantificación inversa

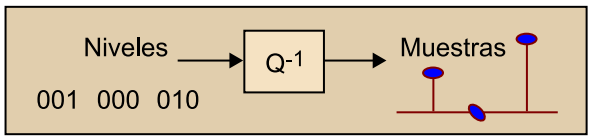


Figura 7. Representación de la funcionalidad de la cuantificación inversa: la entrada del Q^{-1} se corresponde con una secuencia de niveles, y la salida, con muestras de la señal.

Normalmente, el cuantificador inverso está formado por dos etapas. En la primera etapa se lleva a cabo la correspondencia nivel-valor de muestra según el tipo de cuantificación aplicado (escalar uniforme, no uniforme o vectorial). En la segunda etapa se aplica un interpolador a las muestras obtenidas para suavizar la señal reconstruida. El interpolador más simple es un filtro paso bajo.

Como ya habréis advertido, cuando se realiza una cuantificación inversa no se puede recuperar perfectamente la señal original (la señal antes de codificar). En el proceso de cuantificación puede que haya valores diferentes (por ejemplo, 0,25 y 0,35) que sean cuantificados con el mismo nivel de cuantificación (por ejemplo, 100). Por lo tanto, cuando se vuelve atrás con la cuantificación inversa, no se puede saber si el nivel 100 pertenecía al valor 0,25 o al valor 0,35. En este caso hemos de fijar un valor para la correspondencia nivel-muestra, como por ejemplo el valor medio del intervalo (por ejemplo, 0,30 para el intervalo 0,25-0,35). Por lo tanto, todos los valores que vengan del nivel 100 serán transformados en el valor 0,30.

4. Procesos del audio digital

En este apartado veremos un par de técnicas relacionadas con la cuantificación del audio digital.

4.1. Sobremuestreo

El **sobremuestreo** u ***oversampling*** consiste en **muestrear la señal utilizando una frecuencia de muestreo significativamente más alta que dos veces el ancho de banda de la señal**. Así, el factor de sobremuestreo (β) se define como sigue:

$$\beta = \frac{f_m}{2B}$$

Lectura de la fórmula

f_m es la frecuencia de muestreo.
B es el ancho de banda de la señal.

¿Por qué se decide hacer un sobremuestreo? Básicamente, en la práctica tenemos tres motivaciones:

1) Permite conseguir una resolución más alta en la conversión A/D y D/A, lo que implica aumentar la SNR. En la práctica, la relación entre el aumento de resolución y el aumento de la frecuencia de muestreo viene dada por la relación siguiente:

$$f_{ov} = 4^w 2B$$

Lectura de la fórmula

f_{ov} es la frecuencia de sobremuestreo.
w es el número de bits que queremos aumentar en resolución.

Por lo tanto, por cada bit que queremos aumentar en resolución, debemos multiplicar por cuatro la frecuencia de muestreo.

Si queremos implementar un convertidor de 20 bits, debemos utilizar un convertidor de 16 bits y una frecuencia de sobremuestreo que sea 256 (4^4) veces la frecuencia de Nyquist (es decir, dos veces el ancho de banda de la señal).

Referencia bibliográfica

Th. Zawistowski; P. Shah. "An Introduction to Sampling Theory".

2) Tiene un efecto de *anti-aliasing* (antiencabalgamiento). Si recordamos el teorema de muestreo, muestrear en el dominio de la frecuencia quiere decir repetir la señal cada f_m. Si esta f_m es grande, hay menos probabilidades de encabalgamiento. Podéis ver la figura siguiente:

Ejemplo de muestreo y sobremuestreo

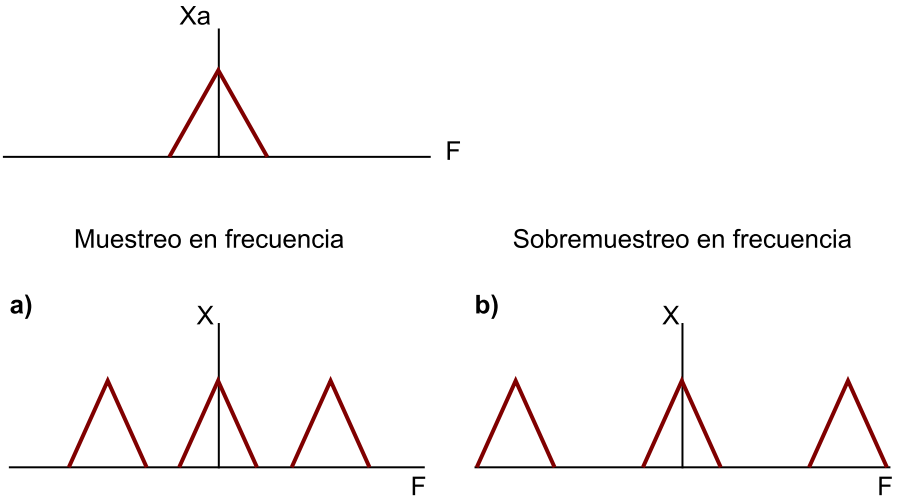


Figura 8. Ejemplo de señal muestreada (*a*) y sobremuestreada (*b*) en el dominio de la frecuencia

3) Reduce el ruido asociado a la señal analógica. Así, si la señal analógica tiene ruido aleatorio y tomamos *N* muestras, lo que hacemos es la media por *N*; por lo tanto, se reduce el ruido en un factor 1/ *N*. Esto significa que la SNR mejora en un factor *N*.

$$x_m(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$$

El sobremuestreo se aplica cuando la señal tiene asociado un ruido aleatorio. Si no, fijaos en que las tres motivaciones principales comentadas no se verificarían necesariamente.

4.2. Tramado

El **tramado** o ***dither*** es un **ruido intencionado que se utiliza para hacer aleatorio el error de cuantificación**, lo que permite reducir el ruido que genera la cuantificación en las grabaciones de audio.

Ejemplo

Imaginemos que tenemos el valor 3,8 y lo debemos cuantificar a 3 o a 4. Sabemos que con un 0,8 de probabilidad este valor debe estar cuantificado a 4, y con un 0,2 de probabilidad, a 3. Si cuantificamos sin tramado, este valor siempre se cuantificará a 4 y, por lo tanto, siempre cometeremos el mismo error. Si utilizamos tramado, debemos hacer lo siguiente: calculamos una serie de números aleatorios entre 0,0 y 0,9. Si aparecen los números 0,0 o 0,1, redondearemos a 3, y con cualquiera otro número redondearemos a 4. Entonces tenemos un 20% de probabilidades de redondear a 3 y un 80% de redondear a 4. El error de cuantificación es aleatorio y, por lo tanto, menos molesto para el oído.

Hemos visto dos técnicas que aplicando conocimiento sobre el sistema acústico humano y la percepción del sonido (podéis ver el apartado "Percepción del sonido") permiten perfeccionar la cuantificación:

- El sobremuestreo consiste en muestrear la señal a una frecuencia de muestreo sensiblemente más elevada que dos veces el ancho de banda que tiene.

- El tramado quiere aleatorizar el error de cuantificación, porque así molesta menos al oído humano.

5. Clasificación de los codificadores de audio

Hasta ahora hemos visto los cuantificadores como sistemas que permiten discretizar la amplitud de las muestras; por lo tanto, los cuantificadores son un módulo necesario para los codificadores. Existen otras técnicas, como las que comentaremos a continuación, que, utilizadas junto con un cuantificador, aumentan la eficacia de la codificación.

La principal estrategia de codificación, utilizada para reducir la tasa de bits manteniendo la fidelidad del audio, es hacer algún tipo de procesamiento en las muestras de la señal antes de aplicar el cuantificador, es decir, entre el bloque de conversión A/D y el cuantificador de la figura 1. Según cómo sea este procesamiento, la codificación se puede clasificar en:

- Codificación de forma de onda.
- Codificación perceptiva.
- Codificación específica para la voz.

