TRATAMIENTO DIGITAL DE SEÑALES EN COMUNICACIONES

# Práctica 5. Codificación con transformadas

**Resumen**

En esta práctica estudiaremos la propiedad de compactación de las transformadas que se usan en codificación. También observaremos el efecto de filtrado que tiene la codificación con transformadas en una secuencia de audio.

## 1. Introducción

En este sección recordaremos los aspectos básicos de la codificación con transformadas. Para codificar con transformadas una secuencia $x[n]$ damos los siguientes pasos:

1. Expandimos $x[n]$ para que el número de muestras sea múltiplo del tamaño de bloque. El resultado de esta operación es la secuencia expandida $x_{\mathrm{E}}[n]$ (figura 1).

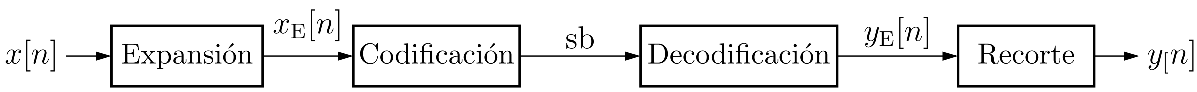


Figura 1. Expansión y recorte.

2. Codificamos $x_{\mathrm{E}}[n]$, dividiéndola en bloques de $N$ muestras y codificando por separado cada bloque. Para codificar un bloque $\underline{x}$, realizamos las siguientes operaciones (figura 2):
   2.1 Aplicamos la transformada a $\underline{x}$, obteniendo sus coeficientes $\underline{c}$;
   2.2 Cuantizamos los coeficientes, obteniendo los índices de cuantificación $\underline{i}$;
   2.3 Codificamos sin pérdidas los índices, obteniendo los bits del bloque.

   La concatenación de los bits obtenidos al codificar todos los bloques constituye la secuencia de bits (sb) a transmitir o almacenar.

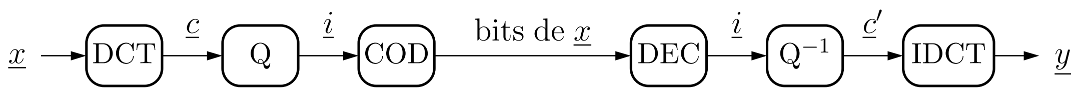


Figura 2. Codificación y decodificación de un bloque $\underline{x}$.

Para decodificar realizaremos los siguientes pasos (figura 2):

1. Decodificamos la secuencia de bits sb, obteniendo la secuencia $y_E[n]$. Para ello se decodifica cada bloque siguiendo los siguientes pasos:
   1. Se decodifican los bits del bloque, generando los índices de cuantificación $\underline{i}$;
   2. Se decuantizan los índices, generando los coeficientes $\underline{c}'$;
   3. Se aplica la transformada inversa a $\underline{c}'$, generando el bloque de muestras $\underline{y}$.
2. Concatenamos los bloques decodificados y eliminamos las muestras que añadió el codificador, obteniendo la secuencia decodificada $y[n]$.

En esta práctica supondremos que la codificación de $x[n]$ tiene las siguientes características:

– La transformada usada es la transformada discreta del coseno (DCT) de 64 muestras.

– La expansión de $x[n]$ se realiza añadiendo ceros al final de $x[n]$.

## 2. Compactación

Supongamos que la división en bloques de $x[n]$ que realiza el codificador proporciona la secuencia de bloques $\underline{x}^{(1)}, \underline{x}^{(2)}, \ldots, \underline{x}^{(N_b)}$. Podemos considerar que los bloques de muestras $\underline{x}^{(m)}$ son realizaciones de un vector aleatorio $\underline{X}$ y que los bloques de coeficientes que obtiene el codificador, $\underline{c}^{(1)}, \underline{c}^{(2)}, \ldots, \underline{c}^{(N_b)}$, son realizaciones de un vector aleatorio $\underline{C}$.

