ETSI Telecomunicación

Aplicaciones del Tratamiento Digital de la Señal

Apuntes de Pak

Nombre	Compact.	Real/Compl.	Complejidad	Fija/var.
KLT	óptima	real	alta	var.
Haar	baja	real	baja	fija
WHDT	baja	real	baja	fija
DFT	alta	compleja	media	fija
DCT	alta	real	media	fija

Interesa:

- Baja complejidad (muy importante en vídeo)
- Compactación de la energía
- Decorrelación de los coeficientes (para codificación independiente)

→ Transformada Karhunen-Loève

$\overline{x}_c = \overline{\overline{A}} \cdot \overline{x}$

A = autovectores en filas
B = autovectores en columnas

$x[n] \rightarrow \overline{\overline{R}}_x \rightarrow \begin{cases} a_k = \text{autovector } k\text{-ésimo} = b_k \\ |X[k]|^2 \text{ (pot)} = \text{autovalor } k\text{-ésimo} \end{cases}$

decorrelación entre coef

diagonal = potencia

↳ Maximiza la concentración de energía en pocos coef
↳ Decorrelación de los coeficientes
→ Es adaptativa, depende de R_x → alta complejidad

Matriz de autocorrelación

$\overline{\overline{R}}_x = E\{\overline{x} \cdot \overline{x}^T\}$ $\quad \overline{x} \equiv$ V.A.'s

$$\overline{\overline{R}}_x = \begin{bmatrix} R[0] & R[1] & \cdots & R[N-1] \\ R[1] & R[0] & R[1] & \\ \vdots & R[1] & R[0] & \\ & & & R[1] \\ R[N-1] & & & R[0] \end{bmatrix}$$

$R[0] = \sigma_x^2$ la pot

si x es AR-1
↳ $\overline{\overline{R}}_x = \sigma_x^2 \cdot \rho^{|m|}$

Autovectores/valores
$|R - \lambda I| = 0 \rightarrow \lambda_i$
$R \cdot v_i = \lambda_i \cdot v_i \rightarrow v_i$
(1 grado libertad)

→ Transformada Haar concreta

- MUY baja complejidad (solo sumas y restas)
- a mayor orden del coef k, de menor nº de muestras temporales depende

ej: $A_8 =$

Transformada concreta del coseno DCT

$b_k[n] = \sqrt{\frac{2}{N}}\, \alpha[k] \cos\left(\frac{\pi k}{N}\left(n + \frac{1}{2}\right)\right)$

funciones base

garantiza la ortonormalidad $\alpha[k] = \begin{cases} \sqrt{\frac{1}{2}} & k = 0 \\ 1 & \text{resto} \end{cases}$

óptima si $x[n]$ es AR-1 con $\rho \rightarrow 1$ (buen modelo para las imágenes)

- Guarda una relación con la FFT que permite un cálculo eficiente

Transformadas 2D

las funciones son MATRICES NxN

Transformada 2D ortonormal de una imagen NxN:

$$\begin{cases} X[k,l] = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x[n,m]\, b_{k,l}[n,m] \\ x[n,m] = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} X[k,l]\, b_{k,l}[n,m] \end{cases}$$

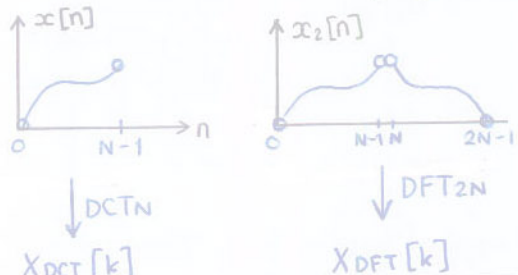
Transformadas 2D separables

$b_{k,l}[n,m] = b_k[n] \cdot b_l[m] \rightarrow \begin{cases} \overline{\overline{X}}_c = \overline{\overline{A}} \cdot \overline{\overline{X}} \cdot \overline{\overline{A}}^T \\ X = A^T \cdot X_c \cdot A \end{cases}$

Es como poder aplicar la transf 1D dos veces

Relación entre DFT y DCT

sea $x[n]$ $\quad x_2[n] = \begin{cases} x[n] & n \in [0, N-1] \\ x[2N-1-n] & n \in [N, 2N-1] \end{cases}$



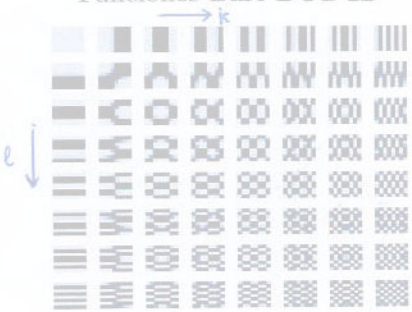
se tiene que $X_{DCT}[k] = X_{DFT}[k] \cdot \beta[k] \cdot e^{-j\frac{\pi k}{2N}}$

↳ $\begin{cases} \frac{1}{2\sqrt{N}} & k = 0 \\ \frac{1}{\sqrt{2N}} & k \neq 0 \end{cases}$

Recuerda que la DFT era la TF de la repetición periódica → saltos bruscos → altas frecuencias

En cambio ahora vemos que la DCT es la TF de la repetición periódica de $x_2[n]$ → sin saltos bruscos → más energía en bajas frecuencias

Funciones Base DCT-2D



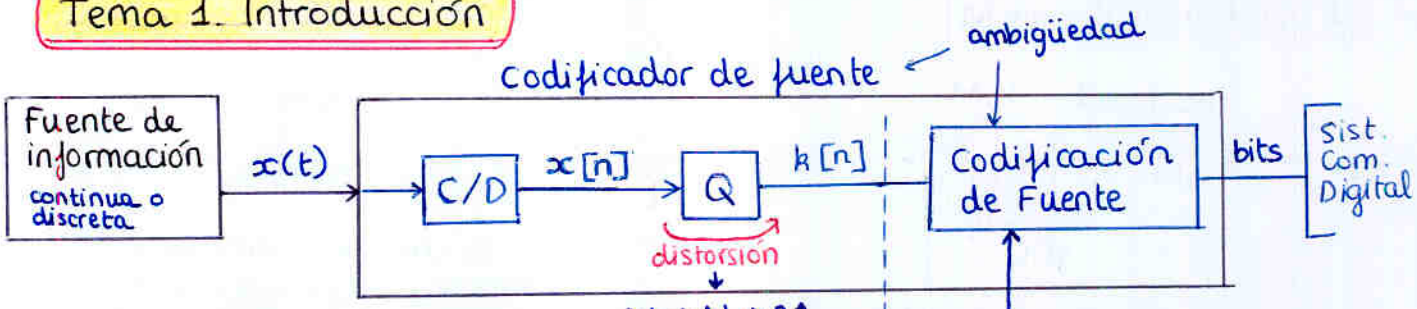
Aplicaciones del Tratamiento Digital de la Señal

Apuntes de Pak (Francisco José Rodríguez Fortuño)
ETSI Telecomunicación. Universidad Politécnica de Valencia.
Segundo cuatrimestre de 4º curso
Curso 2006/2007

Fecha de última actualización: 1 Agosto 2007

Aplicaciones del Tratamiento Digital de la Señal: Referencia rápida

Tema 1. Introducción



$D\downarrow \rightarrow \Delta\downarrow \rightarrow R\uparrow$
distorsión tasa

Existe cuantif. que minimiza la distorsión (para una R dada)

cuantif. no uniforme: intervalos pequeños allá donde $x[n]$ es más probable.

Sin distorsión

- Si logramos tasa igual a la entropía de la fuente, entonces la codificación es óptima (no buscar más)
- Para algunas fuentes, Huffmann (más bits para símbolos más probables) es óptimo.
- Huffman agrupando muchos símbolos ($\rightarrow \infty$) siempre tiende al óptimo, pero se complica mucho
- un método más simple que se acerca al óptimo es la codificación aritmética.
- Si aún así la tasa es muy elevada, recurrimos a codificación <u>con distorsión</u>

Podemos considerarlo como fuente discreta que proporciona símbolos de un alfabeto

Si no son independientes $I(S_1, S_2) < I(S_1) + I(S_2)$ hay que eliminar la redundancia

Teoría de la información de Shannon

Asumiendo símbolos independientes

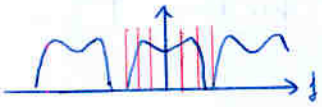
- información de un símbolo: $I(S_i) = \log_2\left(\frac{1}{P_i}\right)$ [bits]
- entropía de la fuente: $H = \sum_{i=1}^{M} P(S_i) \cdot I(S_i)$ $\left[\frac{bits}{simb}\right]$

Clasificación de codificación de fuente

- Codificación de forma de onda:

se codifica la señal en el tiempo
- PCM
- DPCM

- Codificación subbanda o por transformadas:



se codifica el espectro de la señal

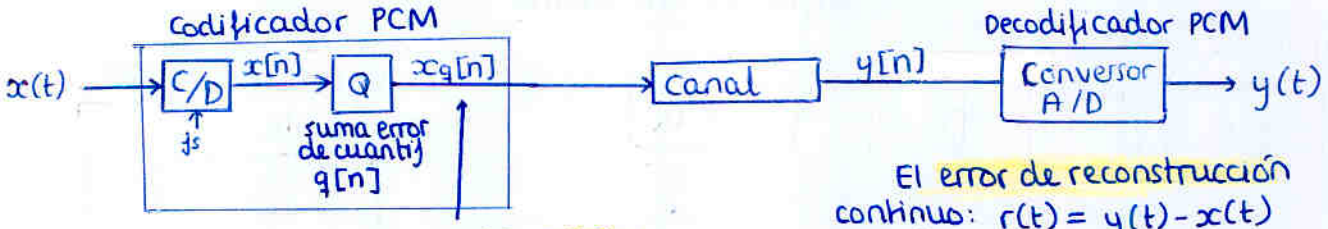
- Codificación basada en modelos:

supongo que mi señal $x[n]$ sigue un modelo matemático con pocos parámetros $\rightarrow$ transmito los parámetros

En la codificación con distorsión podemos jugar D vs R

Tema 2. Codificación Diferencial

1. Codificación PCM



$x_q[n] = x[n] + \underbrace{q[n]}_{\text{error de cuantif}}$

El error de reconstrucción
continuo: $r(t) = y(t) - x(t)$
discreto: $r[n] = y[n] - x[n]$

- En PCM el error de reconstrucción es igual al error de cuantificación $\quad r[n] = q[n] = x[n] - x_q[n]$
- Potencia del error $\quad \sigma_r^2 = \boxed{\sigma_q^2 = \underbrace{\varepsilon^2}_{\text{cte. que depende de } x \text{ y del cuantif}} \cdot 2^{-2\underbrace{R}_{\text{tasa bits/s}}} \cdot \underbrace{\sigma_x^2}_{\text{potencia de señal}}}$ → ¿Porqué? que yo repa $\sigma_q^2 = \frac{\Delta^2}{12} = \frac{(2x_m/2^R)^2}{12}$
- Relación señal a ruido $\quad \boxed{SNR_{PCM} = 10 \log \frac{\sigma_x^2}{\sigma_r^2} = \underbrace{6R}_{\text{cada } +1 \text{ bit/s aumenta SNR} + 6 \text{ dB}} - 10 \log \varepsilon^2}$ (dB)

2. Codificación DPCM

Ya que la pot. del error de cuantif es proporcional a la pot de la señal de entrada, interesaría transformar $x \rightarrow e$ (reversible) con $\sigma_e^2 < \sigma_x^2$ y cuantificar e.

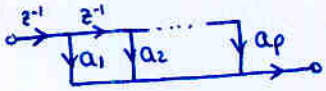
Predicción:

$e[n] = x[n] - \underbrace{x_p[n]}_{\text{predicción}}$ $\qquad$ P: orden del predictor

$x_p[n] = f(x[n-1], x[n-2], \dots, x[n-P])$ $\qquad$ diseñar f para minimizar σ_e^2 (difícil)

Predicción lineal: el predictor es un filtro FIR

$$\boxed{x_p[n] = a_1 x[n-1] + a_2 x[n-2] + \dots + a_P x[n-P]} = \sum_{m=1}^{P} a_m x[n-m]$$



Diseño del predictor: minimizar $\sigma_e^2 = E\{e^2[n]\}$

$\Downarrow$

Ecuaciones normales
o de Yule-Walker
o de Wiener-Hopf

$$\sum_{k=1}^{P} a_k R_x[|m-k|] = R_x[m], \quad m = 1, 2, \dots, P$$

ejemplo $P=2$

$$\left.\begin{cases} a_1 R_x[0] + a_2 R_x[1] = R_x[1] \\ a_1 R_x[1] + a_2 R_x[0] = R_x[2] \end{cases}\right\} \Rightarrow a_1, a_2 \text{ óptimos}$$

tal que $e[n] = x[n] - x_p[n]$ minimiza σ_e^2

en PCM cuantificamos $x[n]$

$$\sigma_q^2 = \varepsilon_{PCM}^2 \cdot 2^{-2R} \cdot \sigma_x^2 \longrightarrow SNR_{PCM} = \frac{\sigma_x^2}{\sigma_r^2}$$

en DPCM cuantificamos $e[n]$

$$\sigma_q^2 = \varepsilon_{DPCM}^2 \cdot 2^{-2R} \cdot \sigma_e^2 \longrightarrow SNR_{DPCM} = \frac{\sigma_x^2}{\sigma_r^2}$$

en ambos casos $\sigma_r^2 = \sigma_q^2$ (bucle cerrado)

(en abierto sería $\sigma_r^2 = \sigma_q^2 \cdot \Sigma |h[n]|^2$)

ganancia de predicción:

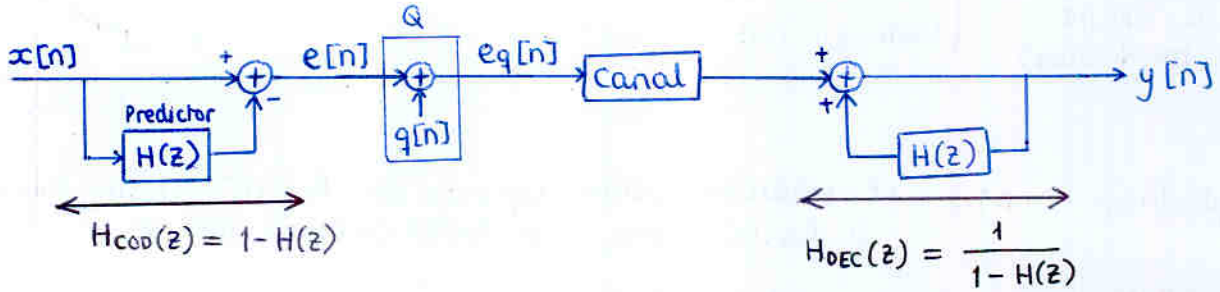
$$g_p = \frac{SNR_{DPCM}}{SNR_{PCM}} = \frac{\sigma_x^2}{\sigma_e^2}$$

$\sigma_e^2 \rightarrow$ justo lo que hemos minimizado

$$SNR_{DPCM}(dB) = SNR_{PCM}(dB) + G_p(dB)$$

- Esquema de bucle abierto

· lo primero que se nos ocurre
· no es bueno ya que $\sigma_r^2 \underset{>}{\neq} \sigma_q^2$



Si no existen errores de canal:

$$y[n] = e_q[n] * h_{DEC}[n]$$
$$= (e[n] + q[n]) * h_{DEC}[n]$$
$$= x[n] + \underbrace{q[n] * h_{DEC}[n]}_{r[n]}$$

$e * h_{DEC} = x$

se ve amplificado por el decodificador

$r[n]$ error de reconstrucción

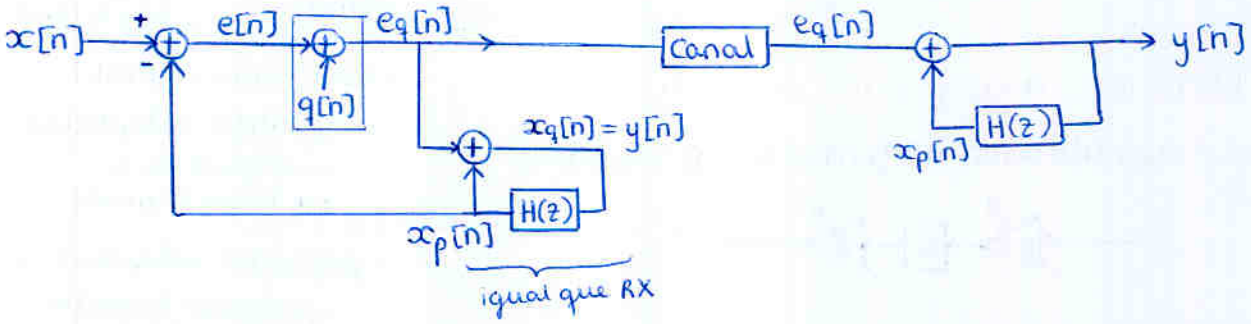
pot: $\sigma_r^2 = \sigma_q^2 \cdot \sum_n |h_{DEC}[n]|^2 > \sigma_q^2$

· En bucle abierto el error de reconstrucción es mayor al de cuantificación.