En los apartados siguientes veremos ejemplos de estos tres tipos de cuantificación.

6. Codificadores de forma de onda

Los codificadores de forma de onda intentan representar de manera compacta la forma de onda de la señal, independientemente del origen que tenga. Por lo tanto, este tipo de codificadores se pueden utilizar para codificar cualquier tipo de señal (audio, vídeo, comunicaciones) o cualquier tipo de datos.

Los codificadores de forma de onda pueden trabajar tanto en el dominio temporal como en el dominio frecuencial. En ambos casos, intentan eliminar la redundancia que hay en la señal de entrada para reducir la tasa de bits. Los codificadores de forma de onda más utilizados que veremos en las próximas secciones son los siguientes:

- Codificadores en el dominio del tiempo: PCM, DPCM y ADPM.
- Codificadores en el dominio de la frecuencia: en subbandas y por transformada.

Los codificadores de forma de onda son codificadores sin pérdidas, es decir, la señal obtenida de la descodificación es muy similar a la señal de entrada de la codificación. Solo se ha introducido el ruido procedente del error de cuantificación.

Este tipo de codificadores son robustos frente a ruidos y errores de transmisión y proporcionan una calidad alta de la voz codificada-descodificada cuando trabajan con tasas de bits media, en torno a 32 kbps. Para tasas de bits más pequeñas, la calidad proporcionada es peor que la de otros codificadores que veremos más adelante.

6.1. PCM: modulación en impulsos codificados

La modulación en impulsos codificados o *pulsecode modulation* (PCM) es el codificador más simple. El audio de entrada se muestrea a una velocidad constante y la amplitud de cada muestra cuantifica con un cuantificador de los que hemos explicado anteriormente (uniforme, no uniforme o vectorial). En la descodificación, simplemente se aplica el cuantificador inverso para obtener otra vez las muestras de la señal de audio. Podéis ver la figura 9.

Codificación PCM

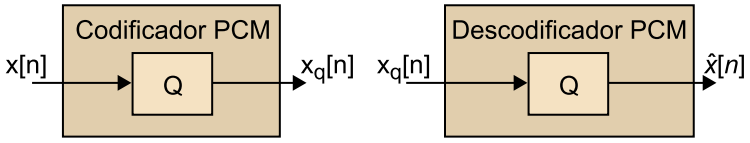


Figura 9. Diagrama de bloques de un codificador PCM (izquierda) y del descodificador (derecha)

Si la velocidad de muestreo es bastante elevada, se puede considerar que en el proceso de codificación no se pierde información; la única fuente de error es la cuantificación. Cuantos más niveles de codificación se utilicen, más fiel a la señal de entrada es la señal de salida. Sin embargo, cuanta más velocidad de muestreo y más niveles de codificación existan, más coste computacional y de almacenamiento habrá.

Ejemplo

A pesar de ser la codificación con una estructura más simple, el uso que se hace de ella está muy extendido; por ejemplo, se utiliza para codificar los CD comerciales de música. Las características principales de la codificación utilizada en los CD son las siguientes:

- Frecuencia de muestreo: 44,1 kHz. Por lo tanto, el ancho de banda de la señal que se ha de codificar es de 22,05 kHz (recordad que el oído humano tiene el límite de audición en torno a 20 kHz).
- 16 bits por muestra; esto significa 2^{16} = 65.536 niveles (rango dinámico de 90 dB).
- Estéreo: dos canales. Cada canal tiene una tasa de bits de 44,1 kHz * 16 = 705, 6 kbps; en total 1.411 kbps.

En un CD de música caben unos setenta y cinco minutos de audio codificado.

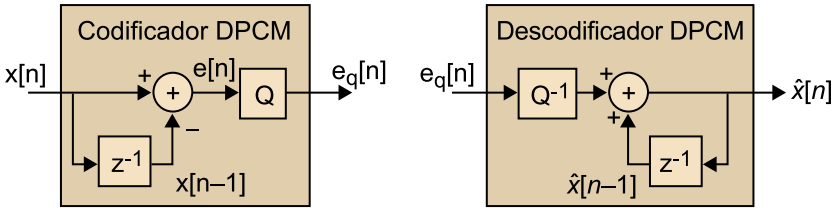
Para saber más

K. Immink (1998). "The compact Disc Story". *JAES* (vol. 46, núm. 5, pág. 458-462).

6.2. DPCM: modulación diferencial por impulsos codificados

Cuando la señal que se debe codificar presenta mucha correlación (similitud) entre las muestras adyacentes, como es el caso de la voz, se puede bajar la tasa de bits de los codificadores PCM codificando la diferencia entre las muestras adyacentes, en lugar del valor de cada muestra (podéis ver la figura 10):

Codificación DPCM



$$e[n] = x[n] - x[n-1]$$

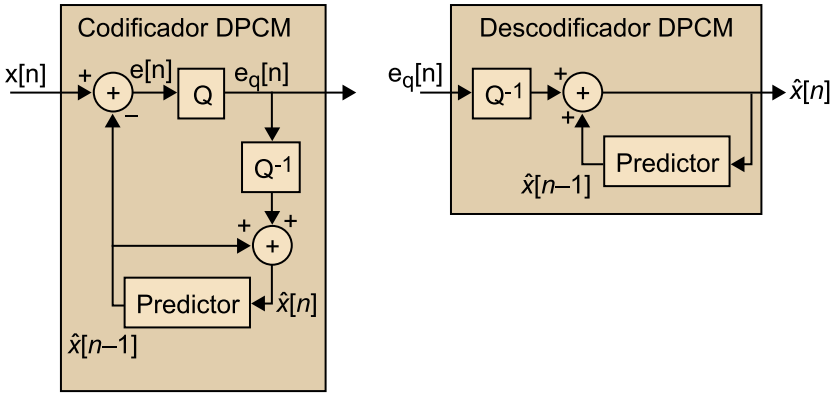
$$\hat{x}[n] = \hat{x}[n-1] + e_q[n]$$

Figura 10. Diagrama de bloques de un codificador DPCM (izquierda) y del descodificador (derecha)

El rango dinámico de la señal diferencia $e[n]$ es mucho más pequeño que el rango dinámico de la señal de entrada $x[n]$ (siempre que las muestras de $x[n]$ estén correlativas entre sí). Por lo tanto, se necesitan menos niveles para codificar $e[n]$ que para codificar $x[n]$ con el mismo error de cuantificación. Si hay menos niveles, hay un número más pequeño de bits por muestra.

Una implementación alternativa del codificador DPCM consiste en incorporar el descodificador en el proceso de codificación (implementación denominada también *análisis por síntesis*). En la figura 11 podéis ver un diagrama de bloques de un DPCM basado en análisis por síntesis:

Análisis por síntesis



$$e[n] = x[n] - \hat{x}[n-1]$$

$$\hat{x}[n] = \hat{x}[n-1] + e_q[n]$$

Figura 11. Diagrama de bloques de un codificador DPCM basado en análisis por síntesis

Opcionalmente, el retardador del DPCM se puede sustituir por un módulo predictor para intentar minimizar todavía más el rango dinámico de $e[n]$.

Predictor

Un predictor es un sistema en el que cuando se introduce una señal en la entrada proporciona a la salida una predicción de cuál será la muestra siguiente de la señal.

Fijaos en que el descodificador no ha cambiado respecto a la primera implementación DPCM que hemos presentado en la figura 10 (a excepción del predictor que podría ser un retardador).