La codificación con transformadas aprovecha el hecho de que la desigualdad entre las varianzas de las variables aleatorias de $\underline{C}$ suele ser mayor que la desigualdad entre las varianzas de las variables aleatorias de $\underline{X}$. Esta propiedad se llama **compactación**.

Podemos cuantificar el grado de compactación que hay en un vector aleatorio mediante la **ganancia** $G$ definida como

$$G = \frac{\dfrac{1}{N}\displaystyle\sum_{i=0}^{N-1}\sigma_i^2}{\left[\displaystyle\prod_{i=0}^{N-1}\sigma_i^2\right]^{1/N}} \tag{1}$$

donde $\sigma_i^2$ es la varianza de la $i$-ésima variable aleatoria del vector. Esto es, la ganancia es el cociente entre la media aritmética (MA) y la media geométrica (MG) de las varianzas $\sigma_0^2, \ldots, \sigma_{N-1}^2$ ($G$ = MA/MG). Cuanto mayor es $G$, mayor es el grado de compactación (o desigualdad entre varianzas). La menor compactación se da cuando todas las varianzas son iguales, en cuyo caso MA = MG y $G = 1$. Tener una alto grado de compactación es importante en codificación con transformadas porque puede demostrarse que **cuanto mayor es el grado de compactación en $\underline{C}$, más eficiente es la codificación con transformadas de $x[n]$**.

✎ Vamos a comparar la compactación que tenemos en $\underline{X}$ con la compactación que tenemos en $\underline{C}$. Para ello, escribe un script de MATLAB en el que a partir de la secuencia de voz $x[n]$ de `v1.wav` y teniendo en cuenta las características del codificador de la sección anterior:

- Obtenga las varianzas de las variables aleatorias de $\underline{X}$
- Obtenga las varianzas de las variables aleatorias de $\underline{C}$
- Muestre ambos conjuntos de varianzas en dos gráficas (usa `subplot`)
- Obtenga la ganancia del vector $\underline{X}$ y la del vector $\underline{C}$

Al escribir el script ten en cuenta las siguientes consideraciones:

– Obtén $x_{\mathrm{E}}[n]$ añadiendo ceros al final de $x[n]$.

– No utilices ningún bucle. Para ello, transforma $x_{\mathrm{E}}[n]$ en una matriz $64 \times N_b$ usando `reshape` (cada columna de la matriz debe ser un bloque de $N$ muestras).

  Obtén la transformada de todos los bloques con una única instrucción `dct` aplicada a la matriz de muestras.

– Obtén la ganancia a partir de un vector de varianzas con la ayuda de las funciones `mean` y `geomen`.

Observa en los resultados que has obtenido que:

– La compactación en $\underline{X}$ es muy pequeña.

– La compactación en $\underline{C}$ es mucho mayor que en $\underline{X}$.

– La varianza de los coeficientes de alta frecuencia es muy pequeña.

Verifica que obtenemos conclusiones similares con otras secuencias de voz.

✎ Si $x[n]$ fuera una realización de una secuencia aleatoria estacionaria, ¿cuál sería el valor de la ganancia de $\underline{X}$?

## 3. Efecto de la cuantificación sobre las frecuencias altas

Supongamos que queremos conocer el bloque decodificado $\underline{y}$ que se genera al codificar y decodificar un bloque de muestras $\underline{x}$ (figura 3.a). Como no estamos interesados en la secuencia de bits que generamos al codificar, podemos eliminar los bloques COD-DEC y obtener así el diagrama simplificado de la figura 3.b. También podemos fundir los bloques Q y $\mathrm{Q}^{-1}$ en un único bloque de cuantificación C que transforma el bloque de coeficientes $\underline{c}$ en el bloque de coeficientes $\underline{c}'$ que se usa para sintetizar $\underline{y}$ (figura 3.c).