· Interesaría que el predictor en el receptor dispusiera de las mismas muestras que el transmisor

- Esquema en bucle cerrado → DPCM

· el transmisor 'imita' lo que hará el receptor, para tener las mismas muestras



$$y[n] = x_q[n] = x_p[n] + e_q[n]$$
$$= x_p[n] + e[n] + q[n]$$
$$= x_p[n] + x[n] - x_p[n] + q[n]$$
$$= x[n] + \underbrace{q[n]}_{r[n]}$$

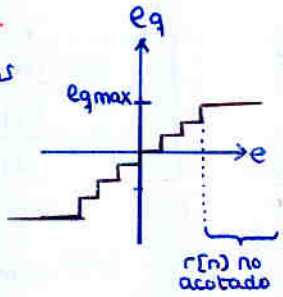
El error de reconstrucción sí es igual al de cuantificación:

pot de error:

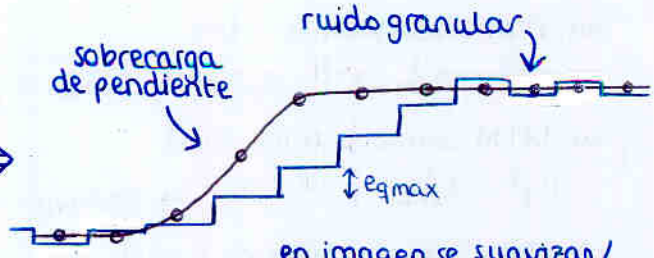
$\sigma_r^2 = \sigma_e^2$

Distorsión en DPCM

Al cuantificar $e[n]$ lo limitamos a un valor máximo e_{qmax} a partir del cual el error de reconstrucción/cuantificación no está acotado



⇒

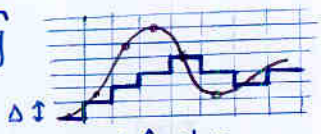


en imagen se suavizan/emborronan los contornos

Modulación Delta (DM)

Caso particular muy simple de DPCM (0 productos, 1 resta)

$$\begin{cases} \text{Predictor: } P=1,\ a_1=1 \rightarrow e[n] = x[n] - \overbrace{x[n-1]}^{x_p[n]} \\ \text{Cuantif: } 1\text{bit} \rightarrow e_q[n] = \begin{cases} +\Delta/2 \\ -\Delta/2 \end{cases} \end{cases}$$



distorsión ↑ salvo si $f_s \gg 2W$

Adaptative DPCM

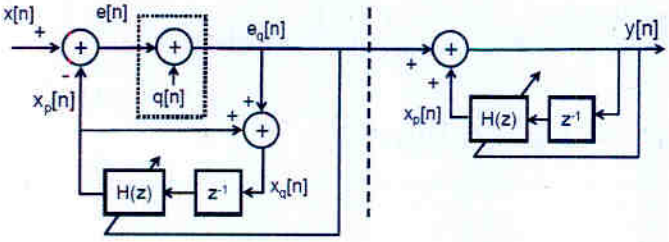
- El predictor óptimo depende de $R_x[m]$ (sistema Yule-Walker)
- Si $R_x[m]$ varía, el predictor debería variar

→ DPCM con predicción adaptativa

Usamos un F.A. con algoritmo Leaky LMS

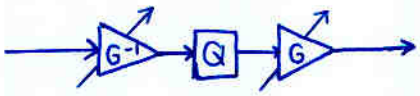
$$\underline{W}[n+1] = \alpha\, \underline{W}[n] + 2\mu\, e_q[n] \cdot \underline{X}[n]$$

↓ minimiza el efecto de la deriva si el canal introduce errores, ya que un error en el canal, además de afectar a las ∞ siguientes (por ser DPCM) cambiaría los coeficientes; con éste α el cambio se va diluyendo con el tiempo

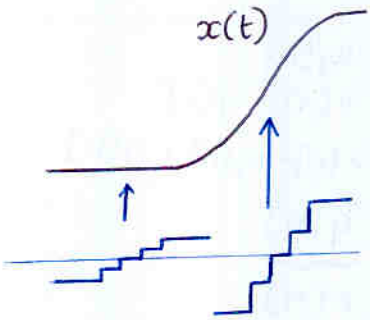


→ DPCM con cuantificación adaptativa

- cuantificación uniforme con Δ variable

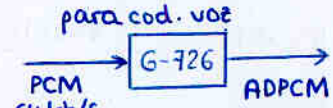


- Δ pequeña para reducir el ruido granular
- Δ grande para reducir la sobrecarga de pendiente



Estandar ITU G-726

para cod. voz



PCM 64 kb/s ley A o μ → ADPCM 16, 24, 32 y 40 kb/s

- Codificación ADPCM
  - cuantific. adaptativa
    - cómputo de G
    - 4, 7, 15 o 31 niveles
  - predicción adaptativa
    - predicción basada en x_q y e_q (no lo hemos visto (2 entradas al predictor))
    - adaptación de a_k y b_k por algoritmo del gradiente

3. Mejora en la conversión A/D → Conversores Sigma-Delta

- Es muy difícil construir cuantificadores de muchos niveles (2^n escalones exactos)
- Además debe ser analógico

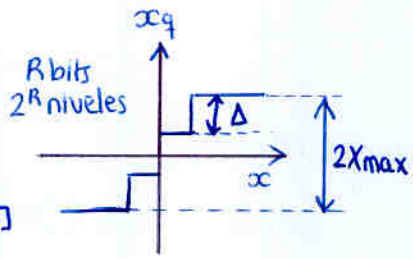
→ solución: cuantificador de pocos niveles (ej: 2) + tratamiento de señal

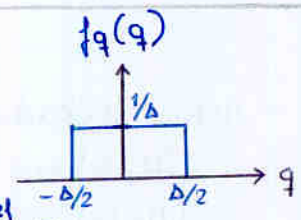
Conversor $\Sigma\Delta$; se basa en dos técnicas
- Sobremuestreo (oversampling)
- conformación de ruido (noise shaping)

Recordemos: Cuantificación en PCM

q[n]

si R↑↑ ⇒
- estacionario
- blanco
- incorrelado con x[n]
- fdp unif. en $-\frac{\Delta}{2}, \frac{\Delta}{2}$





$$\sigma_q^2 \overset{def}{=} \int_{-\infty}^{\infty} (q-\bar{q})^2 \cdot f_q(q)\, dq = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \frac{q^2}{\Delta} dq = \frac{\Delta^2}{12}$$

$$\boxed{\sigma_q^2 = \frac{\Delta^2}{12}} = \frac{(2X_{max}/2^R)^2}{12}$$

ej: sinusoide $\sigma_x^2 = \frac{X_{max}^2}{2}$

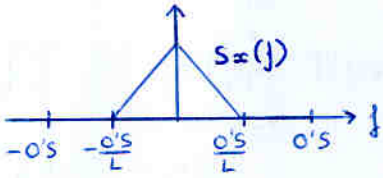
$$\boxed{SNR_Q = 10 \log \frac{\sigma_x^2}{\sigma_q^2} = 10\log\left(\frac{\sigma_x^2}{X_{max}^2}\right) + 4'77 + 6'02R}$$

$\boxed{+6dB \Leftrightarrow +1 \text{ bit}}$

3.1 Sobremuestreo

L: oversampling ratio

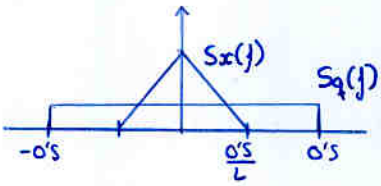
1. Sobremuestreamos a $f_s = L \times 2W$



d.e.p. [W/Hz]

$S_q(f) = \sigma_q^2 = \frac{\Delta^2}{12} \ \forall f$

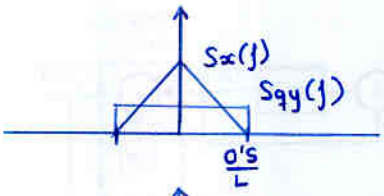
2. Cuantificamos
   - $S_q(f)$ blanco de pot $\frac{\Delta^2}{16}$



$$\sigma_q^2 = \int_{-0'5}^{0'5} S_q(f)\, df = S_q(f)$$

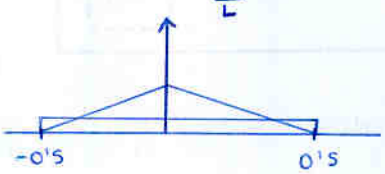
3. Filtramos paso bajo para eliminar ruido de cuantif.

$\sigma_{qy}^2 = \frac{\sigma_q^2}{L}$



$$\sigma_{qy}^2 = \int_{-\frac{0'5}{L}}^{\frac{0'5}{L}} \underbrace{\sigma_q^2}_{S_q(f)} df = \frac{\sigma_q^2}{L}$$

4. Reducimos f_s a la de Nyquist (diezmado por L)



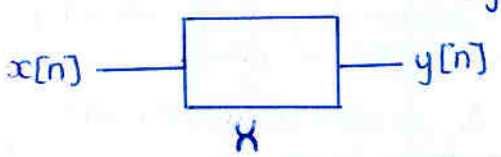
Obviamente al filtrar paso bajo, las muestras de salida tendrán más niveles (en teoría infinitos) y por tanto más bits, la ventaja es que esa nueva cuantificación de paso muy pequeño es digital

$$\Rightarrow \boxed{SNR = SNR_Q + \underbrace{10 \log (L)}_{3'01r}}$$

$L = 2^r$

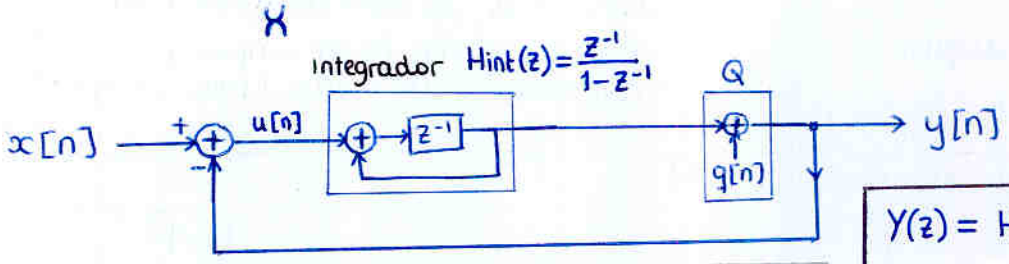
obviamente sólo sirve para señales W↓ ya que $f_s = L \cdot 2W$

3.2 Conformación de ruido


x[n]
y[n]
X

lograremos:

$$Y(z) = \overbrace{H_Q(z)}^{\text{filtro que actúa sobre el ruido}} \cdot Q(z) + \overbrace{H_x(z)}^{\text{filtro que actúa sobre la señal}} \cdot X(z)$$


Integrador $H_{int}(z) = \frac{z^{-1}}{1-z^{-1}}$
Q
x[n]
u[n]
z^{-1}
q[n]
y[n]

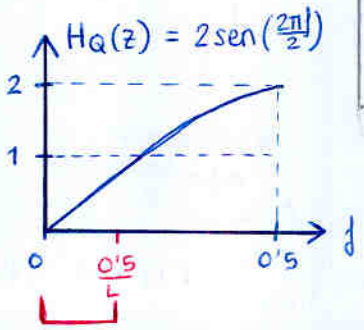
$$Y(z) = H_{int}(z) \cdot U(z) + Q(z)$$
$$= \frac{z^{-1}}{1-z^{-1}} \cdot (X(z) - Y(z)) + Q(z)$$
$$\Downarrow$$
$$Y(z) = \underbrace{z^{-1}}_{H_x(z)} X(z) + \underbrace{(1-z^{-1})}_{H_Q(z)} Q(z)$$

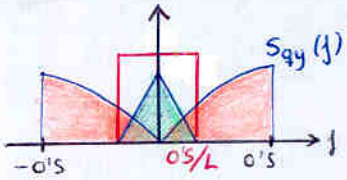
Hemos logrado:

$$\begin{cases} H_x(z) = z^{-1} & \text{(un simple retardo)} \\ H_Q(z) = 1 - z^{-1} \end{cases}$$

$$|H_Q(z)|^2 \underset{z=e^{j\omega}}{=} 4\,\text{sen}^2(\omega/2)$$

$|a|^2 = a \cdot a^*$



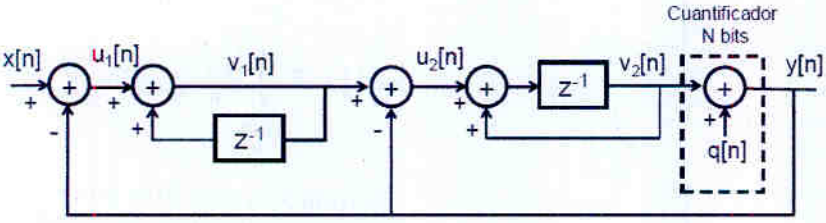


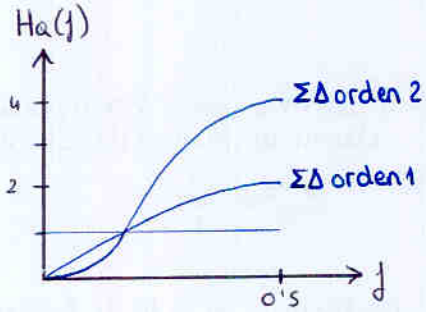
Modulación Sigma-Delta $\Longrightarrow$ combinándolo con sobremuestreo + F.P.Bj, nos quedamos sólo con la parte de poco ruido

$$\sigma_{qy}^2 = \int_{-\frac{0'5}{L}}^{\frac{0'5}{L}} S_{qy}(f)\,df = \int_{-\frac{0'5}{L}}^{\frac{0'5}{L}} |H_Q(f)|^2 \cdot \overset{\sigma_q^2}{S_q(f)}\,df = \frac{\pi^2}{3}\left(\frac{1}{L}\right)^3 \cdot \sigma_q^2$$

$$\boxed{SNR = SNR_q - 10\log\left(\frac{\pi^2}{3}\right) + \underbrace{30\log L}_{9'03r}}$$

Modulación ΣΔ de segundo orden:

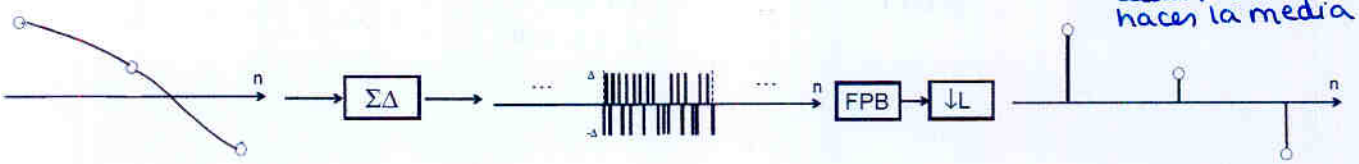




Es típico usar un cuantificador inicial de 1 bit (2 niveles) (muy sencillo de hacer) y luego utilizar L muy grande

$$\boxed{SNR = SNR_q - 10\log\left(\frac{\pi^4}{5}\right) + \underbrace{50\log L}_{15'05r}}$$

es como si para cada muestra hacer la media



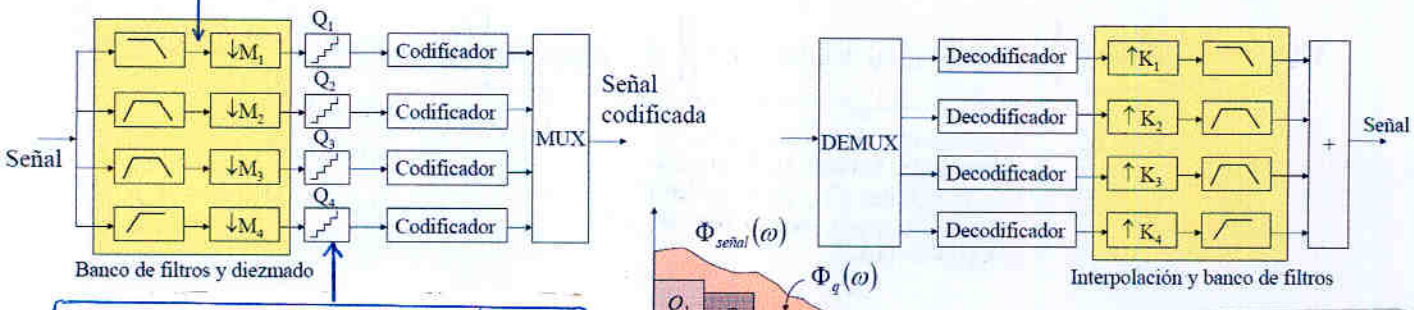
Tema 3. Codificación subbanda de Audio

Para que SNR_Q sea constante en todas las frecuencias

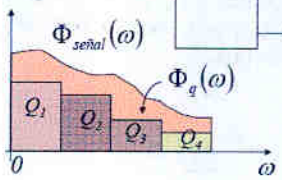
1. Ejemplo: Codificación subbanda por banco de filtros

señal sobremuestreada podemos diezmar

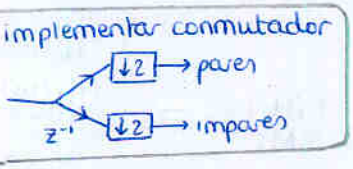
Diezmado crítico: $M_i = K_i = L$ (nº bandas)



En los cuantificadores ajustamos nivel máximo y nº de bits



implementar conmutador



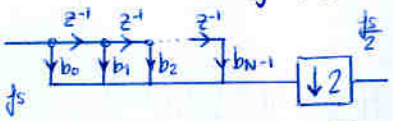
→ Implementación eficiente

a) Filtros o estructuras polifase

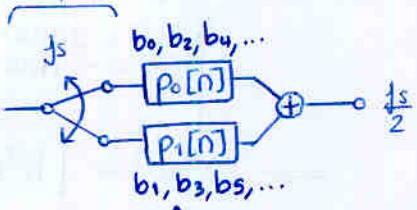
Filtro + Diezmador

elimina una de cada dos muestras

ej: M=2



al eliminar una muestra de cada 2 a la salida, las muestras pares sólo se multiplican por coefs impares

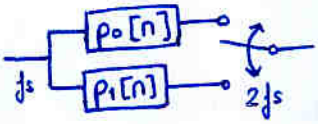


versión diezmada del filtro; si el original es plano, ambas versiones diezmadas también lo serán (diferencia sólo en fase)

Se reduce el número de operaciones en M

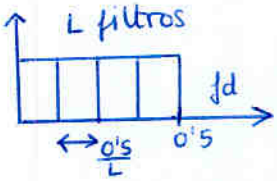
Interpolador = insertador de ceros + filtro

una de cada 2 muestras es cero



las muestras impares tenían multiplicación de los coef pares por cero, por tanto en muestras impares sólo importan coef impares

b) Realización en árbol

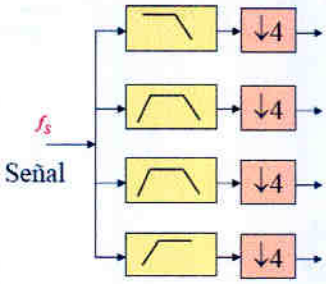


Un filtro requiere más coefs (i.e. más productos /muestra) cuanto menor sea su BW

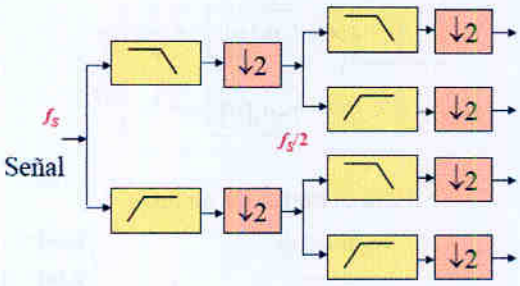
$$\text{long}(h_i) = \frac{Q}{0'5/L} = 2LQ$$

$\propto \frac{1}{BW}$ hipótesis

Q: factor calidad



$$\text{num op/seg} = 4 \cdot \underbrace{f_s}_{\frac{\text{muest}}{\text{seg}}} \cdot \underbrace{2Q4}_{\frac{\text{op}}{\text{muestra}}}$$

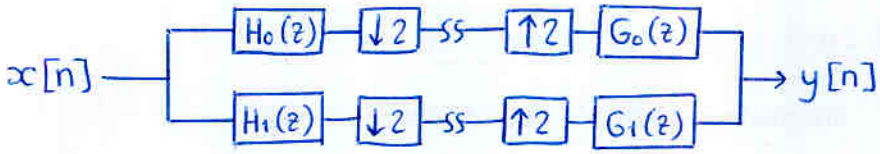


$$\text{num op/seg} = 2 \cdot f_s \cdot 2 \cdot Q \cdot 2 + 4 \cdot \frac{f_s}{2} \cdot 2 \cdot Q \cdot 4 = 2 \cdot f_s \cdot 2 \cdot (2 \cdot Q \cdot 2)$$

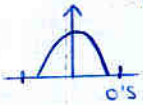
Reducción del número de operaciones por : $\frac{L^2}{4 \log_2 L}$

c) Filtros QMF (Cuadrature Mirror Frequency)

