

# Procesamiento digital de señales de audio

## Codificación de audio

Instituto de Ingeniería Eléctrica, Facultad de Ingeniería  
Universidad de la República, Uruguay

### Grupo de Procesamiento de Audio

- 1 Introducción
- 2 Cuantización
- 3 Asignación de bits
- 4 Codificación por entropía
- 5 Predicción lineal en codificación
- 6 Análisis tiempo-frecuencia

# Codificación de audio

## Motivación

- **contexto:** almacenamiento y transmisión de audio
- **aplicaciones:** compresión, criptografía
- **motivación:** CD/DAT PCM, 44.1 kHz, 16 bits  
estéreo: 1.41 Mb/s, mono: 705.6 kb/s

## Codificación de voz

- **aplicaciones:** telefonía, radio, teleconferencia, etc.
- **objetivo:** comunicación verbal, inteligibilidad
- **ancho banda:** 4 kHz, banda ancha 8 kHz, super/ultra ancha  $\geq 16$  kHz

## Codificación de audio

- **aplicaciones:** cine, radio, televisión, audio hi-fi, video-juegos, VR, etc.
- **objetivo:** transparencia (perceptivamente igual al original)
- **ancho banda:** 20 kHz o más

# Compresión de audio

## Sin pérdidas

- se puede obtener el original bit-a-bit
- explota redundancia (DPCM, entropía)
- estado del arte: de 2:1 a 4:1

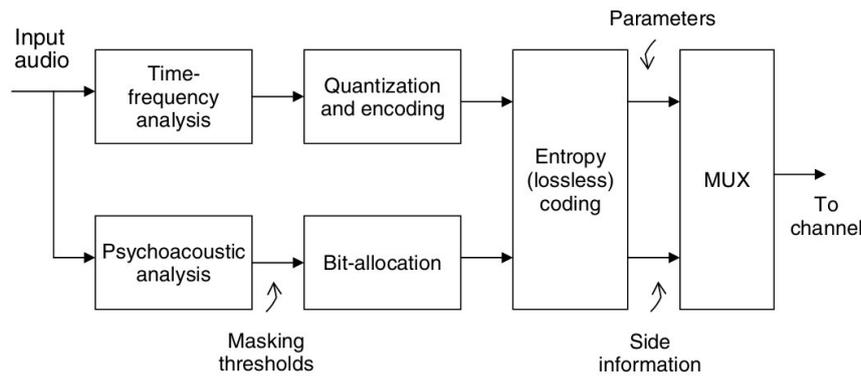
## Con pérdidas

- hay pérdida de información
- explota irrelevancia perceptiva (modelos psicoacústicos)
- estado del arte: de 10:1 a 25:1

## Ejemplos

- sin pérdida: **FLAC** <http://flac.sourceforge.net/>
- con pérdida: **ogg** <https://xiph.org/vorbis/>  
> `flac input-file.wav -o output-file.flac`  
> `oggenc -b 192 input-file.wav -o output-file.ogg`

# Codificador perceptivo genérico



[Spanias et al., 2007]

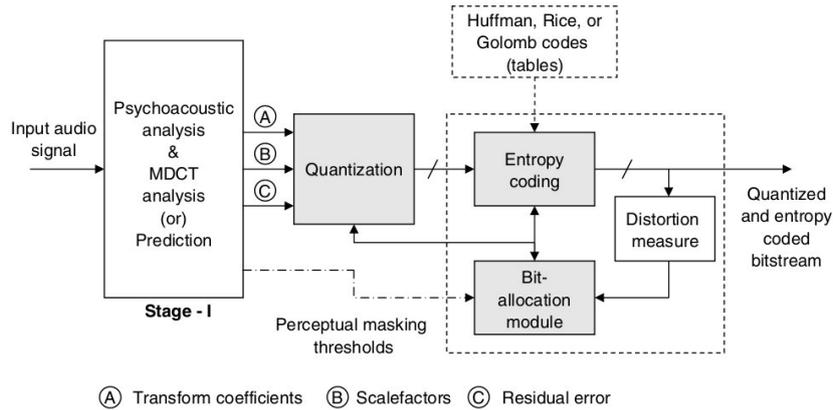
- análisis tiempo-frecuencia: estimación de componentes en cada frame
- modelo psicoacústico: aprovechar irrelevancia perceptiva
- cuantización también puede explotar redundancia (e.g. DPCM)
- quitar más redundancia con codificación por entropía (e.g. Huffman)
- control psicoacústico variante en el tiempo, i.e. tasa variable  
tasa fija con esquemas de buffer, lo que introduce retardos

## Atributos de un codificador de audio

El objetivo general de un codificador de audio es lograr alta calidad (transparencia) a baja tasa de bits ( $< 32$  kb/s), con un retardo aceptable ( $\sim 5$  a  $10$  ms) y con baja complejidad computacional ( $\sim 1$  a  $10$  MIPS).

- **calidad de audio:** se han propuesto varias medidas de calidad subjetivas y objetivas, por ejemplo: Noise-Mask-Ratio (NMR), perceptual audio quality measure (PAQM) y perceptual evaluation (PERCEVAL) [Spanias et al., 2007].
- **tasa de bits:** baja tasa de bits implica alta compresión y en general baja calidad de reproducción. Los primeros codificadores usaban altas tasas para obtener audio transparente (e.g. Dolby AC-3 32-384 kb/s, MPEG-1 32-448 kb/s), mientras que los más modernos permiten tasas entre 8 y 32 kb/s (e.g. MPEG-4).
- **complejidad:** se busca tiempo real y bajo consumo de energía. Se mide en MIPS y depende de la plataforma. Complejidad asimétrica (encoder 80%, decoder 20%)
- **retardo:** depende de la aplicación, e.g. streaming y audio-on-demand son tolerantes a retardo, mientras que en VOIP es importante bajo retardo (10-20 ms)
- **robustez frente a errores:** manejar canal ruidoso y variante en el tiempo