6.3. ADPCM: modulación diferencial adaptativa por impulsos codificados

La codificación de modulación diferencial adaptativa por impulsos codificados (ADPCM) se basa en convertir el cuantificador constante de la codificación DPCM en adaptativo. Es decir, los niveles de cuantificación que se aplican a $e[n]$ varían según la propia señal $e[n]$. Así, el paso de cuantificación es variable dependiendo de la diferencia entre una muestra de la señal de entrada en un instante y la correspondiente al instante anterior (podéis ver la figura 12).

Adaptación del paso de cuantificación

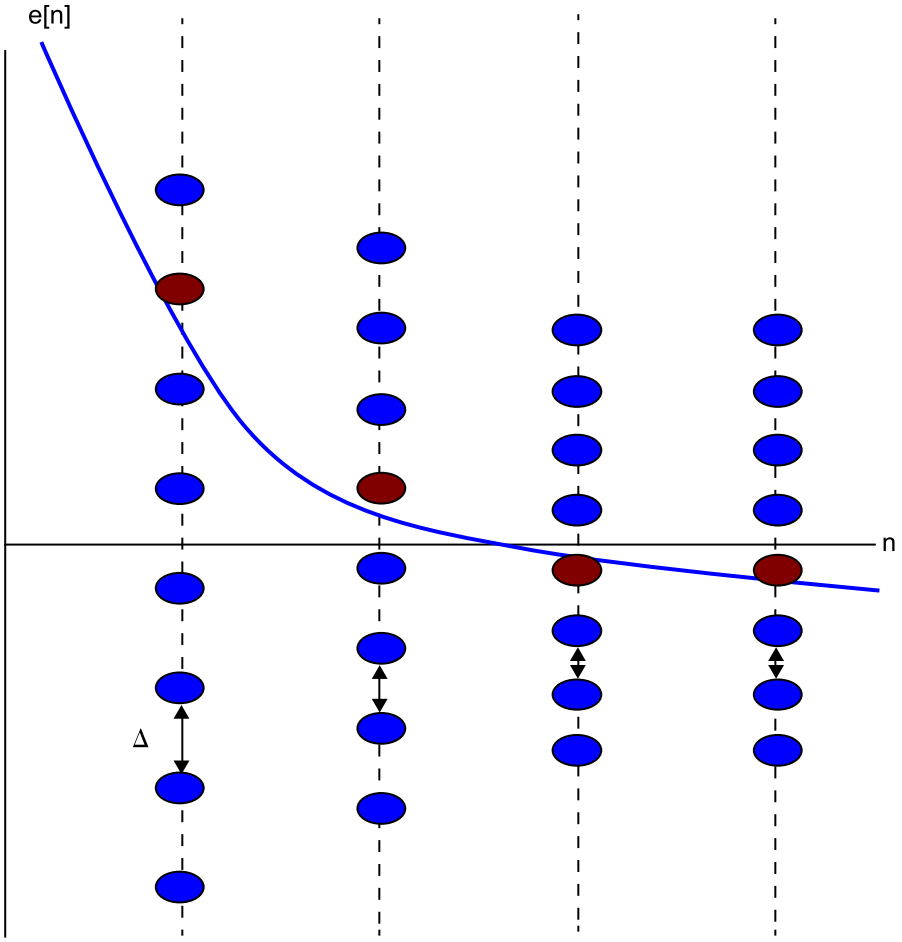


Figura 12. Adaptación del paso de cuantificación (Δ) según el nivel de señal que se debe cuantificar

Se aplica la adaptación para:

- Reducir el error de cuantificación, manteniendo el mismo número de bits por muestra.

- Reducir el número de bits por muestra, manteniendo el error de cuantificación.

Como ya debéis de haber notado, si el cuantificador del codificador varía los niveles de cuantificación que tiene, el cuantificador inverso del descodificador debe estar totalmente sincronizado para recuperar $\widehat{x}[n]$.

Hay dos estrategias para hacer la adaptación de los niveles de cuantificación: *feedforward* y *feedbackward*. En la estrategia *feedforward*, la estimación de los nuevos niveles de cuantificación se lleva a cabo en el codificador utilizando un bloque de voz. Por lo tanto, por cada paso de cuantificación que se valora, es necesario proporcionar esta información al descodificador, y como consecuencia se incrementa el tráfico de la transmisión o la dimensión del fichero codificado. En cambio, en la estrategia *feedbackward*, no se transmite el paso de cuantificación, sino que la estimación de los nuevos niveles de cuantificación se efectúa a partir de información que está presente tanto en el codifica-

dor como en el descodificador (por ejemplo, $e_q[n]$); por lo tanto, no hay ningún incremento de tránsito o dimensión. Aun así, la estrategia *feedbackward* es menos robusta, puesto que si hay errores de transmisión pueden provocar una desincronización entre el codificador y el descodificador, y, por lo tanto, la versión reconstruida $\hat{x}[n]$ no será tan similar a $x[n]$.

La adaptación de la codificación ADPCM se puede extender al módulo predictor de la señal $x[n]$. Es decir, los coeficientes fijos del módulo predictor del codificador DPCM se convierten en variables para adaptarse a la dinámica de la señal y así proporcionar una predicción mejor. En este caso se intenta minimizar la señal $e[n]$ para reducir todavía más el rango dinámico.

Ejemplo

La estándar G.726 de la ITU-T (sector de estandarización de la International Telecomunication Union) prevé la compresión de una llamada telefónica por medio del codificador ADPCM. En concreto, se utiliza para reducir la codificación ley-μ (Estados Unidos) o ley-A (Europa) PCM de 8 bits por muestra a 4 bits por muestra.

6.4. Codificación en subbandas

La codificación en subbandas consiste en dividir la señal original en diferentes bandas espectrales (proceso denominado *análisis basado en bancos de filtros*) y codificar cada una de estas bandas de manera independiente, utilizando una técnica de codificación de forma de onda (por ejemplo, DPCM o APCM).

Un diagrama de bloques genérico de la codificación por subbandas es el que muestra la figura 13.

Codificación en subbandas

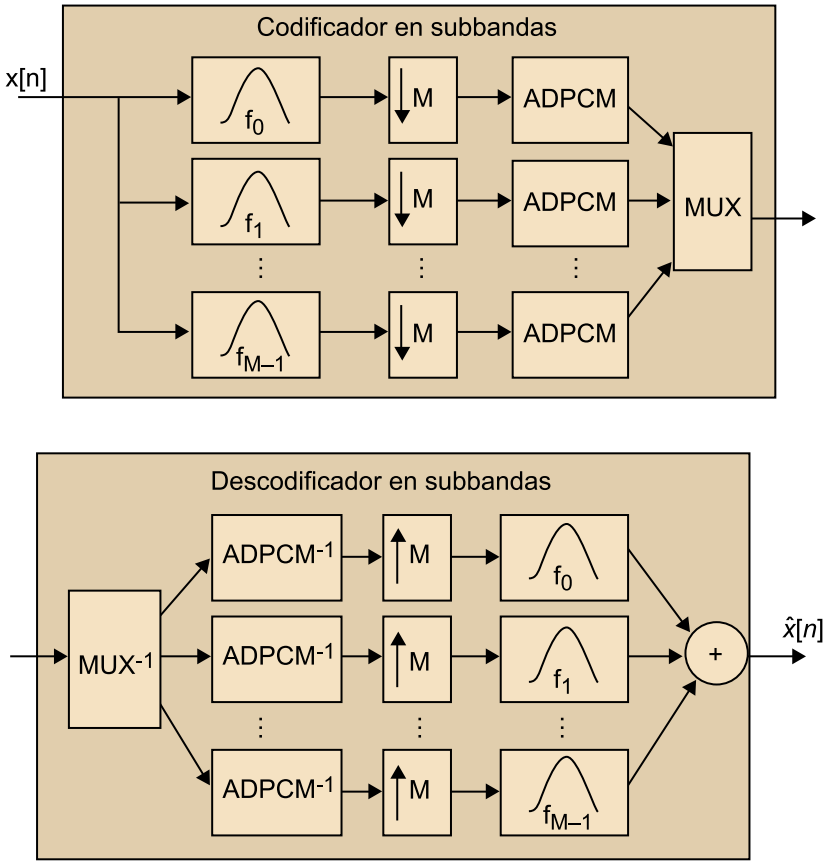


Figura 13. Diagrama de bloques de un codificador (figura superior) y un descodificador (figura inferior) basado en subbandas