Los estudiamos para 2 subbandas (paso alto y paso bajo) pero en realidad valen para más (modulando un mismo filtro con coseno de distintas frecs)



$$Y(\omega) = X(\omega)\cdot\underbrace{\left[\tfrac{1}{2}\left(H_0(\omega)\cdot G_0(\omega) + H_1(\omega)\cdot G_1(\omega)\right)\right]} + X(\omega-\pi)\left[\tfrac{1}{2}\left(H_0(\omega-\pi)G_0(\omega) + H_1(\omega-\pi)G_1(\omega)\right)\right]$$



podemos diseñar para que esto tenga fase lineal y módulo = 1, pero los QMF sacrifican eso a cambio de eficiencia

→ aliasing! se anula con (por ej:)

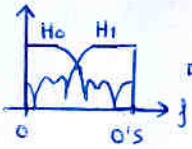
$$\boxed{\begin{aligned} G_0(\omega) &= 2H_1(\omega-\pi) \\ G_1(\omega) &= -2H_0(\omega-\pi) \end{aligned}}$$

Filtros QMF ⟹

$$\boxed{\begin{aligned} H_0(\omega) &= H(\omega) \\ H_1(\omega) &= H(\omega-\pi) \end{aligned}}$$

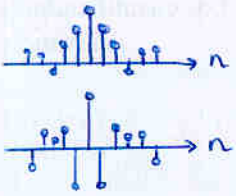
↓y para no aliasing

$G_0(\omega) = 2H(\omega)$
$G_1(\omega) = -2H(\omega-\pi)$



⟹

$$\boxed{\begin{aligned} h_0[n] &= h[n] \\ h_1[n] &= \underbrace{(-1)^n}_{e^{-i\pi n}} h[n] \end{aligned}}$$

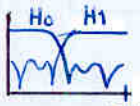


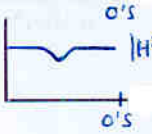
$$\Longrightarrow \boxed{Y(\omega) = \left[H^2(\omega) - H^2(\omega-\pi)\right]X(\omega)}$$

no es posible tener fase lineal y módulo 1

- fase lineal y $|H^2(\omega)-H^2(\omega-\pi)| \simeq 1$
- fase ≃ lineal y $|H^2(\omega)-H^2(\omega-\pi)| \equiv 1$
- compromiso entre ambos

ejemplo:



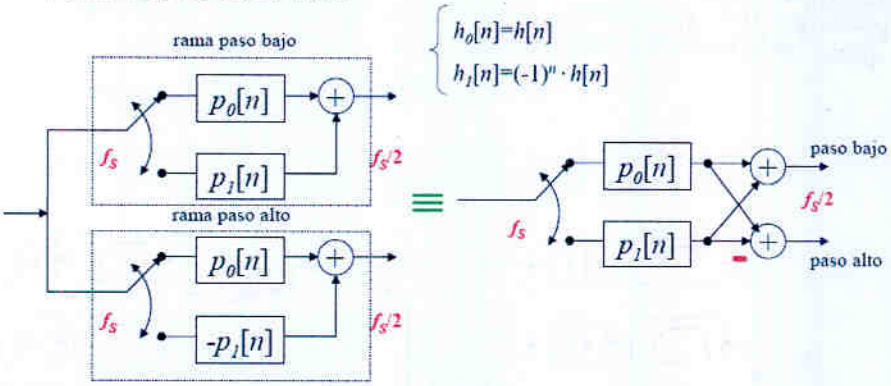


no reconstruye exactamente

a cambio logramos:

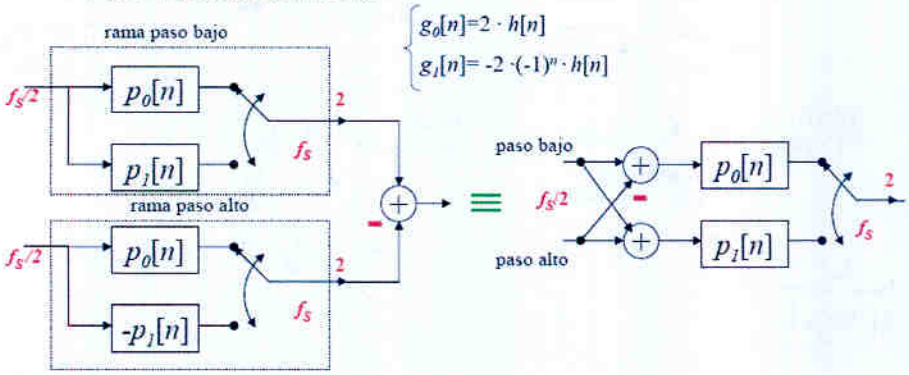
- Implementación eficiente de filtros QMF (usando además polifase)

- Para el banco de análisis:



$$\begin{cases} h_0[n]=h[n] \\ h_1[n]=(-1)^n\cdot h[n] \end{cases}$$

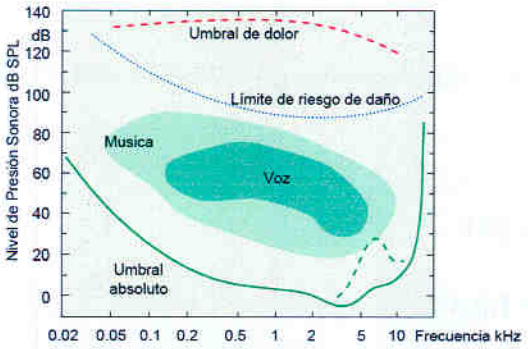
- Para el banco de síntesis:



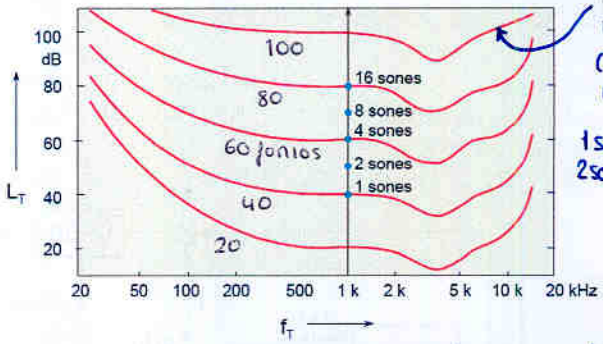
$$\begin{cases} g_0[n]=2\cdot h[n] \\ g_1[n]=-2\cdot(-1)^n\cdot h[n] \end{cases}$$

2. Percepción del sonido

$$dB\ SPL = 20 \log \left(\frac{P[Pa]}{20 \mu Pa} \right)$$



Sonoridad:



curvas isofónicas (se percibe mismo volumen

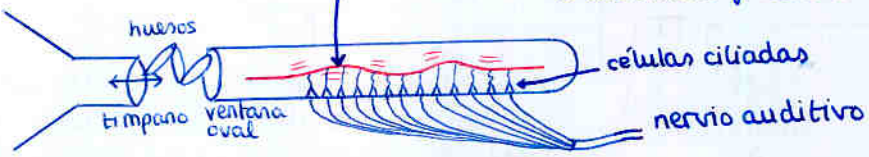
1 son = 40 dB SPL a 1kHz
2 son = se percibe el doble

unidad: fon → nivel de presión sonora de un tono de 1kHz para que tenga el mismo volumen subjetivo

· Bandas críticas:

Transformación frecuencia-posición en la membrana basilar:

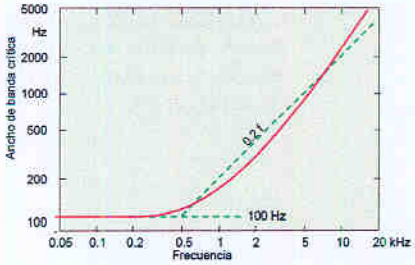
membrana basilar: distintas posiciones resuenan a distintas frecuencias



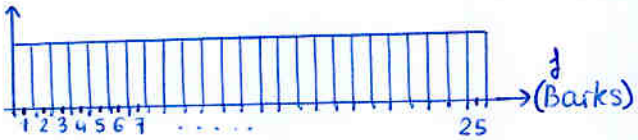
¡¡Es un banco de filtros!!

A cada "filtro" se le llama banda crítica.

El AB de cada banda crece con la frecuencia →



de Hz a Barks (fórmula)



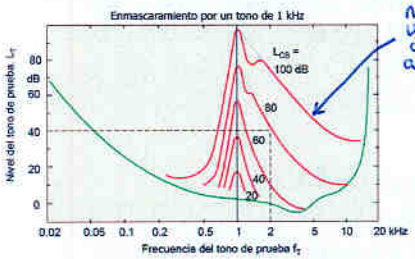
Las bandas críticas dan lugar a enmascaramiento (no distinguir 2 tonos muy juntos, entre otras cosas)

Transformando el eje de frecuencia de Hz a Barks, se tienen 25 bandas críticas centradas en Barks enteros

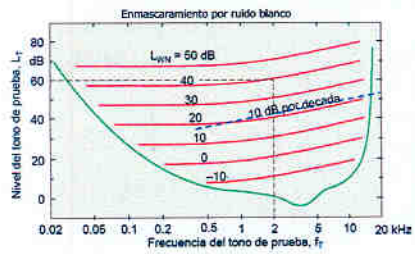
· Enmascaramiento

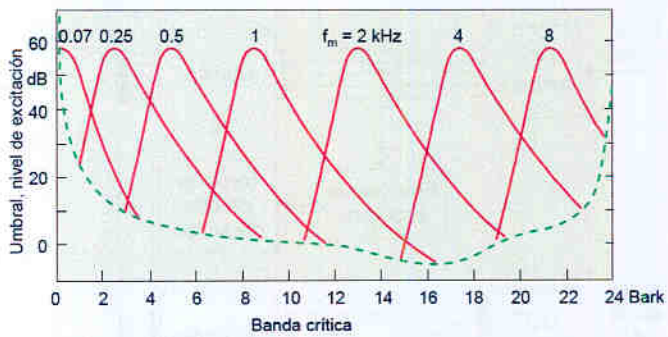
La 'anchura' en frecuencia del enmascaramiento de un tono crece con la frecuencia, pero es cte. en escala de Barks → modelo simple

- Enmascaramiento simultáneo: nuevo umbral de audición al introducir un tono de distintas potencias o ruido blanco

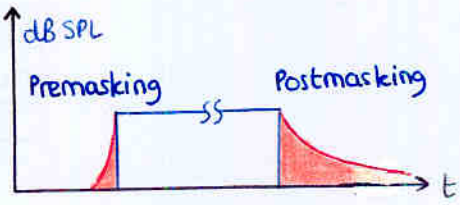


nuevo umbral de audición





- Enmascaramiento temporal

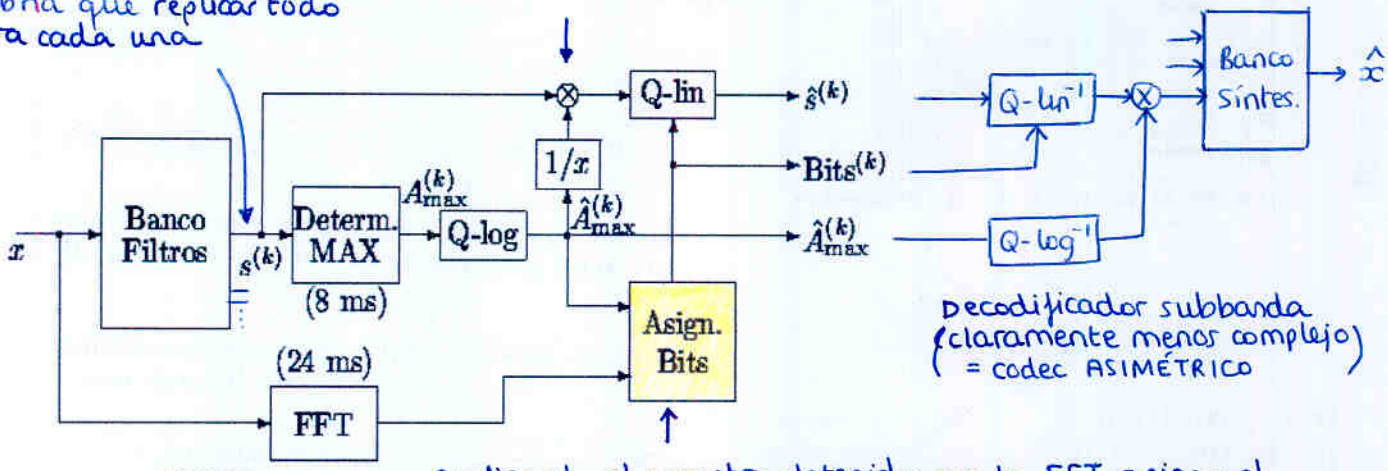


3. Tipos de codificadores

• Codificadores por subbanda

se representa el camino de UNA subbanda. En realidad habría que replicar todo para cada una

se normaliza por Amax (cuantificado, ya que será así como la tenga el receptor)



Decodificador subbanda (claramente menos complejo) = codec ASIMÉTRICO

supone estacionariedad durante esos 24 ms

Analizando el espectro obtenido con la FFT asigno el número de bits del cuantificador que me asegure ruido de cuantificación por debajo del umbral de audición

modelo psicoacústico: no se define en el estándar, se deja a elección del fabricante

MNR: mask to noise ratio

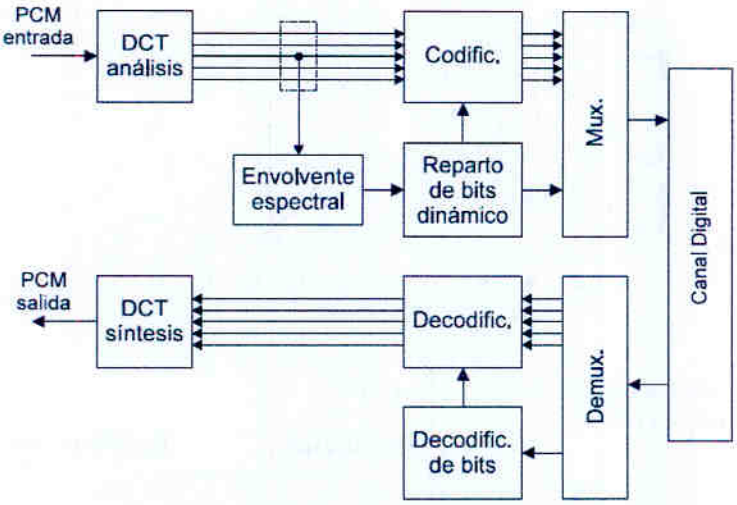
Hay que asegurar MNR ≥ 0 dB en toda la subbanda; se ajustan los bits del cuantificador de cada subbanda para ello

ejemplos: codificadores subbanda

MPEG 1 capas I y II

- 32 subbandas
- modelo psicoacústico
- codec asimétrico

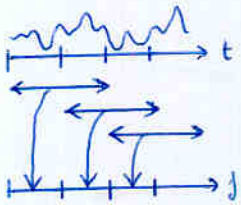
• Codificadores por transformada



- Se calcula una transformada (FFT, DCT, MDCT) y se transmiten los coeficientes (la cuantificación de los coeficientes con más o menos bits según la distribución espectral de energía y evaluando la percepción auditiva)

Transformada MDCT (modified discrete cosine transf.)

- solapa bloques:



bloques de 2N muestras

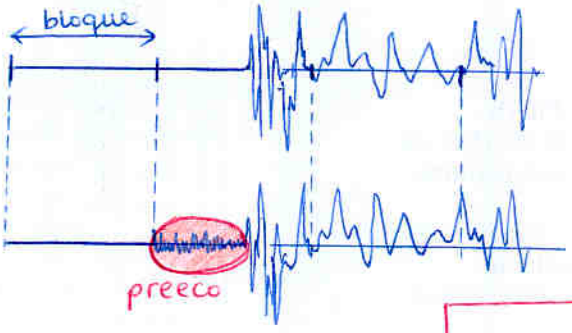
↓

N coef.

Problemas típicos de codificación por transformadas:

- efectos de bloque
- pre-ecos

- cada coeficiente afecta a todas las muestras del bloque
- si parte del bloque está en silencio → la cuantif de los coeficientes lo ensuciará



- Es muy molesto si no es suficientemente pequeño como para enmascararse por premasking
- Solución: bloques de tamaño variable

Bloque corto → más resolución temporal → preeco más corto
Bloque corto → menos coeficientes → menos resolución frecuencial

ejemplos: codificadores por transformadas

- Dolby AC-2
  - MDCT y MDST de 512 muestras cada 256 muestras (5'33 ms)

    256 coeficientes
    $f_s = 48\,kHz \rightarrow \frac{f_s}{2} = 24\,kHz$ → 93'75 Hz/coef

  - Para evitar preecos: bloques de 256 muestras cada 128 (2'67 ms)

    128 coeficientes → 187 Hz/coef → peor resolución espectral implica un peor análisis espectral y peor reparto dinámico de bits
- AT&T PAC

- Codificación híbrida
  - se divide en subbandas con banco de filtros
  - Luego se aplica la MDCT a cada banda

Ventajas:
- alta resolución espetral (hemos dividido 2 veces)
- mayor flexibilidad (diferentes resoluciones en frec. en cada banda)

ejemplos:

- Sony ATRAC
  - 3 subbandas
  - MDCT de 512 ptos (solapam. 50%)
    En caso de preecos:
    - Bandas altas 64 muestras
    - Bandas bajas 128 muestras
  - se usa en Minidisc
  - tasa: 256 kb/s estéreo

- MPEG1 capa III
  - subbandas
  - MDCT
    - 18 ptos caso normal
    - 6 ptos si hay preeco

4. MPEG

ej: 2 idiomas

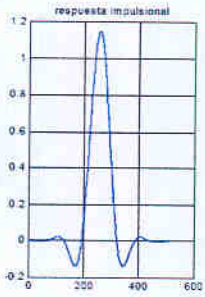
tiene en cuenta la redundancia entre los dos canales

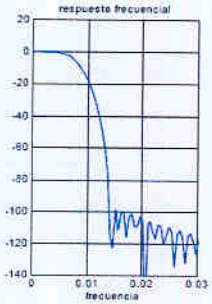
•Cuatro modos: Mono. Estéreo. Dual. Estéreo conjunto

Modelo en capas → Cada capa utiliza la anterior → Aumenta la complejidad y el retardo.
→Disminuye la tasa binaria.

CAPAS I y II

• Banco de filtros:
- **32** subbandas **iguales**: 24 kHz/32 = 750 Hz.
- **Diezmado crítico**: (diezmado por 32) fs= 1500 Hz.
- Filtros: 512 coeficientes con estructura polifase.
- Problemas:
  - Los anchos no coinciden con bandas críticas.
  - No son de reconstrucción perfecta.
  - Solapamiento entre bandas consecutivas.