# Cuantización, asignación de bits y codificación por entropía

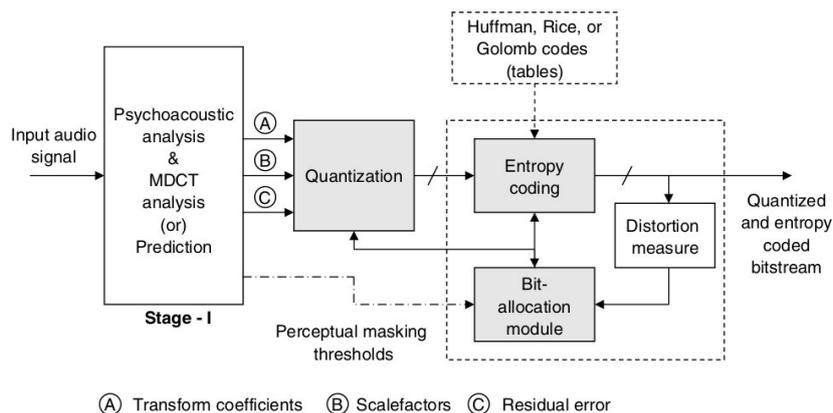


[Spanias et al., 2007]

En una primera etapa el audio es procesado usando por ejemplo alguna transformada de tiempo corto (STFT, MDCT), un banco de filtros ó LPC y se determinan umbrales perceptivos a través de un análisis psicoacústico.

La salida de esta etapa típicamente comprende: coeficientes de la transformada, factores de escala y el residuo. Es lo que hay que codificar.

# Cuantización, asignación de bits y codificación por entropía

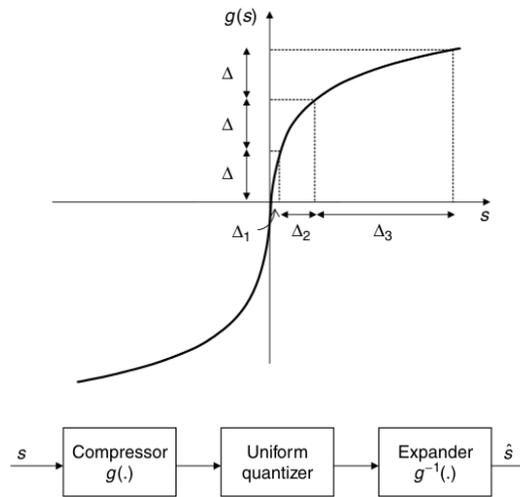


[Spanias et al., 2007]

- Los parámetros obtenidos son cuantizados (e.g. PCM, DPCM, VQ)
- La cantidad de bits usados por frame está determinada por un módulo de asignación de bits que usa umbrales de enmascaramiento
- Un esquema de codificación por entropía permite reducir redundancia
- Por último se calcula una medida de distorsión y se compara con un umbral establecido para determinar si es necesario asignar más bits

# Cuantización escalar

- **cuantización uniforme** (e.g. PCM uniforme)
  - ▶ sin memoria, no explota redundancia, ni valores más frecuentes
- **cuantización no-uniforme** (e.g. ley- $\mu$ , ley-A)
  - ▶ lineal para amplitudes pequeñas y logarítmico para amplitudes grandes
  - ▶ permite obtener mejor SNR, principalmente para bajas amplitudes
  - ▶ proceso de compresión - cuantización uniforme - expansión



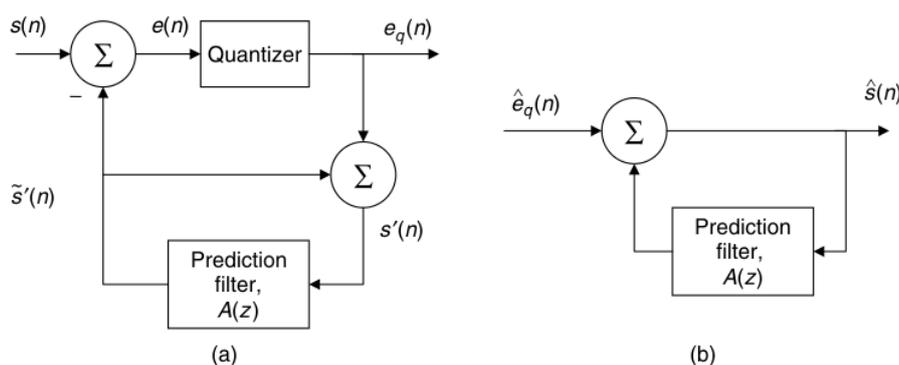
[Spanias et al., 2007]

# Cuantización escalar

- **PCM diferencial (DPCM)**
  - ▶ reduce redundancia explotando correlación entre muestras adyacentes
  - ▶ e.g. diferencia entre muestras sucesivas, e integración en receptor
  - ▶ más comunmente: predicción de tiempo-corto variante en el tiempo,

$$A(z) = \sum_{i=1}^p a_i z^{-i} \quad \tilde{s}'(n) = \sum_{i=1}^p a_i s'(n-i)$$

$a_i$  coeficientes de predicción,  $\tilde{s}'(n)$  predicción de la muestra actual

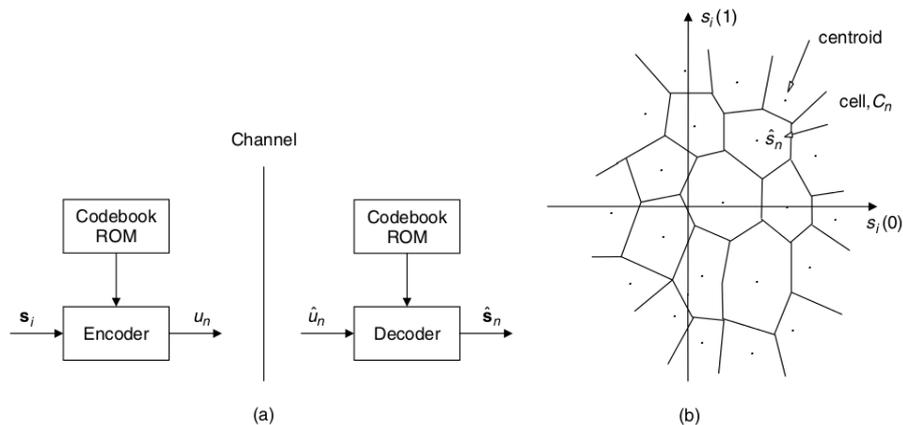


DPCM: a) transmisor, b) receptor [Spanias et al., 2007]