El procedimiento para codificar en subbandas es el siguiente. Primero, se filtra la señal de entrada por medio de *M* filtros paso banda para separar la señal en *M* subbandas. En este punto del proceso, en lugar de tener una sola señal que se ha de codificar ($x[n]$), tenemos *M* señales que se deben codificar (el resultado de filtrar $x[n]$ por cada uno de los *M* filtros); por lo tanto, hemos aumentado por un factor *M* el número de muestras por segundo que se han de codificar. Para mantener constante el número de muestras que se deben codificar por segundo, la salida de los filtros se diezma[1] por un factor *M*. A continuación, se codifica la señal de cada subbanda por medio de un codificador de forma de onda de dominio temporal (por ejemplo, un ADPCM) y se combina la salida de los *M* codificadores con un multiplexor para formar la codificación de la señal de entrada completa.

(1)Diezmar es el proceso por el que en la salida del diezmador solo se mantiene una muestra de cada *M* muestras de la señal de entrada. El resultado es equivalente a una reducción de la frecuencia de muestreo de la señal de entrada por el mismo factor *M*.
Para más información, podéis consultar el libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

Para descodificar la señal, primero se separan las contribuciones de cada subbanda con un desmultiplexor y se descodifica cada aportación por separado. A continuación, se interpola[2] la señal obtenida para cada subbanda con un factor *M* para obtener el número de muestras por segundo originales. Finalmente, se filtra la señal de cada subbanda por el filtro paso banda correspondiente, para asegurar que los procesamientos previos no introduzcan componentes frecuenciales nuevas fuera de la banda de paso, y se suman todas las aportaciones.

(2)La interpolación es el proceso por el que se introducen (M-1) muestras por cada muestra de la señal de entrada. El resultado es equivalente a un aumento de la frecuencia de muestreo de la señal de entrada por el mismo factor *M*.
Para más información, podéis consultar el libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

Como las propiedades espectrales de la señal de entrada cambian con el tiempo, la distribución del número de bits por bandas es adaptativa, de modo que se utilizan más bits en las bandas que tienen más energía y menos en las bandas de menos energía.

El diseño de un banco de filtros paso banda que permita dividir y reconstruir la señal sin pérdida de información de la señal de entrada es un punto crítico para obtener una codificación en subbandas de calidad.

Una de las ventajas de la codificación en subbandas respecto a las otras técnicas en el dominio temporal es que el ruido de cuantificación introducido por cada ADPCM está localizado solo en una banda frecuencial.

6.5. Codificación basada en transformadas

La codificación basada en transformadas consiste en transformar bloques de muestras de la señal de entrada, es decir, muestras en el dominio temporal, en un dominio transformado. Después, se codifican las muestras en el dominio transformado con un codificador de forma de onda (por ejemplo, ADPCM).

Para la descodificación, primero se aplica un descodificador de forma de onda para obtener los bloques de muestras en el dominio transformado y después se aplica la transformada inversa. Como último paso, se deben recombinar los

bloques de muestras para obtener la reconstrucción de la señal de entrada. La figura 14 muestra un diagrama de bloques de un codificador y un descodificador.

Codificación basada en transformadas

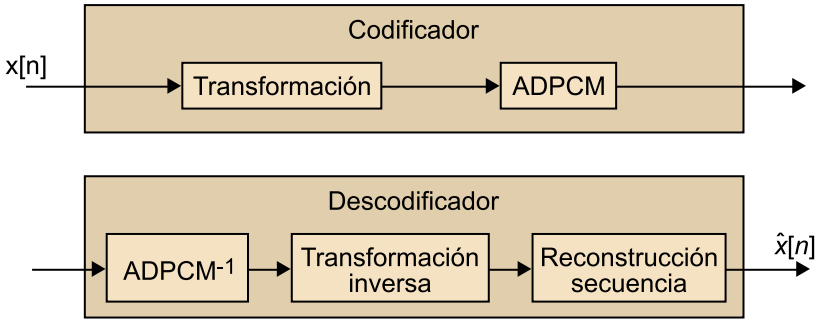


Figura 14. Diagrama de bloques de un codificador (figura superior) y un descodificador (figura inferior) basados en transformadas

Ejemplo

Un ejemplo de transformación de bloques que conocemos todos es la FFT, que toma bloques de muestras (por ejemplo, 512 o 1.024 muestras) y los transforma en coeficientes FFT, sin variar el número de muestras entre la entrada y la salida de la transformación (para el ejemplo anterior tendríamos 512 o 1.024 coeficientes de Fourier).

La transformación más utilizada para la codificación de audio es la transformada discreta de coseno o *discrete cosinus transform* (DCT) o la versión modificada, la MDCT.

Transformada discreta de coseno

Aproximación de una secuencia por medio de una suma de funciones cosenos de diferentes frecuencias:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{2}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1$$

N indica el número de muestras que se deben transformar.

La MDCT es la versión de la DCT para bloques de datos encabalgados.

Uno de los puntos críticos de este tipo de codificaciones es la selección de los bloques de datos que se han de transformar, para evitar que aparezcan distorsiones en la señal reconstruida en los límites de los bloques. Por esta razón, se utilizan ventanas "suaves", en lugar de ventanas rectangulares, con el fin de determinar los bloques de datos que se deben transformar, que tienen un comportamiento frecuencial más adecuado. Además, los bloques de datos se seleccionan encabalgados en el tiempo.

Ved también

Para recordar las propiedades de las ventanas, podéis consultar el módulo "Diseño y análisis de filtros en procesamiento de audio".

La codificación basada en transformadas se utiliza sobre todo para señal de audio y vídeo de ancho de banda grande, pero normalmente no se utiliza para voz. La voz tiene un ancho de banda pequeño, respecto a la música, por ejemplo, y se puede codificar con otras técnicas más sencillas y obtener la misma calidad que con los codificadores basados en transformadas.

Los codificadores de forma de onda intentan representar de manera compacta la forma de onda de **cualquier** señal. Por lo tanto, se utilizan para audio, imágenes, etc.

Los codificadores en forma de onda trabajan en el dominio temporal (PCM, DPCM y ADPCM) y en el dominio frecuencial (codificadores en subbandas y codificadores basados en transformadas). En ambos casos la señal se codifica sin pérdidas.

Las señales codificadas con este tipo de codificadores tienen una calidad buena para tasas de bits elevadas o moderadas (hasta 32 kbps), pero no se utilizan para codificaciones de tasas de bits bajas porque se degrada la calidad.

7. Codificadores perceptivos

Los codificadores perceptivos son codificadores que se basan en las características de percepción del sistema auditivo humano para intentar reducir el número de bits necesarios para realizar la codificación.

En concreto, los codificadores perceptivos explotan dos características de la percepción del oído humano:

- El oído humano tiene un **nivel de sensibilidad** diferente en cada banda frecuencial (como hemos visto en el apartado "Niveles audibles en función de la frecuencia. Curvas isofónicas"). Es decir, en cada banda frecuencial solo oímos los sonidos que están por encima de un umbral, y este umbral es diferente para cada banda.