- **Capa I:**
  - Agrupa muestras por bloques de 12 (8 ms).
  - Se calcula factor de escala (6 bits).
  - 32 subbandas x 12 muestras/bloque = 384 muestras (0-15 bits/muestra)
  - Demasiados bits para la escala, capa II lo soluciona sabiendo que los factores de escala tienen gran redundancia entre bloques cercanos

- **Capa II:**
  Agrupa los factores de escala de 3 bloques y transmite 1, 2, ó 3 factores de escala (según sean iguales o distintos)

- MOS (mean opinion score: 1 a 5) = 4.8 usando 128 kb/s con capa II

CAPA III (el famoso mp3 = MPEG 1 Layer III)

• Estructura híbrida
• Pretende una resolución en frecuencia alta
• Se aplica una **MDCT** (solapamiento 50%) a cada subbanda
  Caso normal: 18 puntos → 750 Hz / (18/2) = 41,67 Hz/coef de resolución
  Evitar preecos: 6 puntos

12 ms
4 ms
nº coefs tras aplicar MDCT a 18 muestras

• Cuantificación
  • No uniforme.
  • Codificación Huffman de la trama final.
  • **Análisis por síntesis** en la asignación de bits.
    El transmisor sintetiza el audio a partir de los coeficientes cuantificados para asegurarse de que se cumple la tasa y de que no se percibe distorsión → proceso iterativo
  • Tasa variable con reserva de bits.

MODELO PSICOACÚSTICO:

El estándar sólo define el decodificador y la estructura de datos, pero en la sección de información sugiere modelos:

• Modelo 1 (capas I y II):
  • FFT de 512 (I) ó1024 (II), ventana Hann.
  • Busca los enmascaradoresy calcula su umbral de enmascaramiento.
  • Evalúa frecuencia, nivel y si son tono o ruido.
  • El umbral total es la **suma de umbrales** incluyendo el umbral de audición.

• Modelo 2 (todas las capas, mucho más complejo)
  • FFT de 1024, ventana Hann.
  • Evaluación de la tonalidad mediante un predictorde primer orden.
  • Se incluyen propiedades oído interno •También se evalúan los pre-ecos.

- **MPEG 2:** (expande a MPEG1 para 5 canales, compatible con MPEG1 para recibir estéreo)
  - Nuevas fs= 16 / 22,05 / 24 kHz
  - Multicanal (5 canales)
  - AAC (abril 1997): estéreo, multicanal.
- **MPEG 4** (estándar de vídeo que incluye audio)
  - Multimedia, voz, acceso móvil
  - Voz: paramétrico, análisis por síntesis

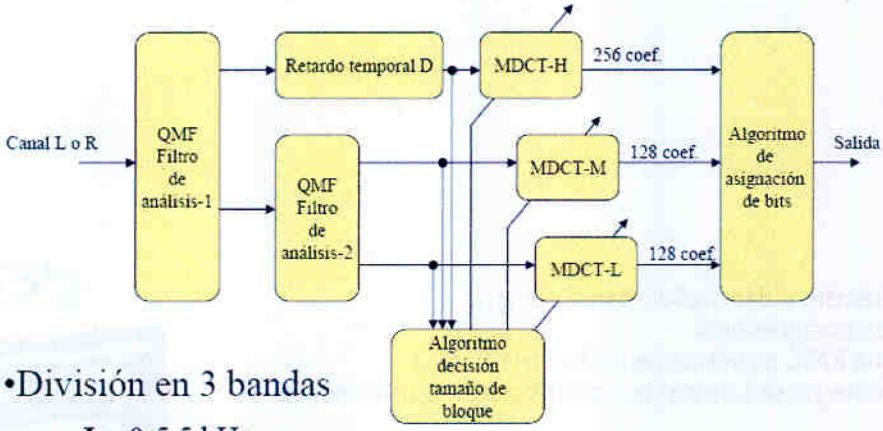
5. Minidisc

- Diámetro 64 mm, menor que un CD
- Capacidad: 1/5 de los datos de un CD.
- Compresión: 5:1 (74 min de audio calidad CD)

•Para ello se emplea una tecnología de compresión llamada ATRAC

-Se parte de la señal de audio estereo cuantificada con 16 bits y muestreada a 44.1-kHz.

-El sistema ATRAC emplea codificación subbanda (**3 bandas no uniformes**) y modelos psicoacusticos.



•División en 3 bandas

- **L**: 0-5.5 kHz
- **M**: 5.5-11 kHz
- **H**: 11-22 kHz.

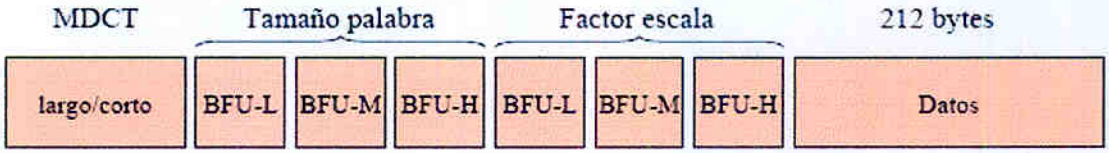
- Tamaño adaptativo de la transformada (2 modos):

  -Largo: compromiso resolución óptima
  - 512 muestras (11.6 ms) todas las bandas de frecuencia

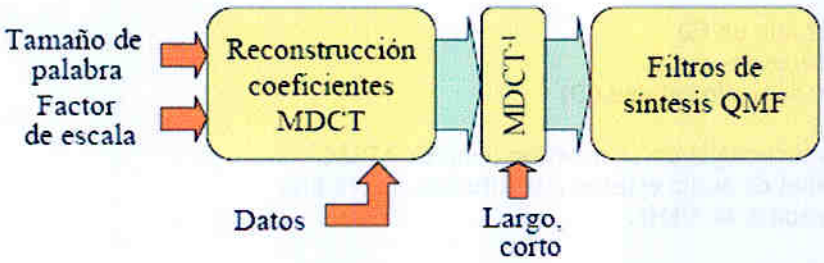
  -Corto: conmuta automaticamenteal modo corto para evitar los pre-ecos
  - 64 muestras (1.45 ms) bandas de alta frecuencia
  - 128 muestras (2.9 ms) bandas de media y baja frecuencia

- Los coeficientes MDCT se agrupan en **bloques (BFU)** y los valores espectrales se cuantifican con 2 parámetros.
  - Tamaño de palabra (resolución)
  - Factor de escala (margen dinámico)
  - Estos 2 parámetros son comunes para todo el bloque BFU (similitud psicoacústica).

- Para cada 512 muestras se codifica la siguiente información:



Decodificador:



- El decodificador es independiente del algoritmo de asignación de bits. Esto permite mejoras futuras.

6. DCC

- PASC (PrecisionAdaptiveSub-bandCoding).
- Factor de compresión 4.
- El sistema PASC es básicamente MPEG-1 Capa 1
- Actualmente no se fabrica: uso profesional exclusivamente.

Tema 4. Codificación de voz usando LPC

Propiedades de la voz

- **Ancho de banda**

  300 – 3400 Hz : voz telefónica ($f_s = 8$ kHz)
  0 – 5000 Hz : voz AM ($f_s = 11$ kHz)
  0 – 7000 Hz : voz banda ancha ($f_s = 16$ kHz)
  > 8000 Hz : calidad microfónica

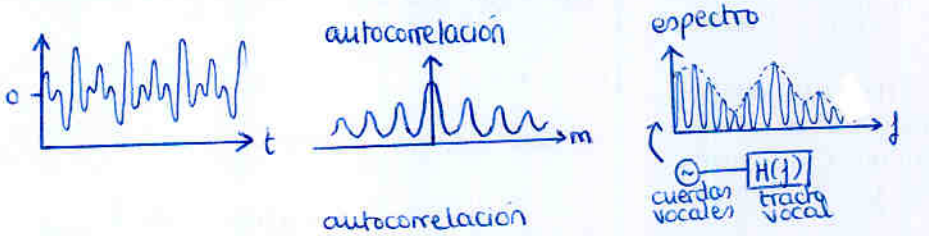
- **Rango dinámico**

  · Entre locutores : 60 dB (mucho menor a los 100 dB en audio)
  · Mismo locutor y tono : 20-30 dB

- **Tipos de sonidos** (tramos cortos 5-30 ms → cuasi-estacionaria)

Sonoros

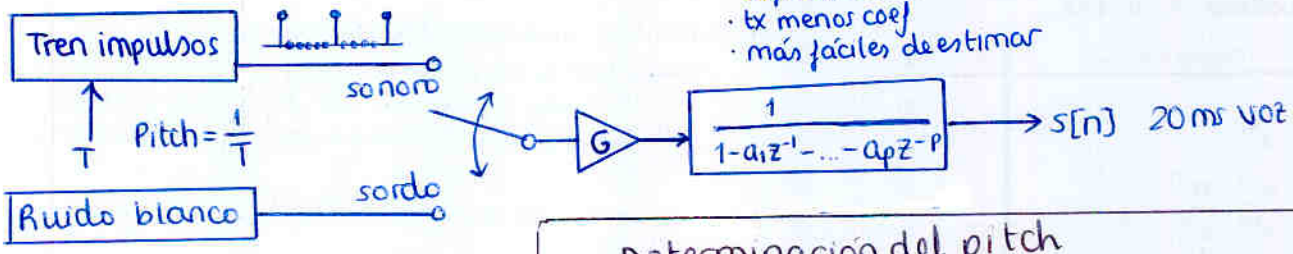
· cuerdas vocales
· estructura armónica
· periodicidades (P: pitch)



Sordos

· turbulencias
· no armónicos
· altas frecuencias

Modelo de producción de voz



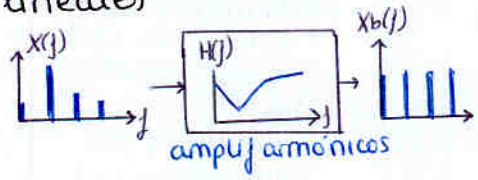
sistema todo polos
· suficientemente bueno
· tx menos coef
· más fáciles de estimar

polos = resonancias

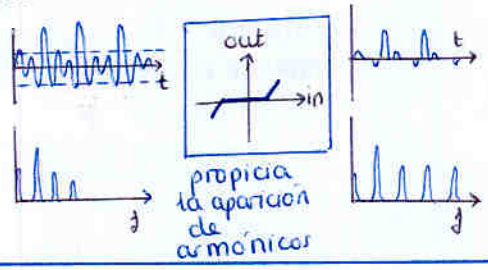
Parámetros necesarios : [sonoro/sordo; si sonoro : T; G; $a_1, a_2, \dots, a_p$]

Procedimientos de blanqueo:

- lineales

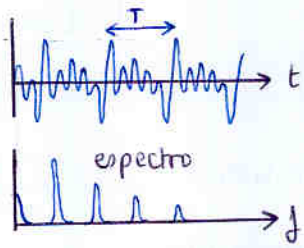


- no lineales

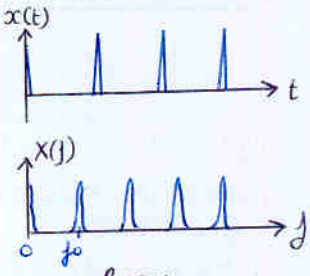


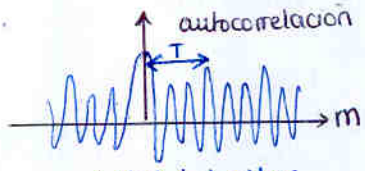
Determinación del pitch

SIN BLANQUEO

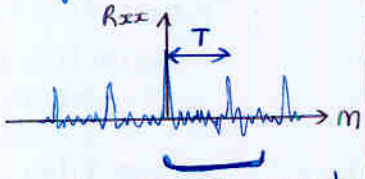


CON BLANQUEO





difícil de hallar el pitch



con la señal blanqueada es fácil buscar picos de la autocorrelación en un rango
100 Hz (hombres)
200 Hz (mujeres)

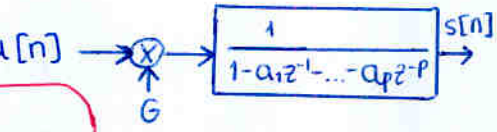
Cálculo parámetros del filtro LPC

Señal de excitación

hipótesis:

- $Pot(u[n]) = 1$
- $R_u[m] = 0$ para $m = 1, 2, \dots, P$

  válido tanto si $u[n]$ es un tren de pulsos como si es ruido blanco



⇒ Sistema de ecuaciones de Yule-Walker

$$\begin{bmatrix} R_s(0) & R_s(1) & R_s(2) & \cdots & \cdots & R_s(P-1) \\ R_s(1) & R_s(0) & R_s(1) & R_s(2) & \cdots & R_s(P-2) \\ R_s(2) & R_s(1) & R_s(0) & R_s(1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & R_s(0) & R_s(1) \\ R_s(P-1) & R_s(P-2) & \cdots & \cdots & R_s(1) & R_s(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{P-1} \\ a_P \end{bmatrix} = \begin{bmatrix} R_s(1) \\ R_s(2) \\ \vdots \\ \vdots \\ R_s(P) \end{bmatrix}$$

$$G^2 = R_s(0) - a_1 R_s(1) - \dots - a_P R_s(P)$$

Hay que resolver éste sistema cada 20 ms (160 muestras a $f_s = 8$ kHz)

↳ Algoritmo de Levinson-Durbin

- matriz toeplitz
- $O(P^2)$ en lugar de $O(P^3)$ cálculo eficiente

Una vez calculados los $R_s[m]$:

1. Hacer: $E_0 = R_s(0)$, $i=0$.
2. Incrementar i en 1.
3. Calcular: $k_i = \left(R_s(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R_s(i-j) \right) / E_{i-1}$
4. Hacer: $a_i^{(i)} = k_i$
5. Calcular: $a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}$ $\quad j = 1, \cdots, i-1$ (sólo para $i > 1$)
6. Calcular: $E_i = (1 - k_i^2) E_{i-1}$

i	Cálculos Parciales	k_i	E_i
0			○
1	$a_1^{(1)}$	○	○
2	$a_1^{(2)}$ $a_2^{(2)}$	○	○
3	$a_1^{(3)}$ $a_2^{(3)}$ $a_3^{(3)}$	○	○
⋮	⋮	⋮	⋮
P	$a_1^{(P)}$ $a_2^{(P)}$ ——— $a_P^{(P)}$	○	

Observaciones:

- Los coeficientes k_i que se obtienen son:

k_i : PARCOR coeficientes de correlación parcial

→ filtro estable si $|k_i| < 1$

→ permiten implementación en celosía:

→ una forma de realizar el filtro (como FDI, FDII, ...) en la cual aparecen los k_i y en la cual el error de redondeo es especialmente bajo

→ la cuantificación de los k_i tiene un efecto pequeño:

↳ el efecto es aún menor con los PARCOR modificados $g_i = \frac{1+k_i}{1-k_i}$

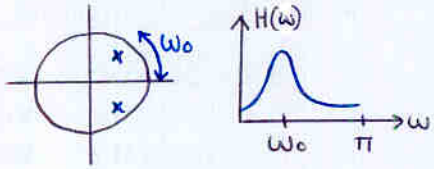
↳ y aún menor con LAR $= \log g_i$ (log area ratios)

Realmente,

hallar los coef $a_1, \dots, a_P$

≡ hallar los polos

≡ hallar las resonancias

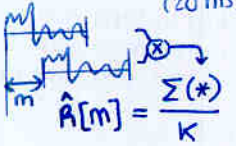


nº polos = nº resonancias × 2

↑ típico 6 a 10

Requiere estimar la autocorrelación

N — 160 muest. (20 ms)



$$\hat{R}[m] = \frac{\Sigma(*)}{K}$$

$K = \begin{cases} N : \text{sesgado} \\ N-m : \text{insesgado} \end{cases}$

Afortunadamente no hay que calcular $R[m]$ $m = 1 \dots N$ sino sólo hasta $m = 1 \dots P \ll N$

ejemplo: Vocoder LPC-10

- Voz a MUY baja tasa (uso militar) → si hay ruido fondo se confunden sonoros y sordos
- $f_s = 8$ kHz
- Tramas de 22'5 ms (180 muestras)

Parámetros excitación:

→ sonoro / sordo : 1 bit

→ G : 5 bits (log)

→ Pitch : 6 bits (log) ← si es sordo se rellena con protecc. errores

Parámetros filtro:

→ $\log g_1$, $\log g_2$, k_3, k_4 : 5 bits cada uno

→ $k_5, \dots, k_8$ (4 bits), k_9 (3 bits), k_{10} (2 bits) } sólo si sonoro

- 53 + 1 bits / trama → 2400 bits / seg
- Inteligible aunque artificial

¿Cómo tx los parámetros?

- Directamente a_i
  - → cuantificar afecta mucho
  - → puede resultar en inestab.
- Polos
  - → modulo y fase → garantizo estabilidad, pero el error aún se nota mucho
  - – implementación en cascada
- PARCOR : k_i
- PARCOR modificados : g_i
- LAR : $\log g_i$
  - → muy inmunes a errores de cuantificación

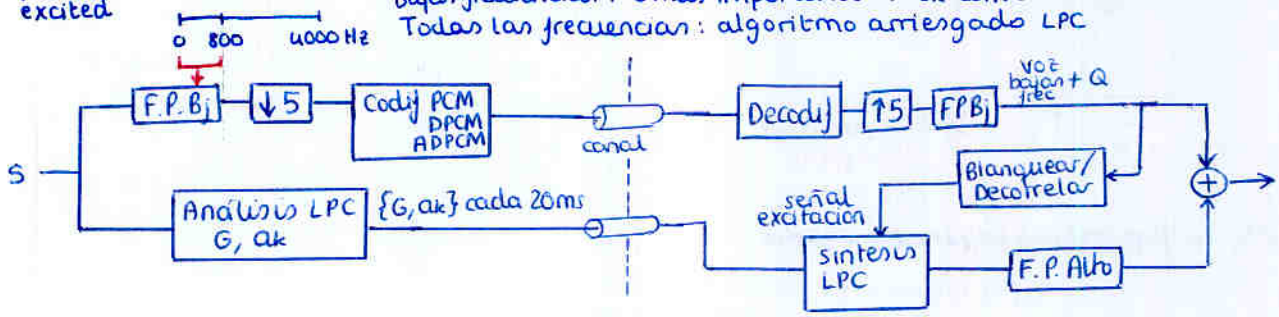
Otras técnicas LPC

Tratan de mejorar la deficiente calidad lograda con LPC

- **VELP**: canal auxiliar paso bajo (que además sirve como excitación mejor)

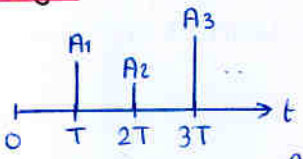
voice excited

Bajas frecuencias: lo más importante → tx convencional
Todas las frecuencias: algoritmo arriesgado LPC

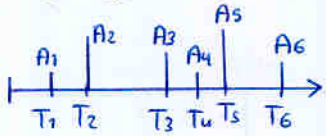


- Utilizar una señal de excitación mejor

→ **RPE** regular pulse excitation — tx las amplitudes de cada pulso junto a T



→ **MPLPC** multi-pulse LPC — Transmitimos amplitud Ai retardo Ti de cada pulso



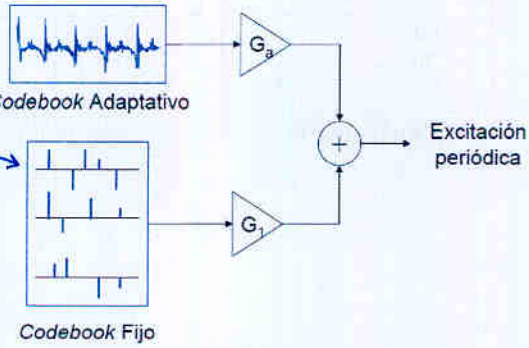
→ **CELP** Code excited LPC

Gran popularidad en los últimos años

biblioteca donde guardo segmentos de voz generados casi idénticos al original → Codebook Adaptativo

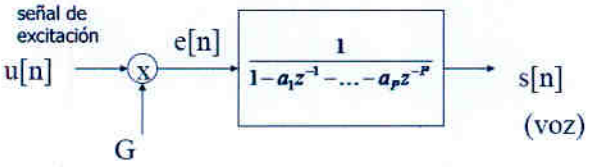
Biblioteca con M posibles señales de excitación. El tx escoge la que le proporcione mejor resultado y se la indica al rx con $n = \log_2 M$ bits

análisis por síntesis



→ LD-CELP
low delay (2ms)

- Transmitir la señal de excitación / error de predicción



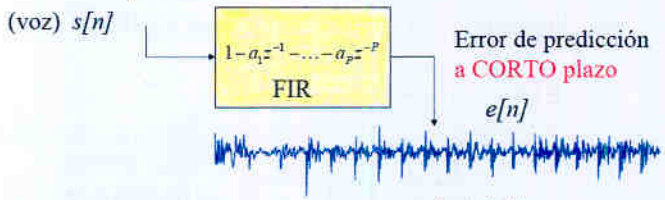
$$s(n)=\underbrace{a_1 s(n-1) + \dots + a_P s(n-P)}_{\text{predicción de la muestra}} + e(n)$$

la excitación tiene la forma/ está actuando como error de predicción

Análisis LPC ⟺ Análisis a corto plazo: $a_1, a_2, \dots, a_P$, G

- Error de predicción a corto plazo:

El filtro de análisis a corto plazo resulta ser el inverso del filtro predictor LPC



Transmitiendo e[n] y los coef del filtro podríamos recuperar exactamente la señal.