# Cuantización vectorial

- Vector quantization (VQ)

- ▶ codificación conjunta de un bloque o vector de datos
- ▶ los vectores de entrada  $s_i = [s_i(0), s_i(1), \dots, s_i(N-1)]^T$  se comparan con un **codebook**  
 $\hat{s}_n = [\hat{s}_n(0), \hat{s}_n(1), \dots, \hat{s}_n(N-1)]^T, n = 1, 2, \dots, L$   
 se transmite el índice de la palabra más cercana



Cuantización vectorial: a) diagrama de bloques, b) celdas para N=2 [Spanias et al., 2007]

## Asignación de bits

- determinar el número de bits necesarios para cuantizar con mínima distorsión audible un frame de audio (coeficientes de transformada, factores de escala y residuo)
- e.g. coeficientes de la transformada de un frame

$$x = [x_1, x_2, \dots, x_{N_f}]^T$$

$N_f$  número total de coeficientes

$N$  número total de bits disponibles, distribuirlos en los  $n_i$  con  $i = 1, 2, \dots, N_f$

$$\min_{n_i} \{D\} = \min_{n_i} \left\{ \frac{1}{N_f} \sum_{i=1}^{N_f} E[(x_i - \hat{x}_i)^2] \right\} \quad \text{sujeto a} \quad \sum_{i=1}^{N_f} n_i \leq N$$

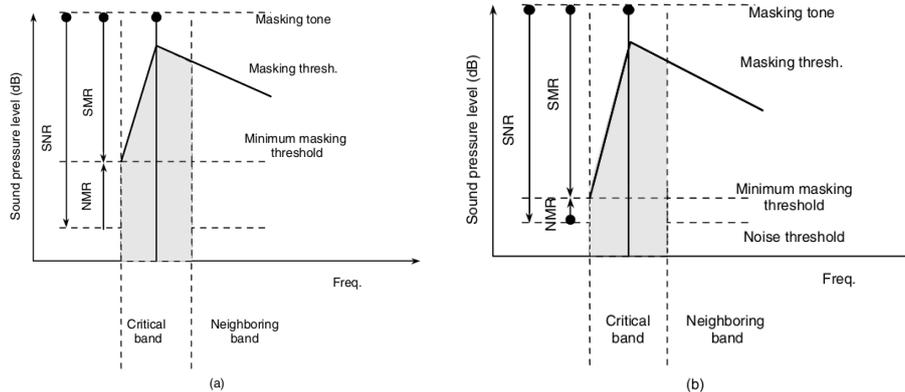
## Asignación perceptual de bits

- número de bits en cada banda de frecuencia tal que ruido de cuantización por debajo de **signal-to-mask ratio** (SMR)
- el **noise-to-mask ratio** (NMR) se obtiene en cada banda como,

$$NMR = SNR - SMR \text{ (dB)}$$

asignar suficiente número de bits a la banda con la menor NMR

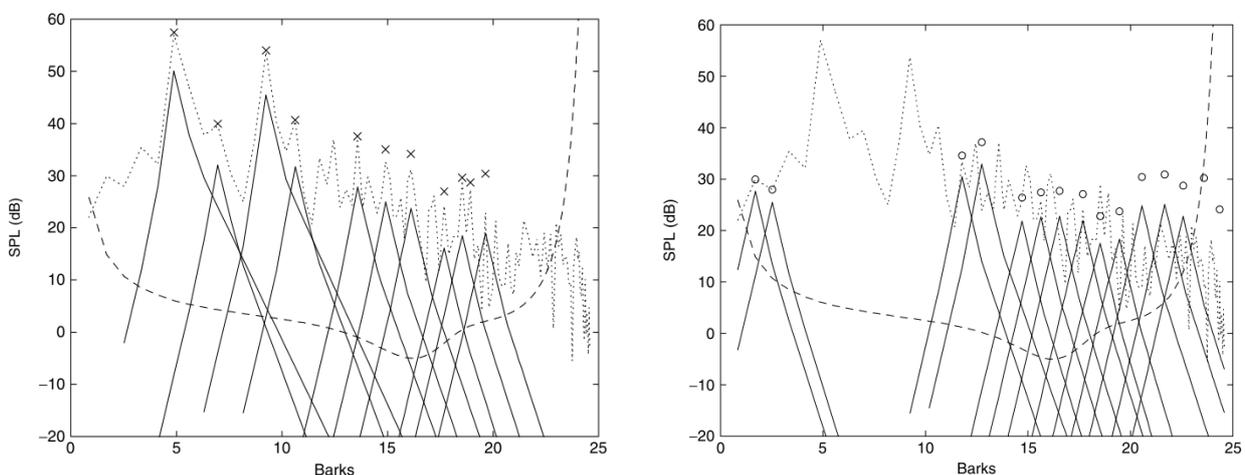
- típicamente se sigue un proceso iterativo que satisface a la vez el **bit-rate** y los requerimientos de los **umbrales de enmascaramiento**



Enmascaramiento, NMR mayor en a) que en b) [Spanias et al., 2007]

## Asignación perceptual de bits

- cálculo de umbrales de enmascaramiento globales [Spanias et al., 2007]
  - 1 análisis espectral y normalización SPL
  - 2 identificación de componentes enmascarantes tonales y de ruido
  - 3 decimación y reorganización de enmascarantes
  - 4 cálculo de umbrales individuales de enmascaramiento
  - 5 cálculo de umbral global de enmascaramiento



[Spanias et al., 2007]

## Codificación por entropía

¿Cuál es el mínimo número de bits necesario para codificar un mensaje?

Consideremos una fuente de mensajes de un alfabeto  $\mathcal{X}$ ,

$$\mathcal{X} = \{x_1, \dots, x_m\} \text{ con probabilidades } p_X(x_i)$$

El teorema de codificación de fuente de Shannon establece que no se puede comprimir más allá de la entropía [Cover and Thomas, 1991],

$$H(X) = - \sum_{i=1}^m p_X(x_i) \log_2(p_X(x_i))$$

A cada uno de los mensajes se le asigna una palabra de código  $C(x_i)$ .  
¿Cómo asignar las palabras de código de forma *óptima* y sistemática?