- El fenómeno de **enmascaramiento**, por el cual un tono de frecuencia más baja pero con una energía alta provoca que nuestro oído no perciba un tono de frecuencia más alta con energía más baja.

Ved también

Podéis revisar el apartado "Enmascaramiento del sonido" del módulo *Introducción a la acústica*.

Las consecuencias de estas propiedades en la codificación tienen los resultados siguientes:

- No hay que codificar los contenidos frecuenciales de entrada que se encuentran por debajo del umbral de sensibilidad.

- No hay que codificar los contenidos frecuenciales de entrada que se encuentran por debajo del umbral de enmascaramiento.

- No se perciben los contenidos frecuenciales introducidos por el ruido de cuantificación que se encuentran por debajo del umbral de sensibilidad.

- No se perciben los contenidos frecuenciales introducidos por el ruido de cuantificación que se encuentran por debajo del umbral de enmascaramiento de la señal que se debe codificar.

Los codificadores perceptivos remodelan el espectro del error de cuantificación para conseguir que sea **inaudible** dada la señal que se ha de codificar. Fijaos en que en ningún momento hablamos de eliminar o minimizar el error de cuantificación, sino de convertirlo en inaudible en la señal reconstruida. De hecho, en las zonas inaudibles, el error de cuantificación introducido por los codificadores perceptivos es más elevado que para otros codificadores, como consecuencia de intentar reducir el error en bandas frecuenciales audibles.

Por lo tanto, los codificadores perceptivos son codificadores con pérdidas, es decir, la señal obtenida de la descodificación no es igual a la señal original, dado que los contenidos frecuenciales no se codifican todos del mismo modo.

La figura 15 muestra un diagrama de bloques genérico para un codificador perceptivo:

Codificación perceptiva

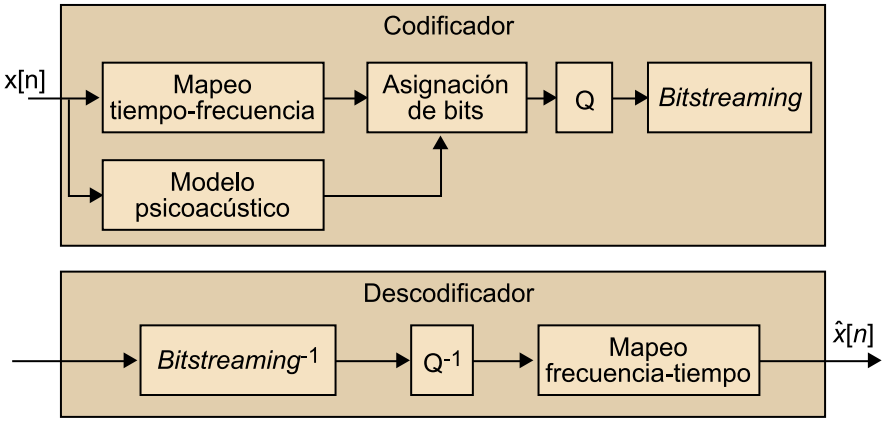


Figura 15. Diagrama de bloques de un codificador (figura superior) y un descodificador (figura inferior) perceptivo

El funcionamiento básico de un codificador perceptivo es el siguiente:

- Descomponer la señal en bandas frecuenciales por medio de un banco de filtros[3] paso banda.

- Analizar el nivel de señal en cada banda frecuencial y determinar el umbral de enmascaramiento y sensibilidad a partir de un modelo psicoacústico de la percepción.

- Determinar, a partir del modelo psicoacústico de percepción para la señal de entrada, el número de bits que se deben utilizar en cada subbanda para que el ruido de cuantificación introducido sea mínimo.

- Cuantificar las muestras de cada subbanda de manera independiente con el número de bits adecuados (módulo *Q* de la figura 15).

- Enviar los bits al descodificador (módulo *bitstreaming* de la figura 15).

(3)Un banco de filtros es un sistema formado por un conjunto de filtros paso banda situados en paralelo (es decir, la entrada de cada uno de estos filtros es la señal de entrada al banco de filtros). La salida del banco de filtros son *N* señales, donde *N* es igual al número de filtros paso banda.

A continuación, veremos las características más importantes de cada módulo de los codificadores perceptivos.

7.1. Mapeo tiempo-frecuencia

Según la psicoacústica, el sistema auditivo humano se puede modelar como una serie de filtros paso banda con las bandas de paso encabalgadas, donde el ancho de banda de cada filtro se define como *ancho de banda crítico*[4].

(4)Rango de frecuencias en torno a una frecuencia central en el que el umbral de enmascaramiento es constante (llano). Este valor se puede aproximar con la expresión siguiente:

$$\Delta f_{\text{crítica}} = 25 + 75\left(1 + 1{,}4\left(\frac{f_{\text{central}}}{1.000}\right)^2\right)^{0{,}69} \text{ Hz}$$

Los codificadores perceptivos, como intentan incorporar información psicoacústica, trabajan en el dominio de la frecuencia. Por lo tanto, el primer paso de la codificación es pasar a este dominio.

Tal como hemos comentado en el apartado de codificador de forma de onda, hay dos opciones para trabajar en el dominio frecuencial: utilizar un banco de filtros paso banda o transformar la señal con FFT o DCT.

7.2. Modelo psicoacústico

El módulo llamado *modelo psicoacústico* informa sobre los niveles de sensibilidad y los niveles de enmascaramiento dado el espectro de la señal de entrada. Es decir, dada una señal de entrada, el modelo psicoacústico nos dice cuál es el nivel de señal mínimo que distingue el oído humano. Este nivel mínimo es lo que se conoce como *umbral de enmascaramiento* de esta señal ($\sigma^2_{\text{umbral de enmascaramiento}}$).

Con la información proporcionada por este módulo se puede calcular la *signal to mask ratio* (SMR) para cada banda frecuencial. La SMR se define como la proporción de nivel entre un componente frecuencial de una señal y el umbral de enmascaramiento que genera esta frecuencia:

$$\text{SMR} = \frac{\sigma^2_{\text{señal}}}{\sigma^2_{\text{umbral de enmascaramiento}}}$$

Los valores altos de la SMR indican que la señal de entrada provoca menos enmascaramiento.

7.3. Asignación de bits

Los codificadores perceptivos asignan a cada subbanda el número mínimo de bits necesarios para codificar la señal sin introducir diferencias perceptivas respecto a la señal original.

El objetivo del módulo de asignación de bits es encontrar cuál es el número de bits necesarios en cada subbanda que minimiza el ruido **audible** introducido por el cuantificador.

El número de bits utilizados por el cuantificador en cada subbanda debe ser conocido tanto en la etapa de codificación como en la etapa de descodificación. Por lo tanto, junto con la señal cuantificada se ha de transmitir y alma-

cenar la distribución de bits utilizada. A pesar de que esto implica un aumento de la cantidad de bits generada por el codificador, la reducción del número de bits conseguida cuando se utiliza información perceptiva es mucho mayor.

Existen algunos estándares de codificación que replican alguna parte del cálculo de la asignación de bits del codificador en el descodificador para minimizar la cantidad de bits de la codificación. Sin embargo, esto implica incrementar la complejidad del descodificador y que sea más sensible a errores de transmisión.

Los codificadores perceptivos remodelan el espectro del error de cuantificación para conseguir que sea **inaudible** dada la señal que se debe codificar.

Para saber qué niveles de señal son inaudibles, se calcula el **nivel de enmascaramiento** en cada subbanda de la señal que se ha de codificar mediante un modelo psicoacústico.

Cada subbanda frecuencial de la señal de entrada se cuantifica con un número de bits diferentes. El número de bits utilizados para codificar cada subbanda se calcula para que la cantidad de ruido de cuantificación que se encuentra por encima del nivel de enmascaramiento (es decir, el ruido que podemos oír) sea mínima.