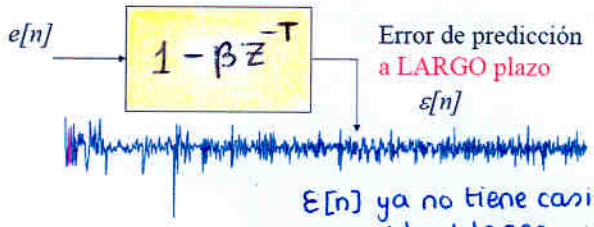
Tx e[n] (y no s[n]) tiene la ventaja de que e[n] tiene menos correlación.

Pero aparecen picos en e[n] cada cierto tiempo; eso indica una correlación, no con la muestra siguiente, sino a LARGO plazo

- Error de predicción a largo plazo

e[n] resulta ser cuasi-periódico en tramos sonoros

$$e[n] = \beta \cdot e[n-T] + \varepsilon[n]$$



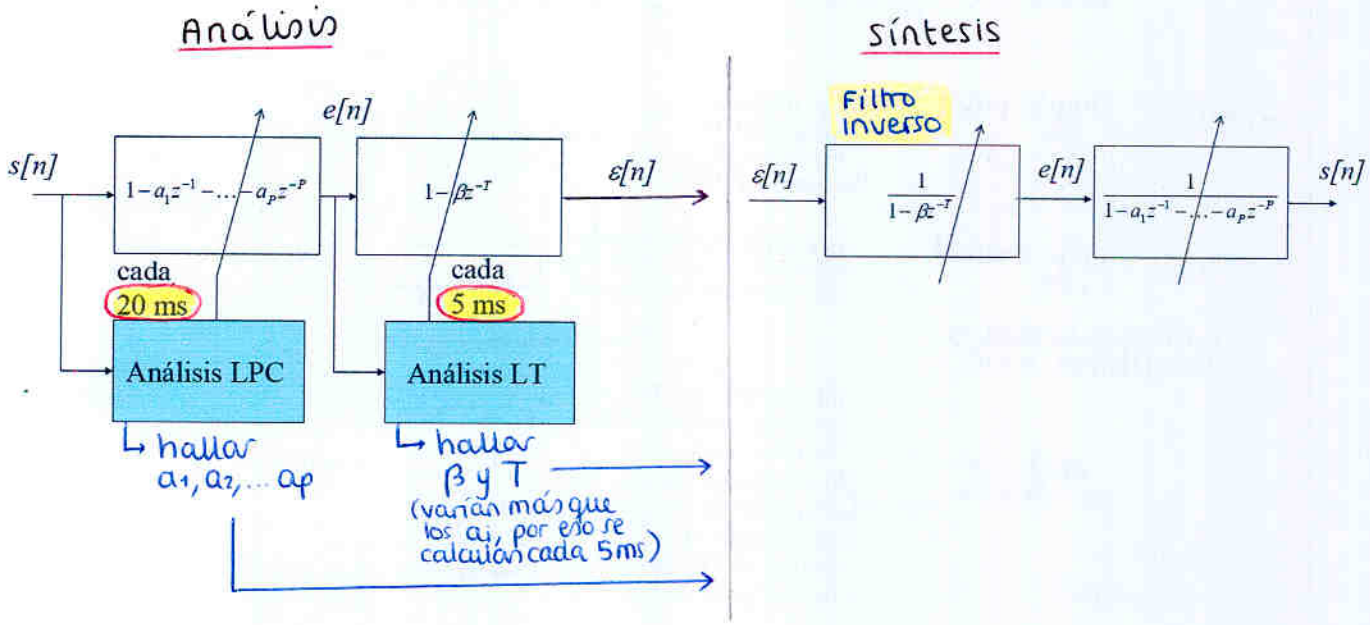
ε[n] ya no tiene casi correlación
~ ruido blanco → ideal para tx

Análisis LT (long Term)
- hallar T y β

T: buscando pico de Re[m] en cierta ventana $T = \arg(\max(Re[m]))$

$$\beta = \frac{Re[T]}{Re[0]}$$

- **LPC con predicción a largo plazo**

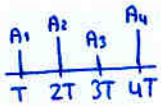


↳ hallar $a_1, a_2, \ldots a_p$

↳ hallar β y T (varían más que los a_i, por eso se calculan cada 5ms)

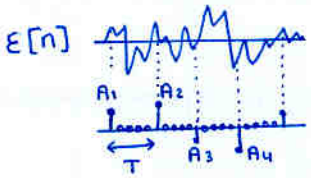
- **GSM : combina RPE con LTP**

RPE → Regular pulse excitation: Tx Ai de cada pulso



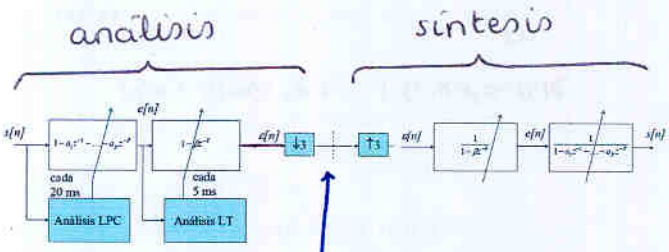
LTP → LPC con predicción a largo plazo (Long Term)

Lo que hace es que las Ai de los pulsos que transmite son justamente los valores de ε[n] en ese instante



→ ésta es la señal que tximos (más barata de txir que ε[n])

Si lo piensas, lo que en realidad se está haciendo es transmitir ε[n] diezmado



cuando diezmamos, ¿qué subtrama de las 3 posibles cogemos?

$n = 3k$
$n = 3k+1$
$n = 3k+2$

→ se coge la que tenga mayor energía (se indica cual es con 2 bits (posición retícula))

GSM : RPE-LTP 13 kbps

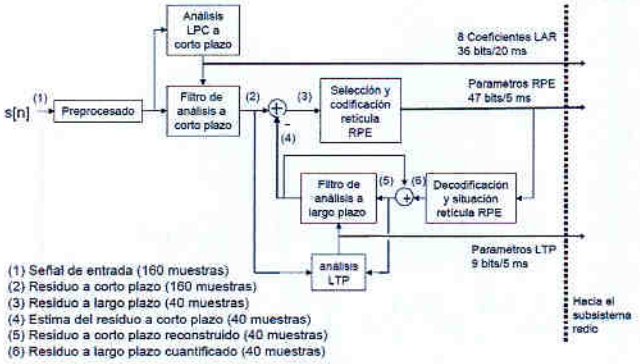
Codificación de los parámetros :

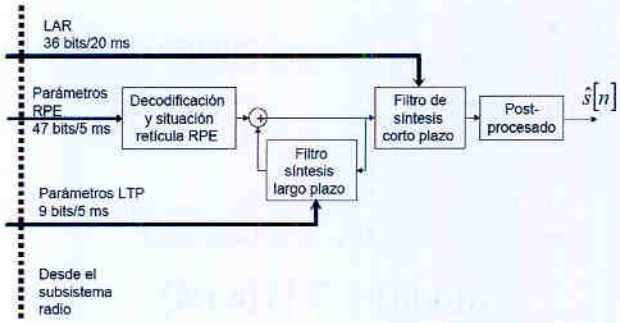
- **Filtro LPC** (orden 8)
  - coeficientes codif como LAR $\{\overset{n^o bits}{6/6/5/5/4/4/3/3}\}$ → 36 bits cada 160 muestras
- **Filtro LTP**
  - retardo T : 7 bits
  - ganancia β : 2 bits

  } → 9 bits cada 40 muestras
- **RPE** (amplitud de pulsos)
  - 40 muestras de $\varepsilon[n]$ diezmadas por 3 → 13 pulsos en APCM, 3 bits por pulso] 39 bits cada 40 muest.
  - amplitud bloque : 6 bits
  - posición retícula : 2 bits

  } → 47 bits cada 40 muestras

Trama de salida : 13 kbps → 260 bits/20 ms





Otros aspectos codec GSM

- Retardo 20 ms
- Detector actividad vocal (ruido confort)
- Protección contra errores selectiva (se dividen parámetros en 3 grupos) y cada uno se protege distinto

Tema 5. Codificación de imágenes

1. Señales y sistemas 2D

Sistema lineal:

$$T(\alpha s(x,y)+\beta v(x,y)) = \alpha T(s(x,y)) + \beta \cdot T(v(x,y))$$

sistema invariante en el espacio:

$$T\{s(x,y)\} = g(x,y) \rightarrow T\{s(x-x_0, y-y_0)\} = g(x-x_0, y-y_0)$$

Sistema LSI (lineal e invariante)

↳ respuesta al impulso

$$h(x,y) = T\{\delta(x,y)\}$$

↳ salida

$$g(x,y) = h(x,y) * s(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(x',y')\,h(x-x',y-y')\,dx'\,dy'$$

Transformada de Fourier 2D

$$S(\Omega_x,\Omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(x,y)\,e^{-j(\Omega_x x+\Omega_y y)}\,dx\,dy$$

→ convolución:

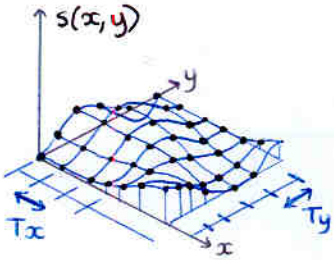
$$s(x,y) * h(x,y) \leftrightarrow S(\Omega_x,\Omega_y)\cdot H(\Omega_x,\Omega_y)$$

→ separabilidad

si $s(x,y) = a(x)\cdot b(y)$

$\Rightarrow S(\Omega_x,\Omega_y) = A(\Omega_x)\cdot B(\Omega_y)$

Muestreo ortogonal



$$s[n,m] = s(x,y)\big|_{x=nT_x,\ y=mT_y}$$

puede recuperarse si:

$$T_x \leq \frac{1}{2W_x} \qquad T_y \leq \frac{1}{2W_y}$$

→ Filtros 2D discretos LSI

$$h[n,m] = T\{\delta[n,m]\}$$

→ convolución discreta

$$g[n,m] = s[n,m] * h[n,m] = \sum_{n'=-\infty}^{\infty}\sum_{m'=-\infty}^{\infty} s[n',m']\cdot h[n-n',m-m']$$

→ Transformada de Fourier

$$S(e^{j\omega_x}, e^{j\omega_y}) = \sum_{n}^{\infty}\sum_{m}^{\infty} s[n,m]\,e^{-j(\omega_x m+\omega_y n)}$$

y aún se cumple que la convolución en espacio es el producto en ω

La señal de imagen

- RGB: $R(x,y)$, $G(x,y)$, $B(x,y)$ — las tres tienen igual importancia
- Y C_b C_r: $Y(x,y)$, $C_b(x,y)$, $C_r(x,y)$ → Y tiene más importancia
  - Puedo introducir más distorsión en C_b y en C_r
  - Tx con menor ancho de banda

2. Codificación PCM de imágenes

Dos discretizaciones:
- discretización en x, y → muestreo → pixeles
- cuantificación de los valores de cada una de las componentes → bits

Muestreo:

RGB: las 3 con la misma resolución

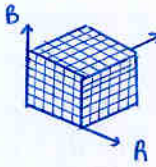
YCbCr: mayor frec de muestreo para Y

ejemplos
BMP (Microsoft) 2 tipos
- paleta 256 colores
- RGB 24 bit/pixel

TIFF (Adobe) 4 tipos
- TIFF-R: RGB-24 bit/pixel
- TIFF-P: paleta tamaño var.
- TIFF-G: grises 8 bit/pixel
- TIFF-B: binario 1 bit/pixel
- Opcionalmente: compresión entrópica (sin pérdidas)

Cuantificación

- Cuantificador óptimo: depende de fdp(R), fdp(G), fdp(B) varía para cada imagen
- Cuantificador uniforme:
  8 bits → 256 niveles
  24 bits/píxel



256^3 posibles colores

- Cuantificador vectorial
  VQ (paletas) (en desuso)
  - me fijo en los colores que predominan y empleo en ellos más tasas binarias
  - cada imagen necesita su 'paleta' (qué color representa cada palabra)
  - ya que la paleta ocupa memoria existe la paleta óptima

ej: imagen sin verde
R = 8 bits
G = 0 bits
B = 8 bits

→ **Necesaria codificación** para bajar la tasa (eliminar redundancia)

[con pérdidas / sin pérdidas]

[dominio espacial / dominio transformado]

a) Eliminar redundancia considerando pixeles aisladamente
unas amplitudes más probables que otras (codif. entrópica)

b) Eliminar redundancia entre pixeles

↳ **Codificación run-length**
1. Transformar símbolos en ráfagas de símbolos
   $S_0 S_0 S_0 \; S_1 S_1 \; S_2 S_2 S_2 S_2 S_2 \rightarrow \underbrace{0_3}\,\underbrace{1_2}\,\underbrace{2_5}$
   → long máxima de ráfaga para que las tablas de Huffmann no enormes
2. codificar las ráfagas (Huffman, cod aritmética, ...)

↳ codificación DPCM

Tasa R vs Distorsión D

¿cómo mido D?
- subjetivo: ej MOS
- objetivo:

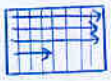
→ **MSE**: mean squared error entre antes y después

→ SNR: signal to noise ($S/_{MSE}$)

→ **PSNR**: SNR suponiendo pot máxima de señal

$$PSNR = 10 \log \frac{255^2}{MSE}$$

3. Codificación DPCM de imágenes



Raster scan: por filas de izquierda a derecha

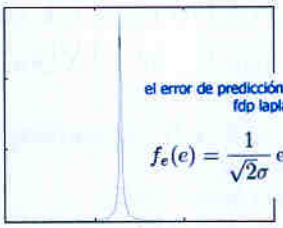
se aplica DPCM: $x[n,m] = \hat{x}[n,m] + e[n,m]$
predicción con píxeles anteriores

Imagen original

Histograma

Error de predicción

gris = 0, blanco = >0, negro = <0

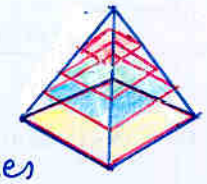


- si hay alta correlación entre píxeles vecinos ⟹ $\sigma_e^2 \ll \sigma_x^2$
- cuantificación suele ser uniforme, y al ser $\sigma_e^2 \downarrow \rightarrow \sigma_q^2 \downarrow$ (error cuantif.)

DPCM es muy sencillo pero logra poca compresión

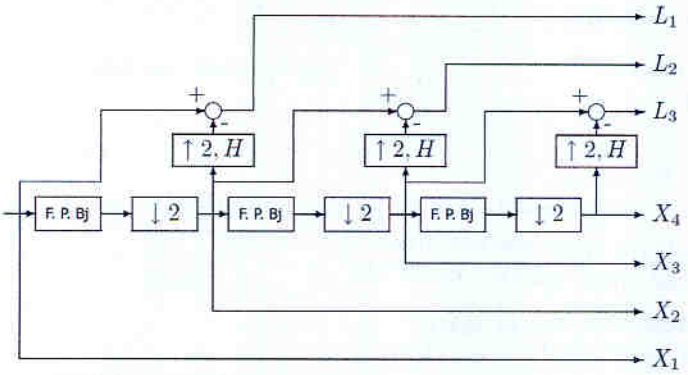
4. Codificación piramidal

Pirámide: Representación de la señal a distintas frecs de muestreo espaciales



- Pirámide gaussiana: los planos son las imágenes a distintas resoluciones
- Pirámide laplaciana: los planos son la diferencia entre una resolución y la inmediatamente inferior (tras subirle la frec muestreo para poder hacer la resta)

imágenes tienen fdp laplaciana



Pirámide Laplaciana

Pirámide Gaussiana

($L_4 = X_4$ ya que el último no tiene con quien restarse)

Es como banco de filtros

L_1 (paso alto) $M \times N$ pix
L_2 (paso banda) $\frac{M}{2} \times \frac{N}{2}$ pix
⋮
L_4 (paso bajo) $\frac{M}{2^{P-1}} \times \frac{N}{2^{P-1}}$ pix

- Hay que codificar cada uno de los P niveles
- Pixeles a codificar $> M \times N$
- Codificación más eficiente que DPCM

5. Codificación con transformadas

En el codificador ($x[n] \to X[k]$) $\xrightarrow[X[k] \to \hat{X}[k]\ \text{cuantif}]{TX}$ En el decodificador ($\hat{X}[k] \to \hat{x}[n]$)

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \underbrace{a_k[n]}_{\text{funciones base}}$$

equiv.

$$\overline{x_c} = \overline{\overline{A}} \cdot \overline{x}$$

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} a_0[0] & a_0[1] & \dots & a_0[N-1] \\ a_1[0] & a_1[1] & \dots & a_1[N-1] \\ \vdots & & & \\ a_{N-1}[0] & a_{N-1}[1] & \dots & a_{N-1}[N-1] \end{bmatrix} \cdot \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

las filas son las funciones base (una para cada coef k de la transf)

$$\hat{x}[n] = \sum_{k=0}^{N-1} \hat{X}[k] \cdot b_k^*[n]$$

$$\overline{\hat{x}} = \overline{\overline{B}} \cdot \overline{\hat{x}_c}$$

las columnas son las funciones base b_k (conjugadas si son complejas)

truco: cada función base 'define' un coef k

$$\boxed{\overline{\overline{B}} = \overline{\overline{A}}^{-1}}$$

Transformadas ortogonales

cumplen: producto escalar entre funciones base

$$\sum_{n=0}^{N-1} b_k[n] b_l[n] = E_k \delta[k-l] = \begin{cases} E_k & k=l \\ 0 & \text{resto} \end{cases}$$

y se deduce:

$$a_k[n] = \frac{1}{E_k} b_k[n]$$

Transformadas ortonormales

se cumple: $E_k = 1$

$$\Rightarrow \quad a_k[n] = b_k[n]$$

$$\overline{\overline{A}} = \overline{\overline{B}}^T$$

Parseval: $\sum_{n=0}^{N-1} |x[n]|^2 = \sum_{k=0}^{N-1} |X[k]|^2$

Potencia de error al quitar coef.