Sección basada en transparencias del curso *Introducción a la Teoría de la Información*

## Codificación de fuente

Sea  $\mathcal{D} = \{0, 1, \dots, D - 1\}$  un alfabeto código  $D$ -ario.

$$\mathcal{D}^* = \bigcup_{k=1}^{\infty} \mathcal{D}^k$$

con  $d^k = d_1 d_2 \dots d_k$  con  $d_i \in \mathcal{D}$ , es el conjunto de posibles palabras que pueden formarse con los elementos de  $\mathcal{D}$ .

### Definición (Código fuente)

*Código fuente*  $C$  para variable aleatoria  $X$ : mapeo de  $\mathcal{X}$  en  $\mathcal{D}^*$ , i.e.,

$$C : \mathcal{X} \rightarrow \mathcal{D}^*$$

- Cada mensaje  $x_i \rightarrow$  palabra de código  $C(x_i)$  de largo  $l(x_i) = l_i$ .

### Ejemplo

$\mathcal{X} = \{x_1, x_2\}$  con alfabeto  $\mathcal{D} = \mathcal{B} = \{0, 1\}$ ,  $C(x_1) = 00$  y  $C(x_2) = 11$

## Codificación de fuente

### Definición (Largo medio de un código)

El *largo medio*  $L(C)$  de un código  $C(x)$  para una variable aleatoria  $X$  con distribución de probabilidad  $p(x)$  se define como

$$L(C) = \sum_{x \in \mathcal{X}} p(x)l(x)$$

donde  $l(x)$  es el largo de la palabra de código asignada a  $x$ .

### Ejemplo

- $p_X(\mathcal{X}) = \left\{ \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8} \right\}$  con  $C(X) = \{0, 10, 110, 111\}$ ,  
 $L(C) = H(X) = 1.75$  bits
- $p_X(\mathcal{X}) = \left\{ \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right\}$  con  $C(X) = \{0, 10, 11\}$ ,  
 $L(C) = 1.66 > H(X) = 1.58$  bits

## Clasificación de códigos

### Definición (Código no singular)

Si cada elemento de  $\mathcal{X}$  se mapea en una palabra de código diferente,

$$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)$$

### Definición (Extensión de un código)

Mapeo de una secuencia de símbolos de  $\mathcal{X}$  en un secuencia de  $\mathcal{D}$ ,

$$C(x_1 x_2 \dots x_n) = C(x_1) C(x_2) \dots C(x_n),$$

donde  $C(x_1) C(x_2) \dots C(x_n)$  es la concatenación de las palabras de código.

### Definición (Código unívocamente decodificable)

Un código es *unívocamente decodificable* si su extensión es no singular.

No hay ambigüedades al momento de decodificar una secuencia codificada.

### Definición (Código instantáneo o de prefijo)

Si ninguna palabra de código es prefijo de otra palabra de código.

# Clasificación de códigos

## Ejemplo

$\mathcal{X}$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
$x_1$	0	0	10	0	0
$x_2$	0	010	00	10	10
$x_3$	0	01	11	110	110
$x_4$	0	10	110	1110	111

- $C_1$  es singular, no sirve para mucho
- $C_2$  no es unívocamente decodificable (UD), la secuencia 010 puede decodificarse como  $x_2$ ,  $x_1x_4$  o  $x_3x_1$
- $C_3$  es UD pero no es instantáneo.  
Si se recibe 0010...11...111..110..1101..1100.. se decodifica  
 $x_2x_1x_3x_3..x_3x_4..x_3x_2$
- $C_4$  es instantáneo, es un código de puntuación (de coma), el 0 marca el final de la palabra, ¿es eficiente?

Si se recibe 01101110 se decodifica  $x_1x_1x_1$

- $C_5$  es instantáneo.

## Desigualdad de Kraft

### Teorema (Desigualdad de Kraft)

Para todo código instantáneo sobre un alfabeto de tamaño  $D$  y largos de palabra  $l(x_1), l(x_2), \dots, l(x_m)$  se debe cumplir

$$K = \sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1$$

Igualmente dado un conjunto de largos de código que cumple la desigualdad, existe un código instantáneo con esos largos.

- Las longitudes de las palabras no pueden ser todas “cortas”, si hay una muy corta debe haber otras más largas.
- No especifica cómo asignar los largos.



## Códigos instantáneos óptimos (o cómo buscarlos)

Se puede plantear como un problema de optimización

$$\min_{l_1, l_2, \dots, l_m} L = \sum_i p_i l_i \text{ restringido a (Kraft) } \sum_i D^{-l_i} = 1$$

$$J = \sum_i p_i l_i + \lambda \left( \sum_i D^{-l_i} - 1 \right) \Rightarrow D^{-l_i} = \frac{p_i}{\lambda \ln D}$$

Usando la restricción se llega a  $\lambda = 1/\ln D$  y  $p_i = D^{-l_i}$ , dando los largos

$$l_i^* = -\log_D p_i$$

Con estos largos óptimos el largo medio queda

$$L^* = \sum_i p_i l_i^* = -\sum_i p_i \log_D p_i = H_D(X)$$

Estos largos óptimos no tienen que ser enteros, en la práctica sí.

¿Es un mínimo global?

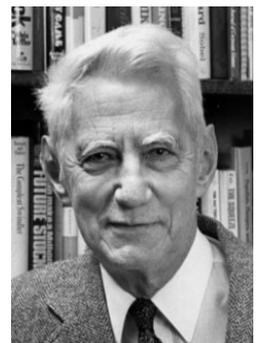
## Primer Teorema de Shannon

### Teorema (Teorema de Codificación de Fuente)

*El largo medio de un código  $C$  instantáneo,  $D$ -ario para una variable aleatoria  $X$  es mayor o igual a la entropía*

$$L(C) \geq H_D(X)$$

*y la igualdad se cumple sii  $p_i = D^{-l_i}$*



- El procedimiento para encontrar el código óptimo sería hallar la distribución  $D$ -ádica más cercana a la distribución de  $X$ . Pero esto no siempre es fácil.
- Es una cota de la longitud media para la descripción de una fuente, “no se puede comprimir más allá de la entropía”.