8. Codificaciones específicas para voz

La voz tiene unas características propias que la diferencian del resto del audio. Por ejemplo, su ancho de banda es más limitado que el de la música y tiene unas propiedades espectrales específicas.

Ved también

En los dos últimos módulos de este material explicaremos con más detalles las características espectrales de la voz.

Las propiedades espectrales de la voz han permitido crear unas codificaciones específicas para voz que disminuyen el número de bits necesarios. Por tanto, se reducen el ancho de banda necesario en una transmisión o la capacidad de almacenamiento necesario.

Ejemplo

Si para codificaciones estéreo de música se pueden utilizar 1.411 kbps, las codificaciones específicas para voz utilizan entre 8 y 16 kbps (codificaciones de tasa media), entre 8 y 2,4 kbps (codificaciones de tasa baja) o incluso menos de 2,4 kbps (codificaciones de tasa muy baja). Si aplicáramos las codificaciones específicas para voz en música, la calidad perceptiva no sería aceptable.

En general, los codificadores específicos para voz se clasifican en codificadores paramétricos y codificadores híbridos. A continuación mostramos un resumen de los principales codificadores de cada tipo:

- Codificadores paramétricos: vocoder[5] LPC y codificación armónica
- Codificadores híbridos: Familia CELP

(5)El término *vocoder* viene del inglés *voice coder*.

Los codificadores paramétricos están basados en el modelado de la voz, con el objetivo de codificar los parámetros de este modelo en lugar de la forma de onda de la señal. Por lo tanto, en la etapa de codificación se calculan los parámetros del modelo de voz para la señal de entrada, se cuantifican y se envían o se almacenan. Para recuperar la señal codificada, se genera la voz a partir del modelo de producción que tiene, utilizando los parámetros codificados previamente. Los codificadores paramétricos trabajan con tasas bajas de bits por segundo, de manera que generan voz inteligible pero sin mucha calidad.

Los codificadores híbridos combinan los codificadores paramétricos, específicos para voz, con los codificadores perceptivos, útiles para cualquier tipo de audio, para obtener voz de más calidad con tasas de bits bajas.

Los dos tipos de codificaciones tienen pérdidas, puesto que las señales obtenidas de la descodificación no son exactamente iguales a la señal original.

8.1. Codificadores paramétricos: vocoder LPC

El vocoder LPC utiliza el modelo LPC (codificación por predicción lineal o *linear predictive coding*) de producción de la voz para hacer la parametrización (podéis ver la figura 16). El modelo LPC está formado por una señal de excitación (denominada también *fuente*) que se filtra por medio de un filtro lineal variante con el tiempo.

Modelo de producción de la voz

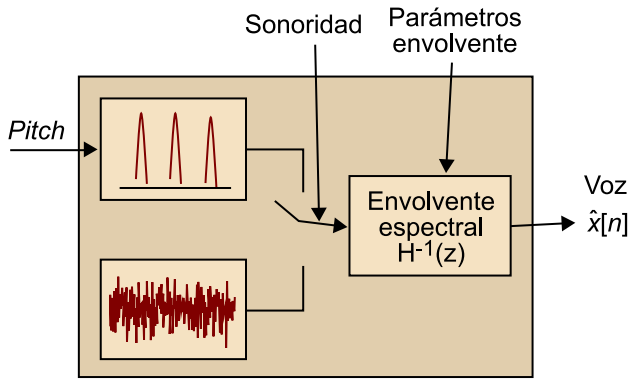


Figura 16. Modelo de producción de la voz LPC. Parámetros del modelo: la altura tonal o *pitch*, la sonoridad y la envolvente. Componentes del modelo: excitación (tren de impulsos o ruido blanco) y filtro de envolvente espectral

Ved también

Recordad que hemos visto el concepto de ruido blanco en el apartado "Espectro de sonido y densidad espectral".

La señal de excitación es de dos tipos: un tren de impulsos (para generar sonidos sonoros) o ruido blanco (para generar sonidos sordos). El filtro utilizado es un filtro IIR todo polos, del tipo siguiente:

$$H(z)=\frac{1}{1+a_1z^{-1}+a_2z^{-2}+\quad\ldots+\quad a_pz^{-p}}$$

Ved también

Recordad que hemos visto el concepto de filtro IIR en el apartado "Diseño de filtros IIR".

El filtro $H(z)$ (es decir, la envolvente espectral) se diseña para que funcione como un predictor de $x[n]$. Si la predicción $\widehat{x}[n]$ se escribe como una combinación lineal de las p muestras anteriores:

$$\widehat{x}[n]=\sum_{i=1}^{P}a_ix[n-i]$$

los coeficientes $\{a_i\}$ se calculan para minimizar el error de predicción $e[n]$:

$$e[n]=x[n]-\widehat{x}[n]=x[n]-\sum_{i=1}^{P}a_ix[n-i]$$

$$\{a_i\}^{\text{opt}}=\underset{\{a_i\}}{\operatorname{argmin}}\left(x[n]-\sum_{i=1}^{P}a_ix[n-i]\right)$$

La señal $e[n]$ también se denomina *residuo*.

En la figura 17 podéis ver un diagrama de bloques de la codificación vocoder LPC:

Codificación LPC

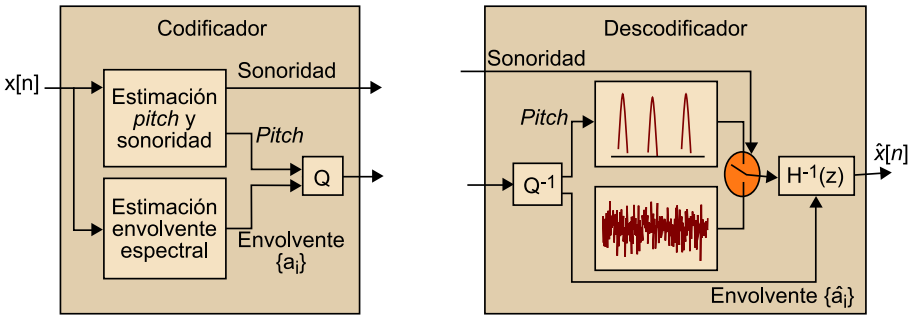


Figura 17. Codificación vocoder LPC: codificación (izquierda) y descodificación (derecha)

En la etapa de codificación, el vocoder LPC calcula tres parámetros del modelo de voz:

- La sonoridad del sonido que se debe codificar (0 para sonidos sordos y 1 para sonidos sonoros).
- En caso de tratarse de un sonido sonoro, estima la altura tonal (frecuencia del tono del sonido que se tiene que codificar).
- Los coeficientes LPC $\{a_i\}$.

El modelo de voz se calcula cada pocos milisegundos (un valor habitual es cada 20 milisegundos) para asegurar que se pueden capturar todos los cambios de la señal de voz. Finalmente, se cuantifican la sonoridad, la altura tonal y los coeficientes LPC $\{a_i\}$. Habitualmente, no se cuantifican directamente los coeficientes $\{a_i\}$, puesto que algunas pequeñas variaciones en los valores de los coeficientes descodificados, debidas al ruido de cuantificación, pueden dar lugar a filtros $H(z)$ inestables. Para evitar este efecto, se aplica alguna transformada en los coeficientes LPC para asegurar más robustez ante el ruido de cuantificación.

Fijaos en que la señal residuo $e[n]$ no se codifica. Internamente, la señal $e[n]$ se calcula durante el cálculo de los coeficientes LPC, la sonoridad y la altura tonal, pero ni se transmite ni se almacena. Los vocoders simplifican el modelo de generación de la voz utilizando como señal de excitación ruido blanco o un tren de impulsos, en lugar del residuo $e[n]$. Gracias a esta simplificación se reduce la tasa de bits del codificador.