$x[n] \xrightarrow{T} X[k] = [X[0]\ X[1]\ X[2]\ X[3] \dots X[N-1]]$

aproximación ↓ despreciamos coefs $k \geq L$

$x_a[n] \xrightarrow{T} X_a[k] = [X[0]\ X[1]\ X[2]\ 0\ \dots\ 0]$

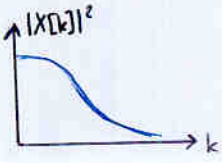
entonces se tiene error $e[n] = x[n] - x_a[n]$

$e[n] \xrightarrow{T} X_e[k] = [0\ 0\ 0\ X[3]\ \dots\ X[N-1]]$

por tanto, por Parseval $\sum |e[n]|^2 = \sum_{k=L}^{N-1} |X[k]|^2$

El error introducido al despreciar $X[k_0]$ de $\{X[k]\}$ es $|X[k_0]|^2$

En general $|X[k]|^2$ disminuye al aumentar k
- No tx últimos coefs
- muestreo zonal



· Transformadas de bloque

Interesa:

- Baja complejidad (muy importante en video)
- Compactación de la energía
- Decorrelación de los coeficientes (para codificación independiente)

Nombre	Compact.	Real/Compl.	Complejidad	Fija/var.
KLT	óptima	real	alta	var.
Haar	baja	real	baja	fija
WHDT	baja	real	baja	fija
DFT	alta	compleja	media	fija
DCT	alta	real	media	fija

→ Transformada Karhunen-Loève

$$\bar{x}_c = \bar{\bar{A}} \cdot \bar{x}$$

A = autovectores en filas
B = autovectores en columnas

$$x[n] \rightarrow \bar{\bar{R}}_x \rightarrow \begin{cases} a_k = \text{autovector k-ésimo} = b_k \\ |X[k]|^2 \text{ (pot)} = \text{autovalor k-ésimo} \end{cases}$$

y la matriz de autocorrelación de los coeficientes

$$\bar{\bar{R}}_c = \bar{\bar{\Lambda}} = \begin{bmatrix} \lambda_0 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_{N-1} \end{bmatrix}$$

decorrelación entre coef

diagonal = potencia coef k

Recuerda: matriz de autocorrelación

$$\bar{\bar{R}}_x = E\{\bar{x} \cdot \bar{x}^T\} \qquad \bar{x} \equiv \text{V.A.'s}$$

$$\bar{\bar{R}}_x = \begin{bmatrix} R[0] & R[1] & \cdots & R[N-1] \\ R[1] & R[0] & R[1] & \\ \vdots & R[1] & R[0] & \ddots \\ & & \ddots & R[1] \\ R[N-1] & & R[1] & R[0] \end{bmatrix}$$

$R[0] = \sigma_x^2$ la pot

si x es AR-1 $\hookrightarrow \bar{\bar{R}}_x = \sigma_x^2 \cdot \rho^{|m|}$

Autovectores/valores
$|R - \lambda I| = 0 \rightarrow \lambda_i$
$R \cdot v_i = \lambda_i \cdot v_i \rightarrow v_i$
(1 grado libertad)

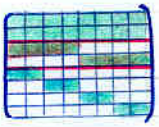
↳ Maximiza la compactación de energía
↳ Decorrela los coeficientes
] → transformada óptima

→ Es adaptativa, depende de $\bar{\bar{R}}_x$ → alta complejidad

→ Transformada Haar discreta

- MUY baja complejidad (se usaba al principio)
- a mayor orden del coef k, de menor nº de muestras temporales depende

ej: $A_8 =$



→ Transformada discreta del coseno DCT

$$b_k[n] = \sqrt{\frac{2}{N}}\, \alpha[k] \cdot \cos\left(\frac{\pi k}{N}\left(n + \frac{1}{2}\right)\right)$$

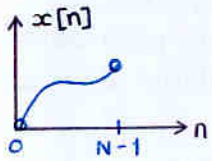
funciones base

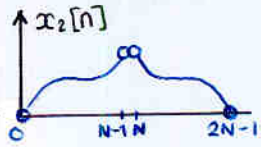
garantiza la ortonormalidad $\alpha[k] = \begin{cases} \sqrt{\frac{1}{2}} & k = 0 \\ 1 & \text{resto} \end{cases}$

- casi óptima si $x[n]$ es AR-1 con $\rho \rightarrow 1$ (buen modelo para las imágenes
- Guarda una relación con la FFT que permite un cálculo eficiente

Relación entre DFT y DCT

sea $x[n]$ $\qquad x_2[n] = \begin{cases} x[n] & n \in [0, N-1] \\ x[2N-1-n] & n \in [N, 2N-1] \end{cases}$





$\downarrow$ DCTN $\qquad \downarrow$ DFT2N

$X_{DCT}[k] \qquad X_{DFT}[k]$

se tiene que $X_{DCT}[k] = X_{DFT}[k] \cdot \beta[k] \cdot e^{-j\frac{\pi k}{2N}}$

$\hookrightarrow \begin{cases} \frac{1}{2\sqrt{N}} & k = 0 \\ \frac{1}{\sqrt{2N}} & k \neq 0 \end{cases}$

Recuerda que la DFT era la TF de la repetición periódica → saltos bruscos → altas frecuencias

En cambio ahora vemos que la DCT es la TF de la repetición periódica de $x_2[n]$ → sin saltos bruscos → más energía en bajas frecuencias

Transformadas 2D

las funciones base son MATRICES NxN

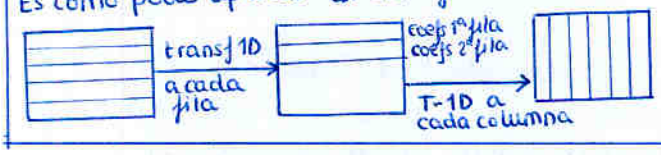
Transformada 2D ortonormal de una imagen NxN:

$$\begin{cases} X[k,l] = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x[n,m]\, b_{k,l}[n,m] \\ x[n,m] = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} X[k,l]\, b_{k,l}[n,m] \end{cases}$$

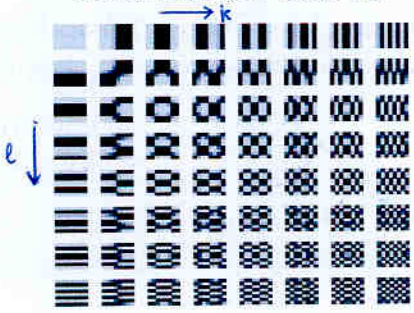
Transformadas 2D separables

$$b_{k,l}[n,m] = b_k[n] \cdot b_l[m] \Rightarrow \begin{cases} \bar{\bar{X}}_c = \bar{\bar{A}} \cdot \bar{\bar{X}} \cdot \bar{\bar{A}}^T \\ X = A^T \cdot X_c \cdot A \end{cases}$$

Es como poder aplicar la transf 1D dos veces



Funciones Base DCT-2D



Bloques

- La transformada no se aplica a toda la imagen sino a bloques

ventaja: - menos memoria
- hardware sencillo

inconven: - menor compactación de energía (la pérdida es despreciable para bloques que superen cierto tamaño que depende de f_{sx}, f_{sy})
- correlación entre $X[0,0]$ de distintos bloques

Tamaño típico 8x8

Coeficientes DCT

- DC ($X[0,0]$)
  - componente continua
  - sin estadística definida
  - gran varianza
  - error de cuantific. MUY visible
- AC ($X[k,l]$)
  - estadísticamente laplacianos de varianza $\sigma^2_{k,l}$
  - $\sigma_{k,l}$ disminuye con k y l
  - muchos coef $\simeq 0$

Cuantificación coeficientes

$$i_{k,l} = \text{sign}(X[k,l]) \cdot \text{round}\left(\frac{|X[k,l]|}{\Delta_{k,l}}\right)$$

→ cuantif uniforme (cercano al óptimo porq usamos codif entrópica)
- paso $\Delta_{k,l}$ depende de k y l (menores pasos para menores frecuencias)

Decuantificación: $\hat{X}_{k,l} = i_{k,l} \cdot \Delta_{k,l}$

Transmisión de los coeficientes

Tras cuantificar aparecen muchos ceros en las frecuencias altas; no los transmitimos.

¿cómo señalizamos qué coef he transmitido y cuales no?

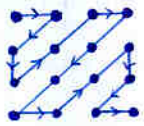
a) Matriz binaria

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Tx tabla = 1bit/coef (tasa elevada)

1 = He tx coef
0 = coef es nulo

b) Exploración en zigzag y codificación run-length



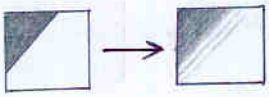
- Ya que los $i_{k,l} \neq 0$ están agrupados en bajas frecuencias, serán frecuentes las ráfagas de ceros

$11100100....0 \rightarrow 1_3\, 0_2\, 1_1\, 0_{10}$

Principales degradaciones con DCT

Degradación aumenta con el factor de compresión

- Pérdida de nitidez: faltan coeficientes AC

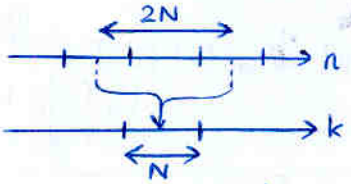


- Rizado (Gibbs): en los cambios bruscos
- Efecto bloque: debido a las discontinuidades por la codificación independiente de cada bloque
  - ↳ es la degradación más visible (SVH tiene alta sensibilidad ante errores dispuestos uniformemente)

Reducción del efecto bloque

- Postprocesamiento:
  - filtrado paso bajo local → afectará también a contornos reales
  - algoritmos de deblocking → discriminar qué transiciones son por efecto bloque y cuales no
- Nuevas técnicas de codificación basadas en bancos de filtros

LOT (Lapped Orthogonal Transform)



- Funciones base de longitud 2N
- Banco de filtros
  - N subbandas
  - filtros longitud 2N

Los bloques ya no son independientes

DWT (Discrete Wavelet Transform)

- es como una codificación subbanda
- filtrado sobre TODA la imagen (elimina totalmente el efecto bloque)
- diezmado crítico 2 bandas, repitiéndolo para la banda pasobajo

se usa en JPEG-2000

6. JPEG

- Estándar para imágenes fijas (ISO - CCITT)
- Requisitos:
  - calidad/tasa parametrizable
  - complejidad razonable
  - 4 modos de operación
  - aplicable a mayoría de las imágenes

ejemplo:

$$i_{k,l} = \begin{bmatrix} 86 & -3 & 4 & 0 & \cdots \\ 3 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & & \\ 0 & 0 & \cdots & & \end{bmatrix}$$

tras cuant.

se transmite:

{86 (DPCM), (0,-3), (0,3), (2,4), (1,1), EOB}

Coeficientes en JPEG

- **Coeficiente DC**

  Existe gran correlación entre el coef DC de bloques vecinos que hay que explotar

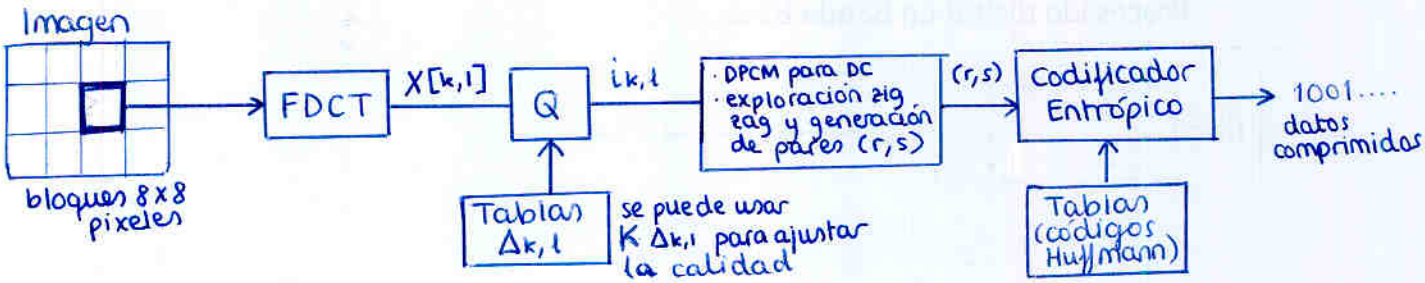
  → Se usa DPCM
  - predicción lineal con bloque anterior
  - bucle abierto (!)

- **Coeficiente AC**
  - Exploración de $X[k,l]$ en zig-zag → secuencia 1D
  - La transforma en una secuencia de pares (r, s)
    - s: valor del coef AC ≠ 0
    - r: nº ceros desde el anterior
    - si r > 16: símbolo F0 (11 bits) sustituye a 16 ceros
  - cuando sólo quedan ceros: EOB (End Of Block) (4 bits) ← GRAN ahorro!

Codificación entrópica

- Se usa codificación entrópica para el error de predicción de DC y para los pares (r, s).
  - Huffmann: tablas por defecto / tx tablas
  - Cod. aritmética: mejora 5-10% pero más complejidad

Esquema general JPEG



Modos de operación

- Secuencial:
  se tx bloque tras bloque
- Sin pérdidas:
  No usa DCT sino DPCM
  Ya no se usa, JPEG-LS mejor
- Jerárquico:
  Tx imágen a distintas resoluciones (ej: para distintos dispositivos)
- Progresivo:
  Divido el bitstream en partes
  - MSB de bajas frec
  - resto bits bajas frec
  - MSB altas frec
  - resto bits altas frec

  → A mayor cantidad de bits recibidos, mejor se ve la imagen
  → útil en browsing

Color en JPEG

a) Codificar 3 imágenes (R, G, B)

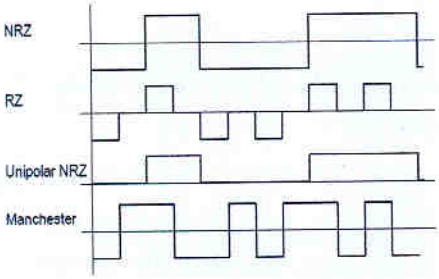
b) más habitual:
- imagen de luminancia M x M
- dos imagenes de crominancia (B-Y), (R-Y) $\frac{M}{2} \times \frac{M}{2}$

total nº pixeles: $1'5M^2$

Tema 6. Transmisión de señales en com. digitales

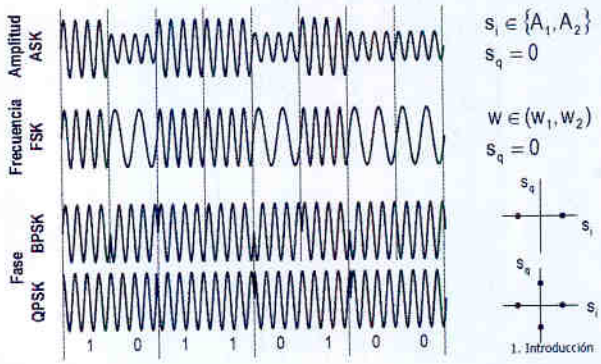
Modulaciones digitales

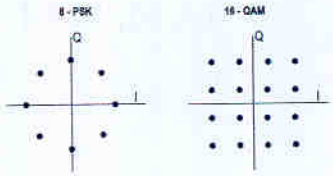
Banda base:



Paso banda:

$$x(t) = A[s_i(t)\cos(wt+\theta) - s_q(t)\sin(wt+\theta)]$$

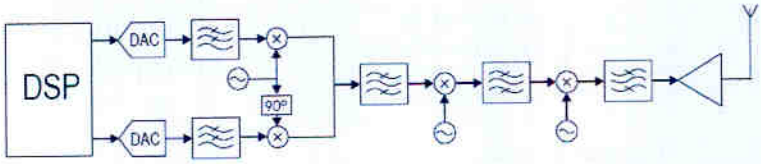


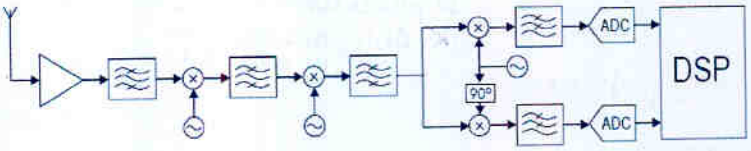


Transmisor con procesamiento digital

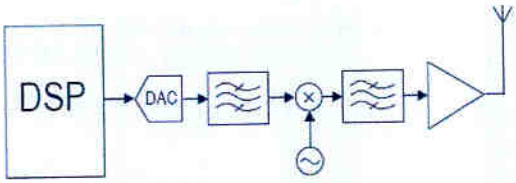
Se tiende a llevar lo digital hasta la frecuencia que se pueda
(reconfigurable, replicable, insensible a Tª)

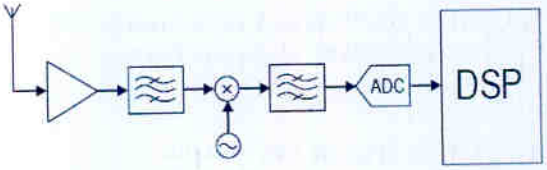
Procesado digital en **Banda Base**:



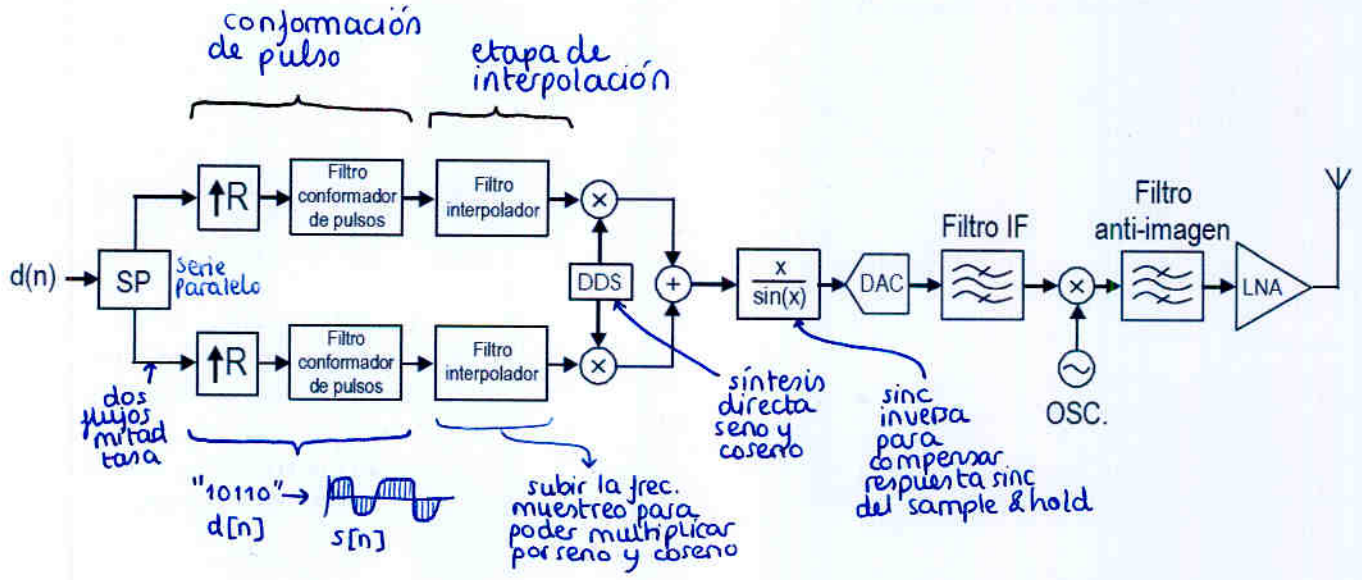


Procesado digital en **Frecuencia Intermedia**





Generación de señales



• Conformación de pulso

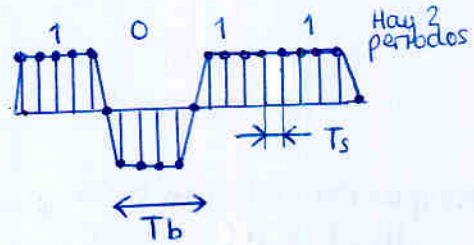
· Necesitamos generar pulsos a partir de los símbolos de información

$$s(t) = \sum_k \overbrace{d[k]}^{\text{señal digital}} \cdot \overbrace{p(t - kT_b)}^{\text{pulso de cada bit}}$$

· ¿Podemos recrear dicha señal s(t) digitalmente? →

$$s[n] = s(t)\big|_{t=nT_s}$$

$$s[n] = \sum d[k] \cdot p(nT_s - kT_b)$$



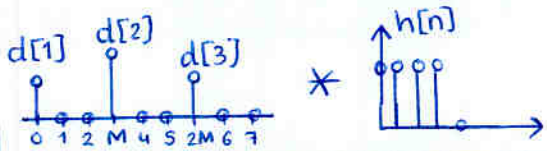
• Si hacemos que $\boxed{T_b = M \cdot T_s}$

$$s[n] = \sum_k d[k]\, p(nT_s - kMT_s)$$

llamando $\boxed{h[n] = p(nT_s)}$ pulso muestreado

$$\rightarrow s[n] = \sum d[k]\, h(n-kM)$$

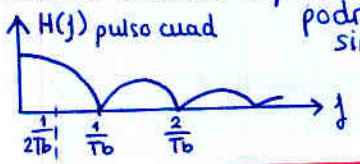
$$\Rightarrow \boxed{s[n] = \left[\sum_k d[k]\, \delta[n-kM]\right] * h[n]}$$



$$d[n] \rightarrow \boxed{\uparrow M} \rightarrow \boxed{h[n]} \rightarrow s[n]$$

¿Qué impulso usamos?

· Pulso cuadrado no porque tiene ancho banda infinito y no podríamos crearlo sin aliasing



β=0 β=1

β: factor roll-off (exceso de AB)

Interesa pulso con ancho $B = \frac{1}{T_b} \cdot \frac{1+\beta}{2}$

$$\left.\begin{array}{l} f_s = M/T_b \\ f_s > 2B = \frac{1+\beta}{T_b} \end{array}\right\} \begin{array}{l} M \geq 2 \\ \text{típico } M=4 \end{array}$$

Pulso COSENO ALZADO

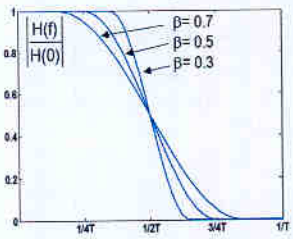
· era difícil conseguirlo analógicamente

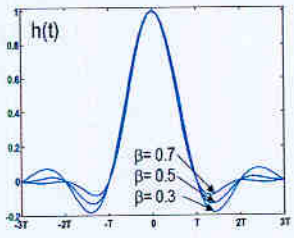
· en digital es muy sencillo

$$H(f) = \begin{cases} \sqrt{T} & 0 \leq |f| \leq \frac{1-\beta}{2T} \\ \sqrt{T}\cos^2[f\cdot(1-\alpha)/2T] & \frac{1-\beta}{2T} \leq |f| \leq \frac{1+\beta}{2T} \\ 0 & \text{resto} \end{cases}$$

- Transmisión en canales con limitación de ancho de banda
- Satisface el criterio de Nyquist para eliminar ISI
- Respuesta truncada cercana a la ideal

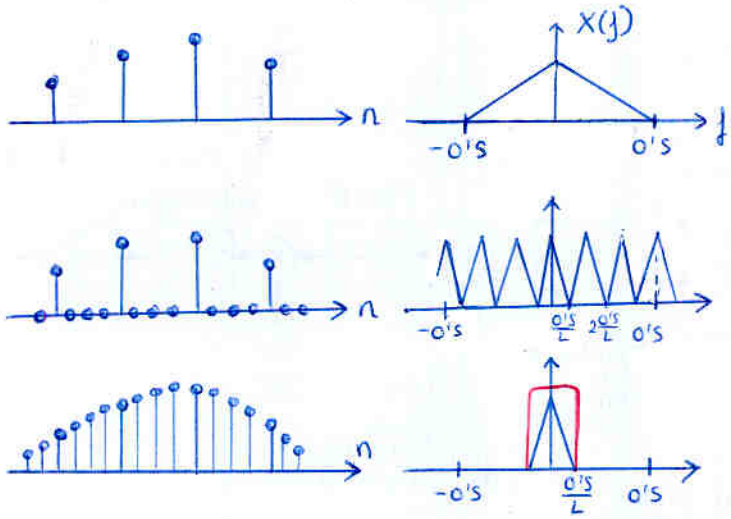
β: factor de *"roll off"* (exceso de ancho de banda)

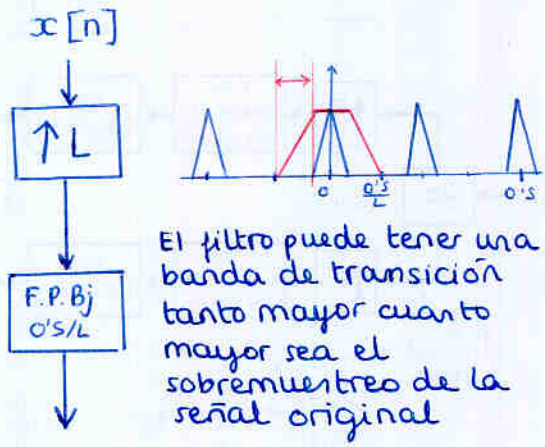




Etapa de interpolación

Recordatorio: interpolación





El filtro puede tener una banda de transición tanto mayor cuanto mayor sea el sobremuestreo de la señal original

¿Cómo podemos hacer una interpolación eficiente? (hay que llevar a FI!!)

en lugar de → ↑8 → H →

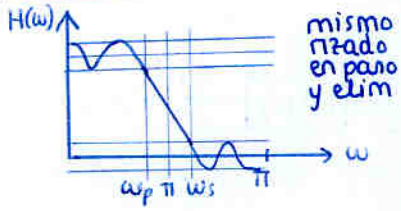
hacemos → ↑2 → Media Banda → ↑4 → CIC →

Media Banda: buena respuesta y eficiente

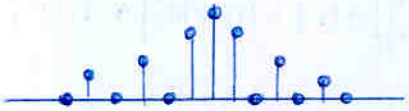
CIC:
- respuesta mala: no pasa nada ya que está sobremuestreado
- cálculo MUY eficiente (sólo sumas!)

a) Filtro media banda

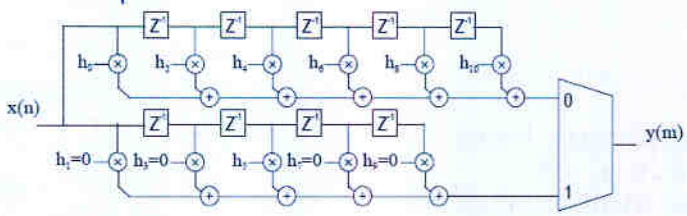
- Respuesta antisimétrica $H(\omega) = 1 - H(\pi - \omega)$ ⟹
  - Muestras impares (salvo la central) = 0
  - Muestras pares tienen simetría par



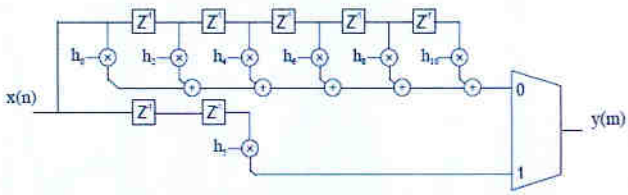
mismo rizado en paso y elim



polifase:

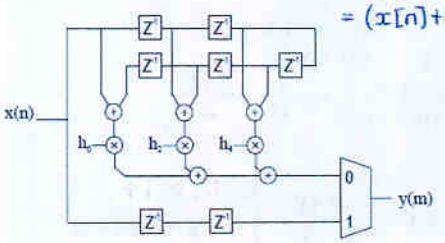


- Aprovecha los coeficientes nulos



- Normalizar coeficientes: $h_5=1$
- Aprovechar simetría

$x[n]h_0 + x[n-10]h_{10}$ $h_0=h_{10}$
$= (x[n]+x[n-10])\cdot h_0$



b) Filtros CIC (Cascade Integrator Comb)

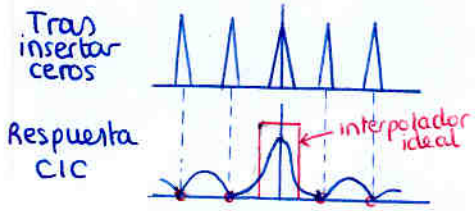
- Se utilizan para interpolación o diezmado de orden elevado sobre señales sobremuestreadas
- No hay multiplicaciones, sólo sumas

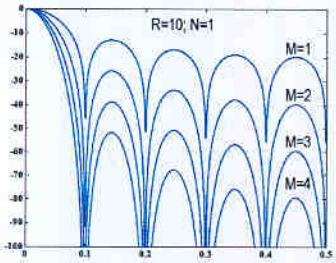
- **Función de transferencia:**
  - sencilla
  - nulos en $f = \frac{k}{NR}$ que justamente elimina las imágenes
  - Poco AB util, por eso requiere señal sobremuestreada

$$H(z) = \left[\frac{1-z^{-RN}}{1-z^{-1}}\right]^M$$

R: factor de interpolación
M: nº etapas
N: N×R×M → orden del filtro

$$|H(f)| = \left|\frac{\sin(\pi N R f_s)}{\sin(\pi f_s)}\right|^M$$





- **Implementación interpolador**

Hace uso de

$$\rightarrow \boxed{\uparrow R} \rightarrow \boxed{H(z^R)} \rightarrow \;\equiv\; \rightarrow \boxed{H(z)} \rightarrow \boxed{\uparrow R} \rightarrow$$

$$\rightarrow \boxed{\uparrow R} \rightarrow \boxed{H(z) = \left(\frac{1-z^{-RN}}{1-z^{-1}}\right)^M} \rightarrow$$

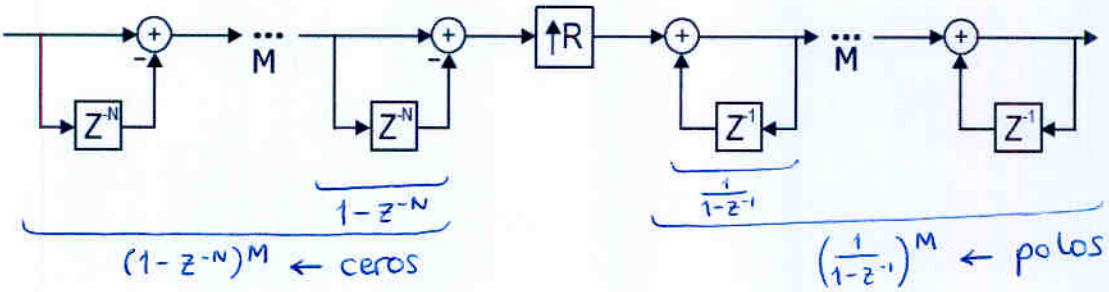
$$\equiv$$

$$\rightarrow \boxed{\uparrow R} \rightarrow \boxed{H_{num}(z^R) = (1-z^{-RN})^M} \rightarrow \boxed{H_{den}(z) = \left(\frac{1}{1-z^{-1}}\right)^M} \rightarrow$$

$$\equiv$$

$$\rightarrow \boxed{H_{num}(z) = (1-z^{-N})^M} \rightarrow \boxed{\uparrow R} \rightarrow \boxed{H_{den}(z) = \left(\frac{1}{1-z^{-1}}\right)^M} \rightarrow$$

$$\equiv$$



- **Implementación diezmador**

$$\rightarrow \boxed{H} \rightarrow \boxed{\downarrow R} \rightarrow x \rightarrow \boxed{H_{den}(z)} \rightarrow \boxed{H_{num}(z^R)} \rightarrow \boxed{\downarrow R} \rightarrow x \rightarrow \underbrace{\boxed{H_{den}(z)}}_{\text{peine de M polos}} \rightarrow \boxed{\downarrow R} \rightarrow \underbrace{\boxed{H_{num}(z)}}_{\text{peine de M ceros}} \rightarrow$$

EXAMEN DE APLICACIONES DEL TRATAMIENTO DIGITAL DE LA SEÑAL
(12/9/2006)

- Profesores: Josep Prades y Alberto González.
- La duración del examen es de 1h 30m.
- Resuelva las cuestiones en el espacio entre ellas.
- Si tiene alguna duda no se levante del asiento, alce la mano y el profesor le atenderá.
- No se permite el uso de calculadora.

CUESTIÓN 1 (2 puntos)

Sea x[n] una señal aleatoria discreta estacionaria de media nula. Se quiere codificar x[n] con un codificador DPCM que utiliza un predictor de primer orden con coeficiente de predicción unidad.

1. Si ρ es el coeficiente de correlación de la señal ($\rho = R_x[1]/R_x[0]$), ¿en qué intervalo de valores debe estar ρ para obtener una ganancia de predicción Gp mayor que la unidad? Suponga que el ruido de cuantificación es despreciable.

$$g_p = \frac{SNR_{DPCM}}{SNR_{PCM}} = \frac{\sigma_x^2}{\sigma_e^2}$$

$x[n]$ = V.A. media nula, $\rho = R_x[1]/R_x[0]$

$e[n] = x[n] - x_p[n] = x[n] - x[n-1]$; $h_{pred} = \overset{1}{a_1} z^{-1}$

$$\boxed{\begin{aligned}\sigma_e^2 &= E\{(e[n])^2\} = E\{(x[n]-x[n-1])^2\} = E\{x^2[n] - 2x[n]x[n-1] + x^2[n-1]\}\\ &= E\{(x[n])^2\} - 2E\{x[n]x[n-1]\} + E\{(x[n-1])^2\}\\ &= \sigma_x^2 - 2R_x[1] + \sigma_x^2\\ &= 2(\sigma_x^2 - R_x[1]) = 2(\sigma_x^2 - \underbrace{R_x[0]}_{\sigma_x^2}\cdot\rho) = 2\sigma_x^2(1-\rho)\end{aligned}}$$

$\rightarrow g_p = \frac{1}{2(1-\rho)} > 1$

$\rightarrow \boxed{\rho > 0'5}$ para $a_1 = 1$

2. Suponga ahora que x[n] es una señal AR de orden 1, esto es, puede obtenerse mediante:

$$x[n] = a\,x[n-1] + e[n]$$

donde: $0 < a < 1$, y, e[n] es ruido blanco de potencia unidad. Para codificar x[n] se consideran dos alternativas: codificación PCM y codificación DPCM del apartado anterior. Indique cómo tomar la decisión en función del valor de a.

$R_x[0] = E\{(ax[n-1])^2 + \cancel{2ax[n-1]e[n]} + e[n]^2\} = a^2\sigma_x^2 + 1 \overset{def}{=} \sigma_x^2 \rightarrow \sigma_x^2 = \frac{1}{1-a^2}$

$R_x[1] = E\{a\,x[n-1]x[n-1] + \emptyset + \cancel{e[n]x[n-1]}\} = a\,R_x[0]$

$R_x[m] = R_x[0]\cdot a^m = \frac{a^m}{1-a^2}$

El error de predicción será $x[n] - x_p[n] = x[n] - \overbrace{a\,x[n-1]}^{\text{predictor óptimo de las ecs de Yule-Walker}} = e[n]$

$a_1 R_x[0] = R_x[1]$

$a_1 = \frac{R_x[1]}{R_x[0]} = a$

$\sigma_e^2 = 1$ (del enunciado)

$g_p = \frac{\sigma_x^2}{\sigma_e^2} = \frac{1/1-a^2}{1} = \frac{1}{1-a^2} > 1 \Rightarrow$ $\boxed{\text{basta con que } |a|<1 \text{ para que interese usar DPCM con } a_1 = a}$

interesa

CUESTIÓN 2 (2 puntos)

Se quiere codificar una señal de audio monofónica muestreada a 48 kHz con MPEG-1 capa III. El análisis psicoacústico determina que no hay posibilidad de preeco. Sobre la MDCT que se aplicaría en la codificación de la señal:

1. ¿Cuál es el tamaño del bloque de muestras de audio sobre el que se aplica la MDCT?

Caso normal – 18 muestras

2. ¿Cuál es el número de coeficientes que se originan cada vez que se aplica la MDCT?

MDCT → 2N muestras → N coefs
18 → 9

3. ¿Cuál es la resolución frecuencial de cada coeficiente MDCT?

Cada una de las 32 subbandas tiene ancho $\frac{(15/2)}{32} = 750$ Hz

9 coef/subbanda → $\frac{750 \text{ Hz}}{9 \text{ coef}} = 83'\hat{3}$ Hz/coef

4. Si el análisis psicoacústico determina que hay posibilidad de preeco, ¿cuál es la resolución frecuencial de cada coeficiente MDCT?

MDCT caso preeco: 6 muestras → 3 coefs

$\frac{750 \text{ Hz}}{3 \text{ coef}} = 250$ Hz/coef

CUESTIÓN 3 (2 puntos)

Defina brevemente los siguientes conceptos en el ámbito de la codificación de voz:

1.- Pitch.

El pitch es la "pseudofrecuencia" $\left(\frac{1}{\text{pseudoperiodo}}\right)$ de un sonido sonoro, que es pseudoperiódico

2.- Algoritmo de Levinson-Durbin.