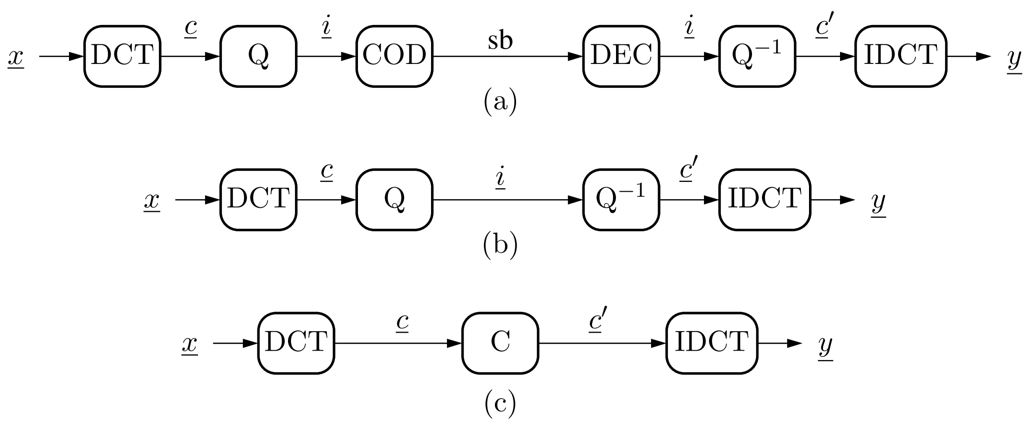


Figura 3. Codificación y decodificación de un bloque $\underline{x}$.

Supongamos que cada coeficiente se cuantifica con un cuantificador uniforme de medio paso. Sea $\Delta_k$ el paso de cuantificación utilizado en la cuantificación de los coeficientes con índice frecuencial $k$. Si

$$-\frac{\Delta_k}{2} < c_k^{(m)} < \frac{\Delta_k}{2} \tag{2}$$

entonces $c_k^{(m)'} = 0$, y consecuentemente, la codificación elimina la $k$-ésima componente frecuencial de $\underline{y}^{(m)}$. Por tanto, uno de los efectos de la codificación con transformadas es el de eliminar algunas componentes frecuenciales de los bloques de la señal. Como hemos visto en la sección anterior, la mayor parte de los coeficientes de alta frecuencia tienen una varianza muy pequeña por lo que cabe esperar que la codificación tenga un efecto de «filtrado pasobajo». No obstante, este efecto depende tanto de la frecuencia del coeficiente como de su amplitud: si $c_k^{(m)}$ es de alta frecuencia pero $|c_k^{(m)}| > \frac{\Delta}{2}$, no se eliminará la componente $k$-ésima del bloque $m$-ésimo.

A continuación comprobaremos este efecto usando MATLAB. Supón que la secuencia de voz `v1.wav` se divide en bloques de 64 muestras (añadiendo ceros al final si es necesario) y que sobre cada bloque $\underline{x}$ se realizan la siguientes operaciones:

1. Se calcula la DCT(64) del bloque $\underline{x}$, generando un bloque de coeficientes $\underline{c}$.
2. Se genera un nuevo bloque de coeficientes $\underline{c}'$ de forma que

$$c'_k = \begin{cases} c_k & \text{si } k = 0, \dots, M-1 \\ 0 & \text{si } k = M, \dots, N-1. \end{cases}$$

3. Se calcula la IDCT(64) de $\underline{c}'$, generando el bloque $\underline{y}$.

Tras concatenar los bloques $\underline{y}$ y eliminar las muestras finales que se habían añadido, obtenemos una secuencia $y[n]$.

✎ Escribe un script de MATLAB que, sin usar bucles, genere $y[n]$ y muestre la relación señal a ruido en dB. Obtén y escucha la secuencia $y[n]$ para $M = 48$, $M = 32$, $M = 16$ y $M = 8$. Comprueba que al disminuir $M$, la voz pierde brillo puesto que estamos eliminando componentes espectrales por encima de una determinada frecuencia (cuanto menor es $M$, menor es dicha frecuencia).

Aparte de la pérdida de brillo, hay otra degradación que es especialmente molesta cuando $M \leq 16$. Esta degradación no aparece cuando reducimos el ancho de banda de la señal utilizando un filtro lineal (por muy grande que sea la reducción) y es una consecuencia del tipo de procesamiento por bloques que realizamos.