En la etapa de descodificación, antes de nada se aplica el cuantificador inverso a la sonoridad, la altura tonal y los coeficientes LPC. Con los valores obtenidos se construye el modelo de señal y se obtiene la voz descodificada eligiendo

la señal de excitación del filtro LPC según el parámetro de sonoridad (ruido blanco si el parámetro de sonoridad es 0 o un tren de impulsos de frecuencia igual a la altura tonal si la sonoridad es 1).

La tasa de bits de los vocoders LPC es muy baja: 2,4 kbps. La voz generada por un vocoder de estas características, a pesar de ser inteligible, tiene una calidad baja (la voz suena metálica). La calidad baja se atribuye sobre todo al hecho de que el vocoder LPC no utiliza como señal de excitación la señal residuo $e[n]$, sino una señal artificial que solo prevé dos estados de sonoridad, sordo y sonoro, sin transiciones o mezclas.

8.2. Codificadores paramétricos: codificación armónica

El modelo armónico aproxima la señal de voz como una suma de componentes espectrales armónicos a la altura tonal w_0, cada uno de los cuales con una amplitud A_k y una fase φ_k diferente:

$$\hat{x}[n] = \sum_{k=0}^{T_0-1} A_k \cos\left(k w_0 n + \varphi_k\right)$$

Lectura de la fórmula

T_0 es el periodo de altura tonal.

Para los sonidos sordos, donde la altura tonal no está definida, se utiliza un valor constante de W_0, como por ejemplo 100 Hz.

La figura 18 muestra un diagrama de bloques de la codificación armónica:

Codificación armónica

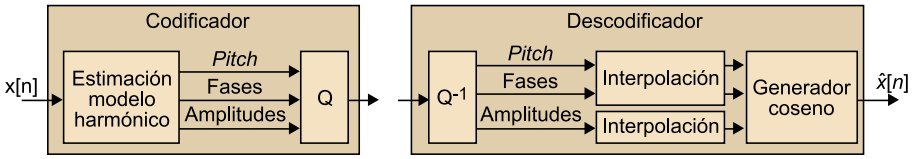


Figura 18. Codificación armónica: codificación (izquierda) y descodificación (derecha)

En el codificador se calculan los parámetros w_0, $\{A_k\}$ y $\{\varphi_k\}$ cada pocos milisegundos y se cuantifican. En el descodificador se aplica el cuantificador inverso y se genera la voz por medio del modelo armónico. Para evitar discontinuidades en la señal generada en los puntos de cambios de parámetros del modelo armónico, se aplican técnicas de interpolación matemática para obtener los valores de amplitud, frecuencia y fase de cada componente espectral.

8.3. Codificadores híbridos: *code exited linear prediction*

Los codificadores híbridos, entre los que se encuentran todos los de la familia *code exited linear prediction* (CELP), son codificadores que intentan solucionar las limitaciones de los vocoders LPC (recordemos que el vocoder es un codificador paramétrico) mediante la utilización de un codificador perceptivo. Los codificadores basados en CELP son los codificadores híbridos más utilizados en transmisión de voz.

Tal como hemos dicho antes, una de las causas principales de la pérdida de calidad de la voz codificada con un vocoder LPC es la utilización de una señal de excitación artificial muy simple (ruido blanco para sonidos sordos o un tren de impulsos para sonidos sonoros), en lugar de la señal residual $e[n]$.

La solución más directa para resolver esta pérdida de calidad de la voz es codificar las muestras de la señal residual $e[n]$ que está disponible en el módulo codificador durante el cálculo de los coeficientes LPC. La dificultad es encontrar la manera de codificar esta señal sin aumentar excesivamente el número de bits necesarios para hacerlo.

Los codificadores CELP cuantifican la señal residual $e[n]$ por medio de un cuantificador vectorial (VQ) utilizando las propiedades de la psicoacústica. Es decir, la señal residual $e[n]$ se codifica por medio de un codificador perceptivo. De este modo la codificación está formada por los mismos bits que el vocoder LCP (la cuantificación de los parámetros de sonoridad y altura tonal y los coeficientes del filtro LPC) más el índice del *codeword* correspondiente a la señal residual.

En la figura 19 se muestra un diagrama funcional de un codificador basado en CELP:

Codificación CELP

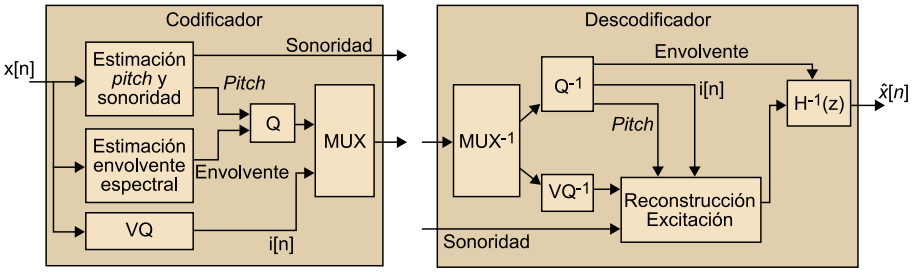


Figura 19. Codificación CELP: codificación (izquierda) y descodificación (derecha)

Hay muchas variaciones de este esquema básico: ACELP, RPE-LPC, VSELP, QCELP. Las diferencias son la manera como se trata la señal residual antes de la cuantificación (por ejemplo, en algunos casos, en lugar de trabajar con toda la señal residual, esta señal se divide en cuatro subseñales que se cuantifican por separado) y en el modo como se crea el diccionario del VQ.

Los codificadores híbridos combinan los codificadores paramétricos (tasa de bits baja) con los codificadores perceptivos para codificar la señal residual $e[n]$ (calidad elevada), para mejorar la calidad del vocoder LPC manteniendo baja la tasa de bits.

9. Formatos de ficheros de audio

En los apartados anteriores hemos visto cómo podemos codificar digitalmente el audio a partir de una señal analógica y cómo podemos obtener de nuevo la señal analógica para oírla. En este apartado veremos de qué manera se almacena el audio codificado, ya sea para reproducirlo más adelante (es decir, para descodificarlo y reproducir la señal analógica) o para transmitirlo (por ejemplo, por Internet).

El audio codificado se puede almacenar en ficheros de diferentes formatos. El formato de un fichero de audio nos indica cómo están estructurados los datos que contiene. Un fichero de audio puede almacenar tres tipos de datos: las muestras codificadas de una señal de audio, la información sobre el proceso de obtención de estas muestras (por ejemplo, frecuencia de muestreo, número de bits por muestra y tipo de codificación aplicada) y otros metadatos[6] (por ejemplo, nombre de la canción e intérprete para música). No todos los formatos de fichero permiten los tres tipos de datos.

(6)Datos sobre los datos, es decir, información complementaria sobre un tipo de datos.

Normalmente, el formato del audio está indicado en la extensión del nombre del fichero (por ejemplo, *fichero.wav* y *fichero.mp3* nos indica que el primer fichero tiene formato WAV y el segundo, MP3).

En principio, el formato es solo la **estructura** del fichero, pero hay formatos que se han diseñado pensados para contener una codificación concreta. Así, muchas veces se utiliza el mismo término para referirse al formato del fichero y al tipo de codificación del audio que contiene.

9.1. Formato de audio con forma de onda

El formato de audio con forma de onda o *waveform audio format* (WAV) fue desarrollado conjuntamente por IBM y Microsoft. Actualmente es el formato más utilizado para almacenar audio sin comprimir en las plataformas Windows.