## Códigos de Shannon-Fano

Es un procedimiento subóptimo para construir un código, que alcanza una cota de  $L(C) \leq H(X) + 2$



- Ordenar las probabilidades en forma decreciente.
- Elegir  $k$  tal que  $|\sum_{i=1}^k p_i - \sum_{i=k+1}^m p_i|$  sea mínima
- Asignar un bit (diferente) a cada uno de los subconjuntos (ceranos a equiprobables) en que se divide la fuente.
- Repetir el procedimiento para todos los subconjuntos.

### Ejemplo

$p_i$	1	2	3	$c_i$
0.25	0	0		00
0.25	0	1		01
0.2	1	0		10
0.15	1	1	0	110
0.15	1	1	1	111

## Códigos de Huffman

Un código instantáneo óptimo (mínima  $\sum p_i l_i$ ) para una distribución dada puede ser construido con un procedimiento recursivo que en cada paso agrupa los  $D$  símbolos menos probables para formar un nuevo símbolo, propuesto por David A. Huffman en 1952.



### Ejemplo

$\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5\}$ , con probabilidades  $p = \{0.25, 0.25, 0.2, 0.15, 0.15\}$

$l_i$	$C(x_i)$	$x_i$	$p(x_i)$
2	01	1	0.25 <sup>01</sup>
2	10	2	0.25 <sup>10</sup>
2	11	3	0.2 <sup>11</sup>
3	000	4	0.15 <sup>000</sup>
3	001	5	0.15 <sup>001</sup>

Diagram illustrating the Huffman tree construction process. The root node is 1. It branches into 0.3<sup>00</sup> and 0.55<sup>0</sup>. The 0.3<sup>00</sup> node branches into 0.25<sup>01</sup> and 0.25<sup>10</sup>. The 0.55<sup>0</sup> node branches into 0.45<sup>1</sup> and 0.3<sup>00</sup>. The 0.45<sup>1</sup> node branches into 0.25<sup>11</sup> and 0.25<sup>10</sup>. The 0.3<sup>00</sup> node branches into 0.25<sup>10</sup> and 0.25<sup>01</sup>. The 0.25<sup>11</sup> node branches into 0.15<sup>000</sup> and 0.15<sup>001</sup>. The 0.25<sup>10</sup> node branches into 0.25<sup>10</sup> and 0.25<sup>01</sup>. The 0.25<sup>01</sup> node branches into 0.25<sup>01</sup> and 0.25<sup>10</sup>.

# Predicción lineal en codificación

- incluido en varios estándares de telefonía y multimedia (e.g. G.729 8kb/s CS-ACELP 1995, MPEG-4 version 2 ISO/IEC 2000)
- codificación explota la correlación de corto y largo plazo
  - ▶ corto plazo: predicción lineal de muestra actual (LP)
  - ▶ largo plazo: correlación del residuo de LP (long-term prediction, LTP)
- puede funcionar en lazo abierto o cerrado
  - ▶ en lazo cerrado se minimiza la diferencia entre la señal original y la reconstrucción ponderada *perceptivamente*
- hay extensiones de LP que incorporan modelos perceptivos
  - ▶ Perceptual LP (PLP), banco de filtros *auditivos*
  - ▶ warped LP (WLP), mapeo del eje de frecuencias a escala Bark

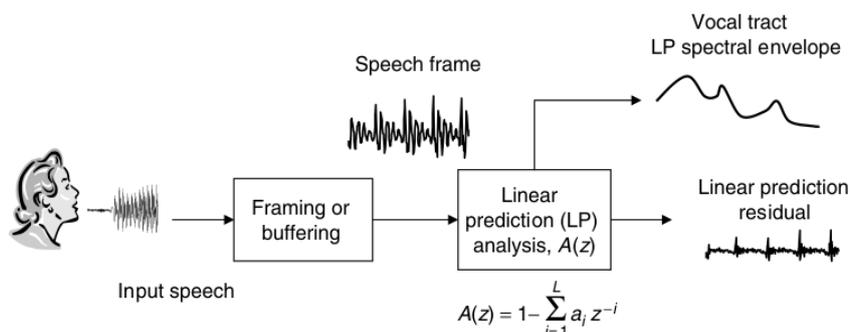
## Predicción lineal (de corto plazo)

- modelo todo-polos:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} \quad s[n] = \sum_{k=1}^p a_k s[n-k] + Gu[n]$$

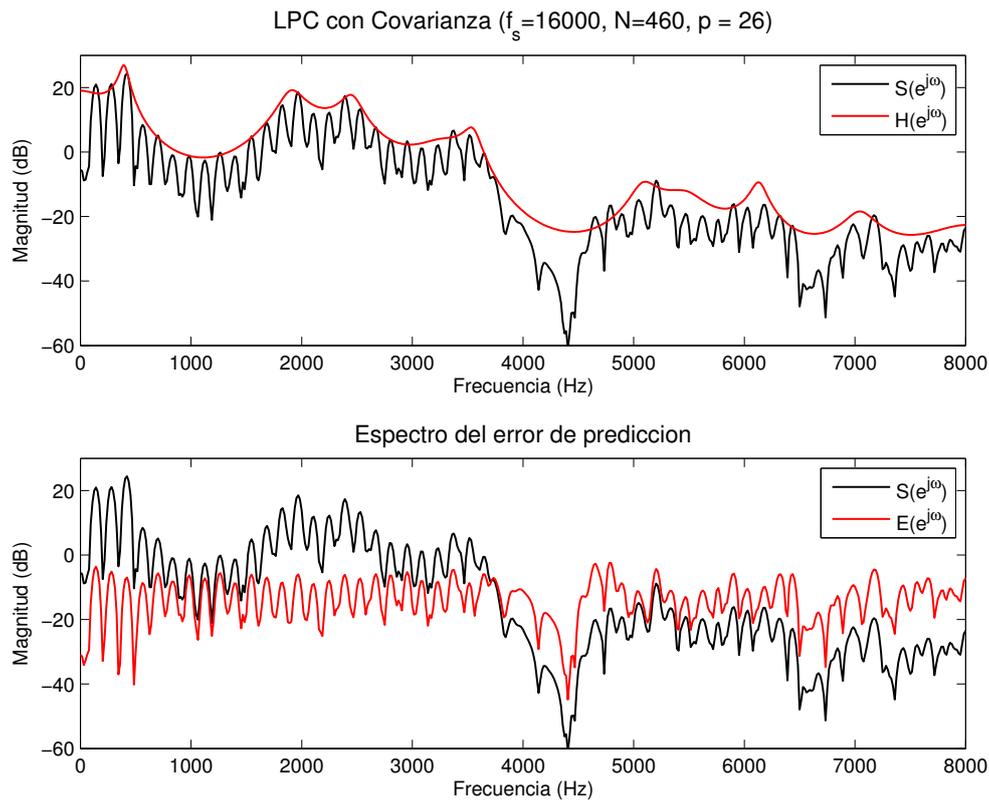
- filtro inverso:

$$H(z) = \frac{G}{A(z)}, \quad A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^p \alpha_k z^{-k}$$



[Spanias et al., 2007]

# Predicción lineal (de corto plazo)



# Predicción de largo plazo (LTP)

busca capturar la correlación presente en el residuo

predicción:

$$\hat{e}[n] = a_D e[n - D]$$

error de predicción:

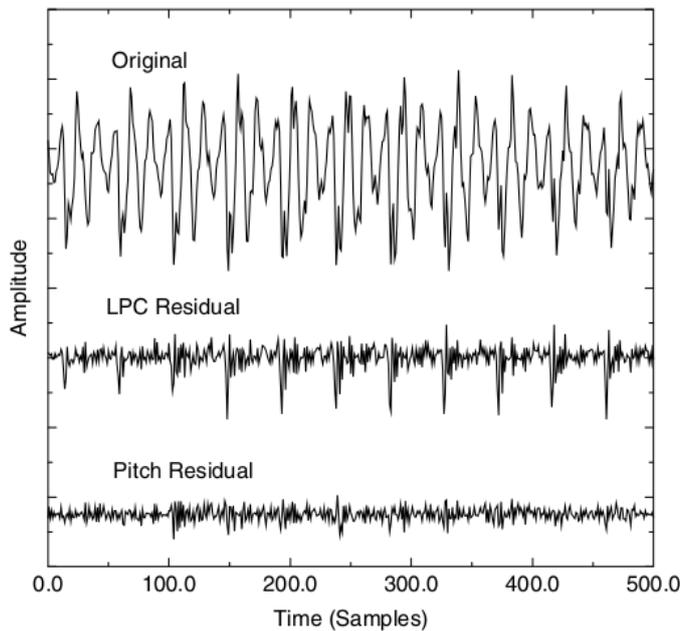
$$e'[n] = e[n] - \hat{e}[n] = e[n] - a_D e[n - D]$$

filtro de LTP:

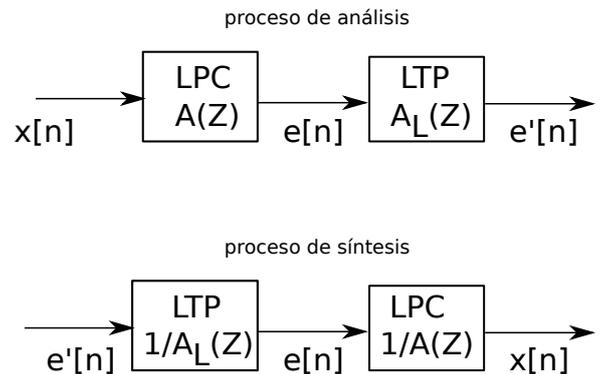
$$A_L(z) = \frac{E'(z)}{E(z)} = 1 - a_D z^{-D}$$

- métodos clásicos para estimar  $D$ , e.g. autocorrelación, AMDF
- permite eliminar periodicidad en el residuo en el caso de sonidos sonoros y se blanquea el espectro en el caso de sonidos sordos

# Predicción de largo plazo (LTP)

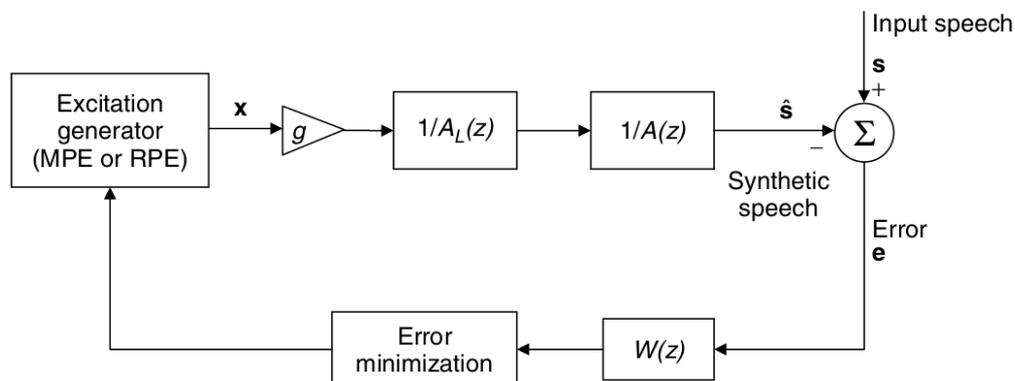


[Kondoz, 2004]



- a los parámetros de LPC
  - ▶ ganancia  $G$ , coeficientes  $a_k, 1 \leq k \leq p$
- se agregan los de LTP
  - ▶ ganancia  $a_D$ , retardo  $D$

## LPC en lazo cerrado (analysis-by-synthesis)



esquema típico de análisis por síntesis [Spanias et al., 2007]

- se determinan parámetros del modelo, luego se optimiza la excitación
- módulos de generación de excitación típicos
  - ▶ regular pulse excitation (RPE) (pulsos equiespaciados y ubicación inicial)
  - ▶ multi-pulse excitation (MPE) (pulsos de diferente posición y amplitud)
  - ▶ code excited linear prediction (CELP) (excitación se elige de un codebook)
- ponderación *perceptual* del error con filtro,  $W(z)$