- Algoritmo para resolver las ecs de Yule-Walker de forma eficiente
- Aprovecha que $R_x[m]$ es matriz Toeplitz
- Logra un coste $O(p^2)$ en lugar de $O(p^3)$
- Proporciona durante su resolución los coeficientes PARCOR k_i

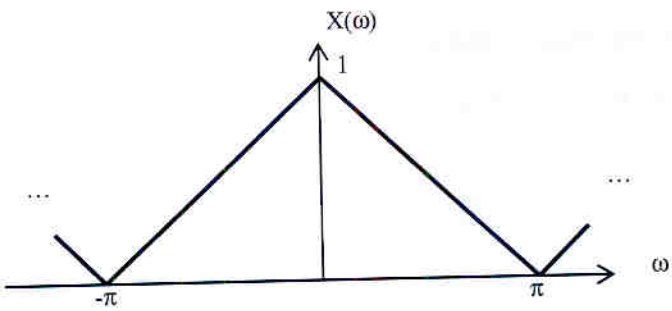
PARCOR → implementación en celosía
→ control de estabilidad $|k_i|<1$
→ menor efecto al cuantificarlos

3.- Codificador CELP.

un codificador LPC que como señal de excitación utiliza una de las señales en su 'biblioteca' (codebook) (indicada con $\log_2 M$ bits) posiblemente sumada a un tramo de voz sintetizado y almacenado previamente (codebook dinámico)

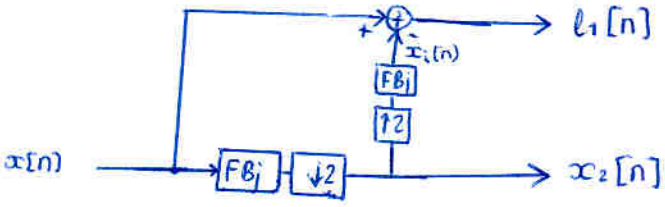
CUESTIÓN 4 (2 puntos)

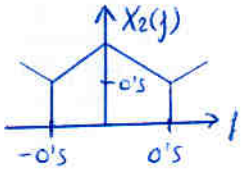
Sea x[n] una señal cuya transformada de Fourier viene dada por:

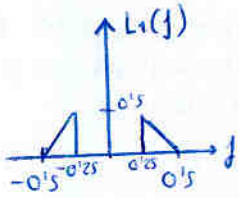


Suponga que se quiere transmitir x[n] utilizando codificación piramidal de dos niveles. El codificador genera dos señales, $l_1[n]$ y $x_2[n]$, a partir de x[n]. Suponga que el codificador y el decodificador utilizan filtros ideales.

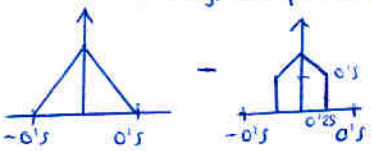
1. Indique el diagrama de bloques del codificador-decodificador.
2. Dibuje las transformadas de Fourier $L_1(\omega)$ y $X_2(\omega)$.
3. Demuestre que con el codificador-decodificador propuesto en el apartado 1. y, en ausencia de ruido de cuantificación, se puede recuperar la señal original sin distorsión.

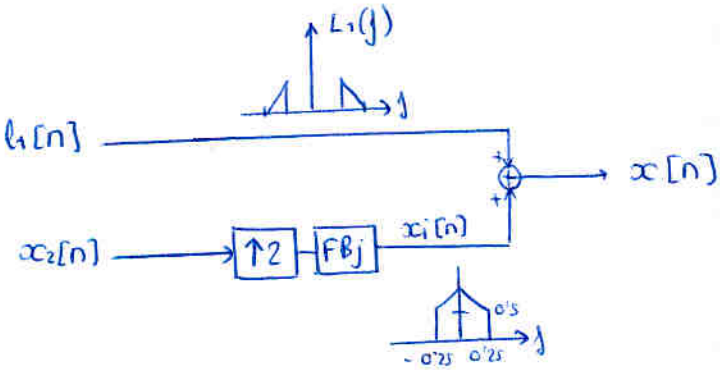






$$L_1(f) = X(f) - \underbrace{X_i(f)}_{X_2(f)\text{ interpolada}}$$

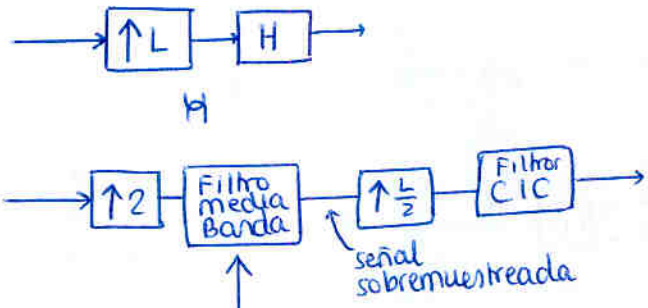




CUESTIÓN 5 (2 puntos)

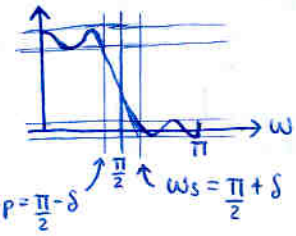
Explique el funcionamiento de un filtro de media banda, para ello responda las siguientes cuestiones: ¿para qué se utilizan los filtros de media banda? ¿cómo son su respuesta en frecuencia y su respuesta impulsional? y ¿cómo se implementan en la práctica?

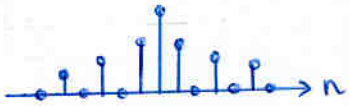
Para interpolar a frecuencias muy altas



cumplen que: $H(\omega) = -H(\omega - \pi)$ $\Longrightarrow$ $\begin{cases} h[n]=0 \text{ para coefs impares salvo el central} \\ h[n] \text{ tiene simetría par en los coeficientes pares} \end{cases}$

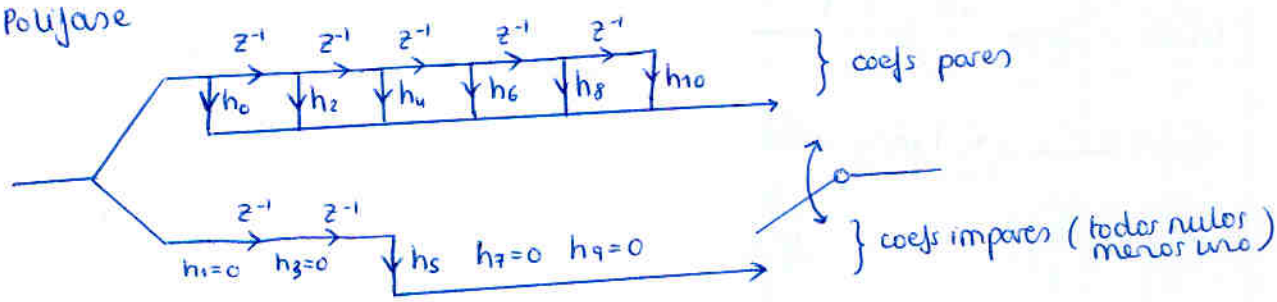
mismo rizado en bandas de paso y elim.



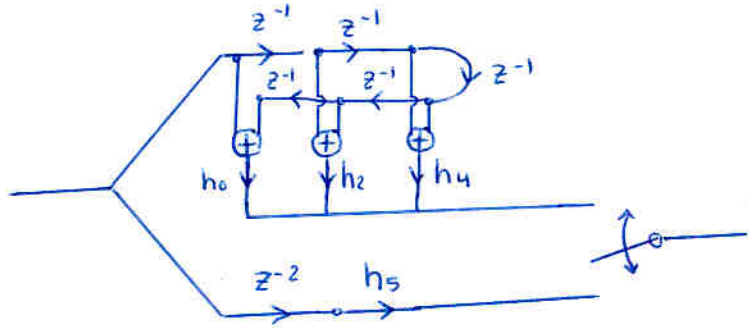


se aprovechan las dos características para hacer:

Polifase



y, aprovechando la simetría de los coef pares:

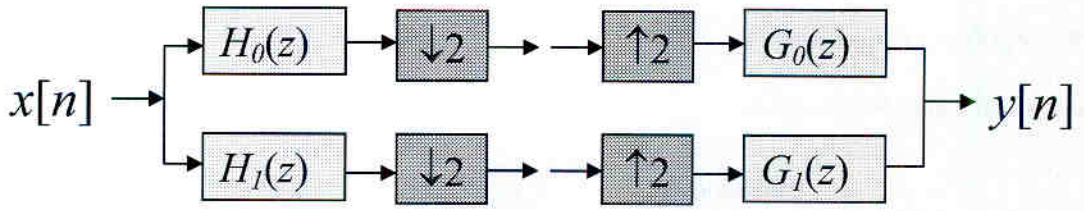


EXAMEN DE APLICACIONES DEL TRATAMIENTO DIGITAL DE LA SEÑAL
(14/6/2006)

- Profesores: Josep Prades y Alberto González.
- La duración del examen es de 1h 30m.
- Resuelva las cuestiones en el espacio entre ellas.
- Si tiene alguna duda no se levante del asiento, alce la mano y el profesor le atenderá.
- No se permite el uso de calculadora.

CUESTIÓN 1 (2 puntos)

La figura muestra un esquema de análisis-síntesis subbanda de una señal. Se pretende estudiar el diseño de los filtros de análisis y síntesis que aparecen en el esquema.



Suponiendo que no se producen errores ni distorsión de la señal en el canal la relación entrada-salida del sistema en el dominio frecuencial viene dada por la siguiente expresión:

$$Y(\omega) = \frac{1}{2}\left[H_0(\omega)\cdot G_0(\omega) + H_1(\omega)G_1(\omega)\right]X(\omega)$$
$$+\frac{1}{2}\left[H_0(\omega-\pi)\cdot G_0(\omega) + H_1(\omega-\pi)G_1(\omega)\right]X(\omega-\pi)$$

1.- Indique la/s condición/es que deben cumplir los bancos de filtros para que no aparezcan términos debidos al "aliasing" frecuencial en la señal de salida. Razone su elección.

Para que no haya aliasing podemos escoger:

$G_0(\omega) = 2H_1(\omega-\pi)$
$G_1(\omega) = -2H_0(\omega-\pi)$

de forma que se anule el segundo término

2.- Encuentre la relación entrada-salida del sistema en el dominio frecuencial cuando se utiliza un solo prototipo paso bajo con respuesta en frecuencia $H(\omega)$ para diseñar todos los filtros del esquema de forma que no haya aliasing.

$H_0(\omega) = H(\omega)$
$H_1(\omega) = H(\omega-\pi)$ $\rightarrow$ $h_0[n] = h[n]$, $h_1[n] = (-1)^n h[n]$

$G_0(\omega) = 2H(\omega)$
$G_1(\omega) = -2H(\omega-\pi)$ $\rightarrow$ término de aliasing se anula

$Y(\omega) = \left[H^2(\omega) - H^2(\omega-\pi)\right]X(\omega)$

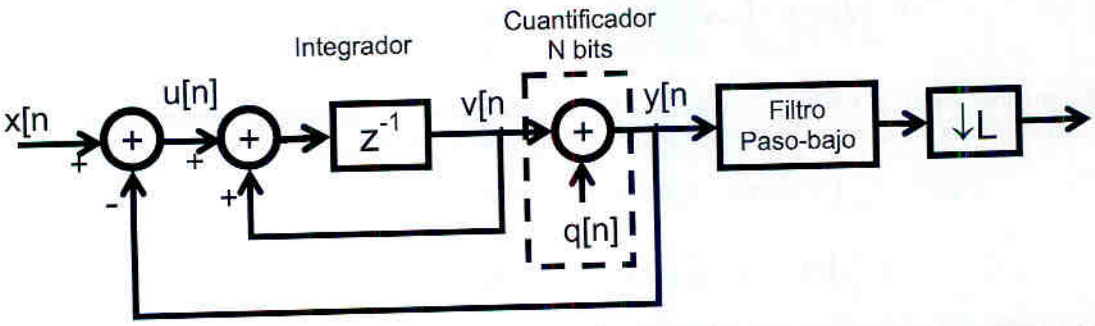
3.- Demuestre que, para el caso del apartado anterior, el módulo de la respuesta en frecuencia del sistema es proporcional a: $|H(\omega)|^2 + |H(\omega-\pi)|^2$, cuando el filtro prototipo paso bajo es un filtro FIR de fase lineal con un número par de coeficientes, N, cuya respuesta en frecuencia viene dada por: $H(\omega) = |H(\omega)|e^{j\omega\frac{N-1}{2}}$.

$$H_{TOT}(\omega) = H^2(\omega) - H^2(\omega-\pi)$$
$$= |H(\omega)|^2 e^{2j\omega\frac{N-1}{2}} - |H(\omega-\pi)|^2 e^{2j(\omega-\pi)\frac{N-1}{2}}$$
$$= e^{2j\omega\frac{N-1}{2}}\left[|H(\omega)|^2 - |H(\omega-\pi)|^2 \underbrace{e^{-2j\pi\frac{N-1}{2}}}_{e^{-j\pi\underbrace{(N-1)}_{N\text{ par}\to N-1\text{ impar}}} = -1}\right]$$
$$= e^{2j\omega\frac{N-1}{2}}\left[|H(\omega)|^2 + |H(\omega-\pi)|^2\right]$$

Por tanto $|H_{TOT}(\omega)|^2 = |H(\omega)|^2 + |H(\omega-\pi)|^2$

CUESTIÓN 2 (2 puntos)

Sea el circuito discreto de la figura:



1.- Encuentre la relación en términos de transformadas Z entre la salida del sistema y[n] y las señales de entrada x[n] y q[n].

integrador: $H_{int}(z) = \frac{z^{-1}}{1-z^{-1}}$ (fácil de comprobar)

$$Y(z) = Q(z) + V(z)$$
$$= Q(z) + H_{int}(z)\cdot U(z)$$
$$= Q(z) + \frac{z^{-1}}{1-z^{-1}}\cdot(X(z)-Y(z)) \Rightarrow \boxed{Y(z) = \overbrace{(1-z^{-1})}^{H_Q(z)}Q(z) + \overbrace{(z^{-1})}^{H_X(z)}X(z)}$$

2.- El filtro paso bajo de la figura es un filtro paso bajo ideal de ancho de banda en frecuencias discretas normalizadas 0.5/L. Considere la señal x[n] una señal paso bajo de ancho de banda en frecuencias discretas normalizadas inferior a 0.5/L. Demuestre que la relación señal a ruido de cuantificación a la salida del filtro paso bajo viene dada aproximadamente por: $SNR \approx SNR_Q + 30\log(L) - 10\log(\pi^2/3)$, donde SNR_Q es la relación señal a ruido de cuantificación del cuantificador utilizado. Recuerde que:

$$sen(x) \approx x - \frac{x^3}{6}.$$

$$|H_X(z)|^2 = 1$$
$$|H_Q(z)|^2 = H_Q(z)\cdot H_Q^*(z)$$
$$= (1-z^{-1})\cdot(1-z^{-1})^* \quad z = e^{j\omega}$$
$$= (1-e^{j\omega})\cdot(1-e^{j\omega})^*$$
$$= (1-e^{j\omega})\cdot(1-e^{-j\omega})$$
$$= 1 - e^{j\omega} - e^{-j\omega} + 1$$
$$= \ldots = 4\,sen^2\left(\frac{\omega}{2}\right)$$

$$\sigma_x^2 = \sigma_x^2$$
$$\sigma_r^2 = \sigma_q^2 \cdot \int_{-\frac{\pi}{L}}^{\frac{\pi}{L}} |H_Q(\omega)|^2 d\omega$$
$$= \ldots = \sigma_q^2 \cdot \frac{\pi^2}{3}\left(\frac{1}{L}\right)^3$$

$$\Rightarrow SNR = \frac{\sigma_x^2}{\sigma_r^2} = \underbrace{\frac{\sigma_x^2}{\sigma_q^2}}_{SNR_Q} \cdot \frac{1}{\left(\frac{\pi^2}{3}\right)\left(\frac{1}{L}\right)^3}$$ en dB

$$SNR = SNR_Q - 10\log\left(\frac{\pi^2}{3}\right) + 30\log L$$

CUESTIÓN 3 (2 puntos)

Respecto al diagrama de bloques del codificador de voz RPE-LTP utilizado en GSM, indique la utilidad y los parámetros que generan los subsistemas:

1. Análisis LPC a corto plazo.

2. Análisis LTP.

3. Selección y codificación retícula RPE.

CUESTIÓN 4 (2 puntos)

Sean las matrices base siguientes:

$$\mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};\ \mathbf{B} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix};\mathbf{C} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix};\mathbf{D} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Suponga que un bloque de una imagen formado por cuatro píxeles se encuentra almacenado en forma de una matriz como:

$$\mathbf{Imag} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}.$$

Sea el bloque de imagen "transformado":

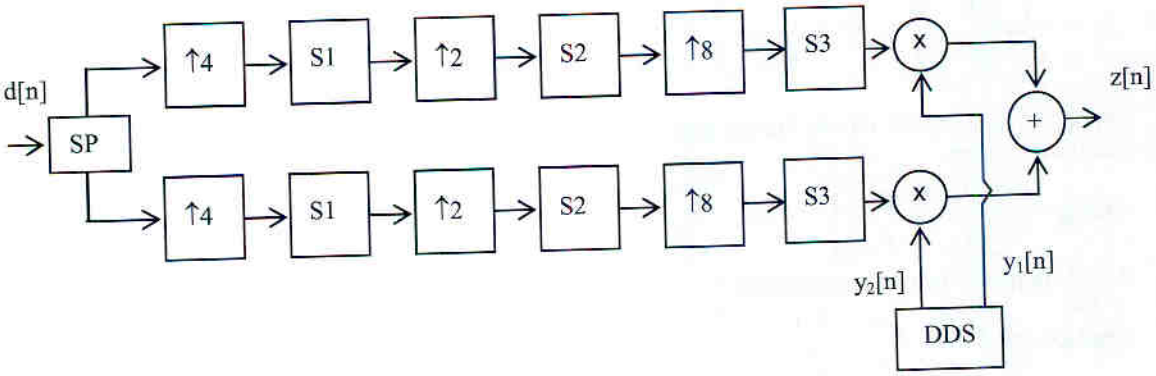
$$\mathbf{Imagtransf} = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

Encuentre los coeficientes transformados, recuerde que la imagen puede recuperarse a partir de la base de la forma:

$$\mathbf{Imag} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = a\mathbf{A} + b\mathbf{B} + c\mathbf{C} + d\mathbf{D}$$, donde a, b, c y d son coeficientes escalares.

NOTA: Los coeficientes de la transformada de bloque de una imagen vienen dados por: $X[k,l] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x[n,m]\, b^*_{k,l}[n,m]$.

CUESTIÓN 5 (2 puntos)

Considere el diagrama de bloques del transmisor digital de la figura en el que la tasa binaria asociada a la secuencia d[n] es 1Mbps.

1. Indique la utilidad de los filtros S1, S2 y S3.
2. Obtenga la respuesta frecuencial ideal de los filtros S2 y S3.
3. Obtenga la expresión de las secuencias $y_1[n]$ e $y_2[n]$. ¿Cómo se puede obtener $y_2[n]$ a partir de $y_1[n]$ utilizando un sistema discreto?