La estructura del formato WAV es muy sencilla: una cabecera en la que se da información sobre el tipo de datos[7], seguido de los datos.

(7)La frecuencia de muestreo del audio digital, el número de canales, el número de bits por muestra, el tipo de compresión, la tasa de bits y el número de muestras totales.

A pesar de que el formato WAV permite almacenar audio codificado con compresión, normalmente los ficheros WAV contienen audio codificado con PCM y cuantificación uniforme.

El formato WAV, a pesar de utilizarse mucho en almacenamiento, no está extendido en el mundo de Internet. La razón es que, dado que al audio no se aplica ninguna compresión, la dimensión de los ficheros es demasiado grande para transmitirse con facilidad, a la inversa de lo que sucede con los formatos que prevén compresión.

Canales de audio

Un canal de audio es una señal de audio que se codifica, transmite y almacena y reproduce de manera independiente el resto de las señales de audio que permite la especificación del formato.

MPEG-3

Es importante no confundir el MPEG-3 con el MP3. El MPEG-3 es un grupo de estándares para la codificación de audio y vídeo, que actualmente no se utiliza, mientras que el MP3 es parte de los estándares MPEG-1 y MPEG -2.

9.2. MPEG-1 audio layer-3

El formato MPEG-1 *audio layer-3*, más conocido como MP3, es un formato específico para ficheros de audio incluido por el MPEG (grupo de expertos en imágenes en movimiento o *moving picture experts group*) como una parte del estándar MPEG-1 y más adelante del MPEG-2.

La estructura de los ficheros MP3 está dividida en segmentos o *frames* de la misma dimensión cada uno. Cada segmento está formado por una cabecera y un bloque de datos. La cabecera contiene seis campos: una palabra de sincronismo, la tasa de bits, la frecuencia de muestreo del audio original, el identificador de la capa o *layer* (en este caso, 3) del MPEG (puesto que las capas 1, 2 y 3 tanto del MPEG-1 como del MPEG-2 comparten la misma cabecera), el modo de codificación (mono, dual mono, estéreo o estéreo conjunto) y un valor de protección frente a copias (a pesar de que es fácil de piratear).

La principal diferencia de formato entre el MP3 y el WAV es el emplazamiento de la cabecera. El formato WAV tiene una única cabecera, que se sitúa al principio del fichero; en cambio, todos los segmentos de los ficheros MP3 tienen cabecera. Por lo tanto, se puede descodificar cualquier segmento del fichero MP3, aunque no dispongamos de la parte inicial del fichero (esto es útil si hay errores de transmisión, pérdida de paquetes, etc.).

En el campo de datos, los segmentos MP3 almacenan audio comprimido con pérdidas. Para hacer la compresión se utiliza un codificador perceptivo y, por lo tanto, los datos son los niveles de amplitud cuantificados, cada uno con el número de bits necesarios para minimizar el error perceptivo del audio reconstruido. Las frecuencias de muestreo de la señal de entrada previstas en los estándares son 32, 44,1 y 48 kHz para el MPEG-1 y 16, 22,05 y 24 kHz para el MPEG-2. La ratio de compresión viene dada por la tasa de bits requerida a la salida del codificador. Los estándares MPEG-1 y MPEG-2 *audio layer-3* permiten diferentes tasas de bits: entre 32 y 320 kbps para el MPEG-1 y entre 8 y 160 kbps para el MPEG-2. La tasa de bits más utilizada es 128 kbps.

Los estándares MPEG-1 y MPEG-2 no incluyen ninguna implementación específica del codificador. Esto ha dado libertad a diferentes fabricantes para realizar implementaciones específicas de los codificadores MP3, con la única res-

tricción de que cumplan las condiciones de formato. Como resultado de esto, un mismo fichero de audio codificado con dos codificadores diferentes puede dar calidades diferentes.

El descodificador sí que está estandarizado y, por lo tanto, el resultado obtenido a partir de un fichero MP3 con descodificadores de diferentes fabricantes siempre es el mismo.

A pesar de que las especificaciones del MP3 no incluyen ningún campo específico de metadatos, se permite incluir etiquetas o *tags* en formato ID3 al principio del fichero o al final, de una manera independiente a los segmentos MP3. Entre los campos que permite el ID3 se encuentran el título, el artista, el álbum y el año de creación del audio.

Las dimensiones reducidas de los ficheros MP3, sobre todo en comparación con los ficheros WAV, ha convertido este estándar en uno de los más utilizados para la compartición de ficheros de audio por Internet.

Lectura recomendada

Para saber más sobre el formato MP3, os recomendamos que leáis:

Karlheinz Brandenburg (1999). MP3 and AAC explained

9.3. *Advanced audio coding*

El MPEG-2 *advanced audio coding* (AAC) fue desarrollado como sucesor del formato MP3 y está incluido en los estándares MPEG-2 y MPEG-4.

El estándar AAC no define un único formato de fichero, sino que proporciona dos ejemplos: *audio data interchange format* (ADIF) y *audiodata transport stream* (ADTS).

El ADIF sitúa todos los datos de control de codificación (la frecuencia de muestreo del audio original, el modo de codificación, etc.) en una sola cabecera, al principio de todo, y los datos después (del mismo modo que hace el formato WAV). Por lo tanto, este formato es adecuado para el almacenamiento o intercambio de ficheros, pero no permite empezar la descodificación en cualquier punto del tiempo, como permite el MP3.

El ADTS utiliza la misma estructura que el MP3, es decir, el fichero se divide en segmentos, cada uno con una cabecera y un campo de datos. Para diferenciar los formatos AAC y MP3, el AAC define el parámetro de control de la capa en el valor 4. El formato ADTS se utiliza más que el ADIF porque permite empezar la descodificación en cualquier instante del audio codificado.

En el campo de datos, los segmentos AAC almacenan audio comprimido con pérdidas, tal como su predecesor MP3. Los dos estándares están basados en codificadores perceptivos. El AAC proporciona una calidad de audio codificado más elevada a la misma tasa de bits que el MP3, especialmente para tasas de bits

bajas, dado que, entre otras mejoras, utiliza un banco de filtros más eficiente para la separación en subbandas. Además, el AAC permite codificar audio de hasta cuarenta y ocho canales.

9.4. *Windows media audio*

Windows media audio (WMA) es un formato desarrollado por Microsoft como alternativa al MP3.

El formato WMA utiliza el contenedor *advanced systems format* (ASF) para encapsular audio comprimido. El ASF especifica una cabecera seguida de un conjunto o más de un conjunto de muestras de audio y, opcionalmente, un índice. En la cabecera, igual que sucede con los otros formatos, se indica qué tipo de datos y características asociadas al muestreo y la codificación se hallan en los campos siguientes. También se pueden incluir metadatos de una manera similar al ID3.

Existen cuatro codificadores diferentes dentro de la familia WMA:

- WAM original
- WAM Pro, que extiende algunas de las funcionalidades del WMA, como por ejemplo el soporte multicanal
- WMA Lossless, que se diferencia de las codificaciones anteriores porque trabaja con una compresión sin pérdidas
- WMA Voice, diseñado específicamente para codificar voz a tasas de bits bajas

9.5. *Vorbis OGG*

Vorbis es un proyecto de código abierto que especifica un codificador perceptivo, y por tanto con pérdidas, liderado por Xiph.org Foundation. Normalmente Vorbis utiliza el contenedor OGG, de manera que los ficheros codificados y almacenados se denominan *Vorbis OGG*.

Los ficheros OGG están estructurados en segmentos de datos, denominados *páginas OGG*. Cada página contiene una cabecera y un campo de datos. El formato de los metadatos, denominado *Vorbis comments*, está basado en etiquetas, como el ID3.

Vorbis OGG es popular sobre todo entre los seguidores del software libre.