# Filtro de ponderación perceptual

- modifica el error tal que el ruido de cuantización sea enmascarado por las formantes de alta energía

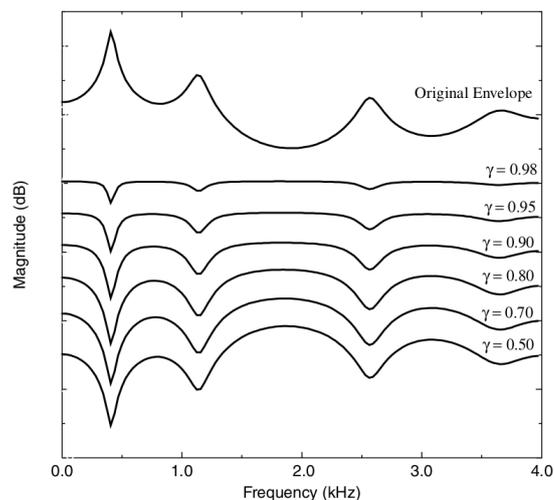
$$W(z) = \frac{A(z)}{A(z/\gamma)}$$

$$= \frac{1 - \sum_{k=1}^p \alpha_k z^{-k}}{1 - \sum_{k=1}^p \alpha_k \gamma^k z^{-k}},$$

con  $0 \leq \gamma \leq 1$

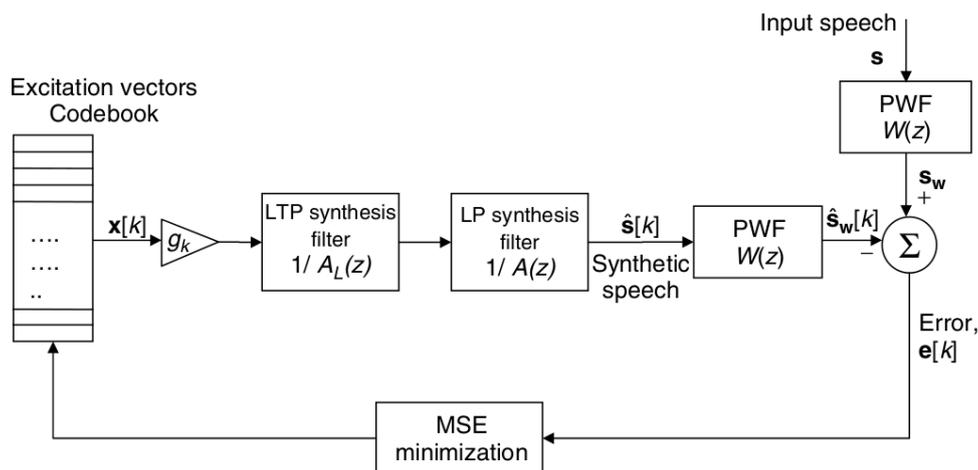
- $\gamma$  no altera frecuencia de formantes pero si su ancho de banda por,

$$\Delta f = -\frac{f_s}{\pi} \ln \gamma$$



[Kondoz, 2004]

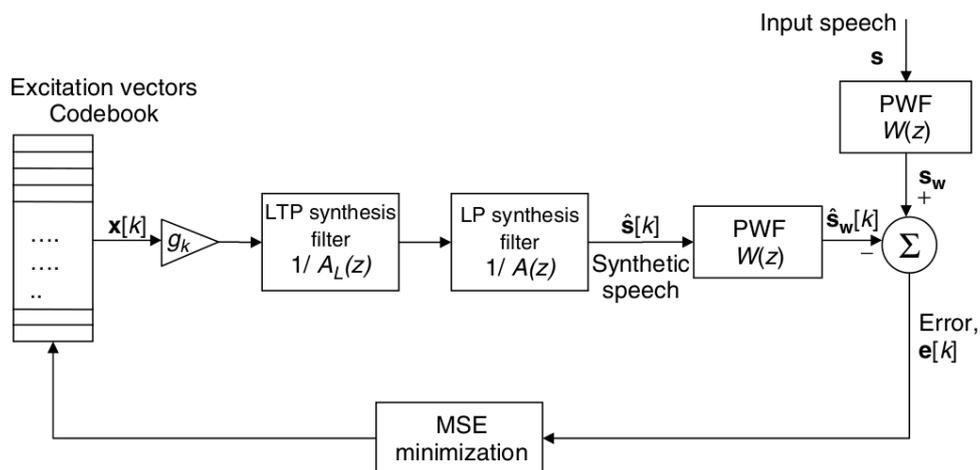
# Code Excited Linear Prediction (CELP)



[Spanias et al., 2007]

- se codifica el índice de la excitación óptima en el codebook
- se construye con muestras de ruido, sintético o entrenado
- cantidad de vectores del codebook grande no se gana al entrenar

# Code Excited Linear Prediction (CELP)



[Spanias et al., 2007]

$$e[k] = \bar{s}_w - g_k \hat{s}_w[k] \quad \bar{s}_w = s_w - s_w^0 \quad \text{con } s_w^0 \text{ estado inicial de } W(z)$$

$$\min_k \epsilon_k = e[k]^T e[k], \quad g_k = \frac{\bar{s}_w^T \hat{s}_w[k]}{\hat{s}_w^T[k] \hat{s}_w[k]}, \quad \epsilon_k = \bar{s}_w^T \bar{s}_w - \frac{(\bar{s}_w^T \hat{s}_w[k])^2}{\hat{s}_w^T[k] \hat{s}_w[k]}$$

$k$ : índice de excitación, se mueve hasta óptimo y luego se calcula  $g_k$

## Análisis tiempo-frecuencia

- permite elaborar representación más compacta que la temporal
- fenómenos psicoacústicos se caracterizan principalmente en frecuencia
- se debe manejar el compromiso de resolución tiempo-frecuencia

tipos principales:

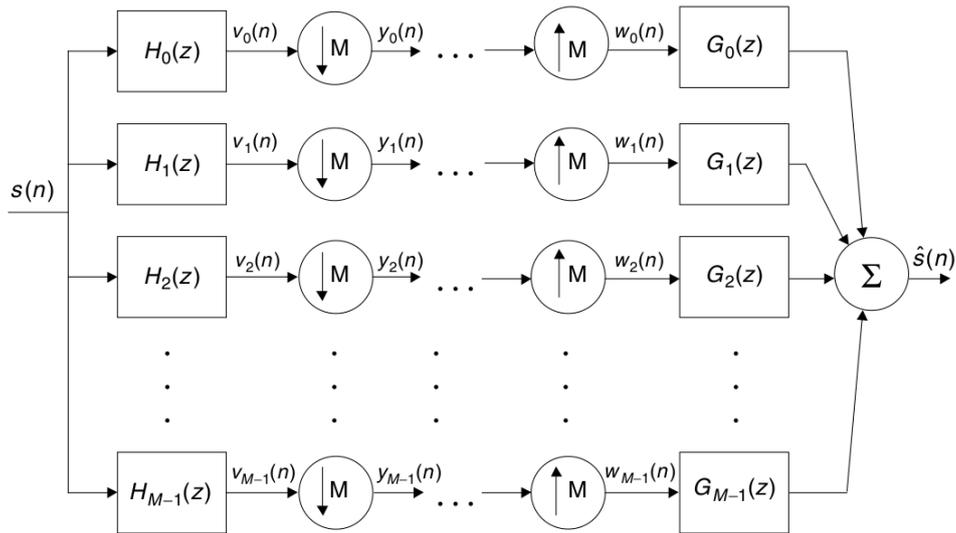
### banco de filtros

- ▶ descomposición en señales de ancho de banda reducido
- ▶ decimación de las señales de cada banda
- ▶ segmentación en bloques en las bandas

### transformadas tiempo-frecuencia

- ▶ segmentación en bloques y cálculo de espectro
- ▶ short-time fourier transform (STFT)
- ▶ modified Discrete Cosine Transform (MDCT)

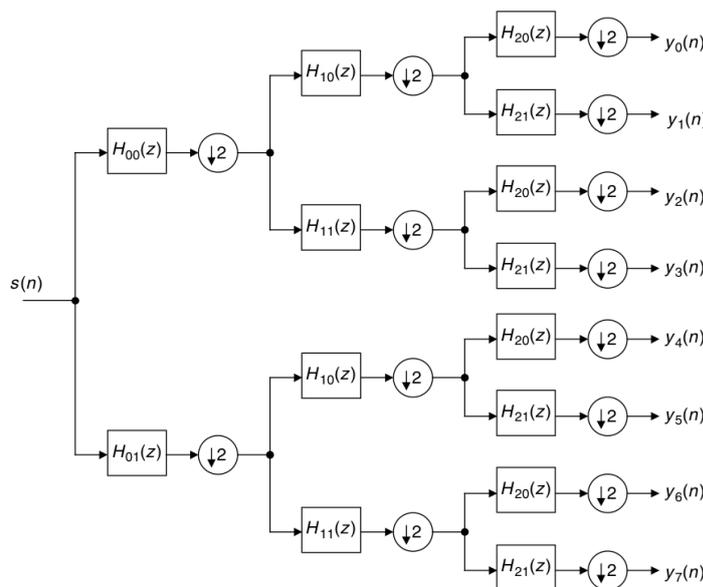
# Banco de filtros



[Spanias et al., 2007]

- filtrado y muestreo crítico de cada banda (máxima decimación)
- cuantización y codificación en sub-bandas, usando modelo perceptual
- filtros de orden finito y muestreo crítico implican aliasing inevitable

# Implementación del banco de filtros



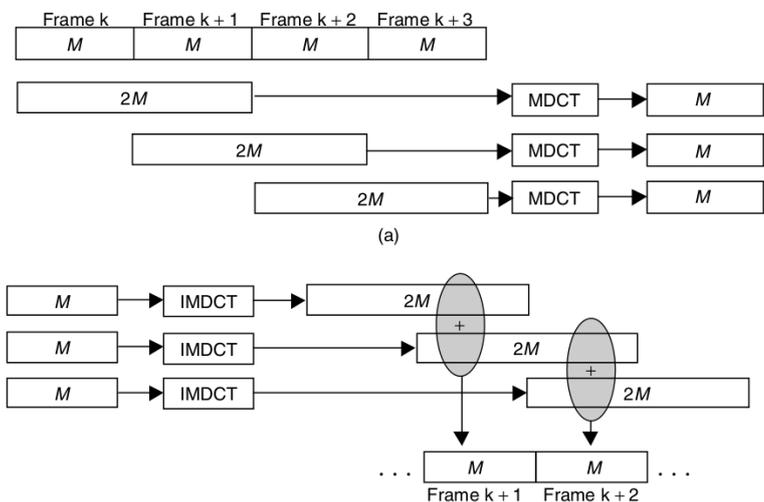
[Spanias et al., 2007]

- filtros en cuadratura: espejados (QMF) o conjugados (CQF)
- buscan respuesta global pasa-todo y fase lineal

# Modified Discrete Cosine Transform

$$X_k = \sum_{n=0}^{2M-1} x[n] \cos \left[ \frac{\pi}{M} \left( n + \frac{1}{2} + \frac{M}{2} \right) \left( k + \frac{1}{2} \right) \right] \quad \mathbb{R}^{2M} \rightarrow \mathbb{R}^M$$

- ventanas de  $2M$  muestras dan bloques de  $M$  bins
- se cancela el aliasing temporal y se obtiene reconstrucción perfecta mediante el solapamiento de bloques consecutivos
- usado en diversos codecs (e.g. MP3, ogg vorbis)



[Spanias et al., 2007]

## Referencias I

-  Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience, New York, NY, USA.
-  Kondo, A. (2004). *Digital Speech: Coding for Low Bit Rate Communication Systems*. John Wiley & Sons.
-  Spanias, A., Painter, T., and Atti, V. (2007). *Audio signal processing and coding*. Wiley-Interscience, 1st ed. edition.