

PARTE I: TÉCNICAS AVANZADAS DE CODIFICACIÓN DE CANAL

1. INTRODUCCIÓN

Shannon:
RSC
sin errores

$$C = W \cdot \log_2 \left(1 + \frac{S}{N} \right)$$

capacidad bits/s

ancho banda Hz

$$S = k \cdot E_b = R \cdot E_b$$

bits símbolo

seg símbolo

$$N = W \cdot N_0$$

Hz

W Hz

$$\frac{S}{N} = \frac{R}{W} \frac{E_b}{N_0}$$

eficiencia espectral

$$\frac{C}{W} = \log_2 \left[1 + \frac{R}{W} \cdot \frac{E_b}{N_0} \right]$$

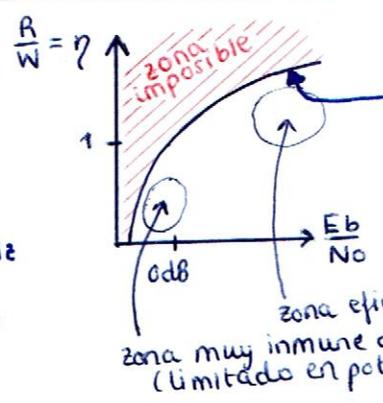
En el caso IDEAL $R = C \Rightarrow \frac{C}{W} = \frac{R}{W} = \eta$
 $\Rightarrow \eta = \log_2 \left(1 + \eta \frac{E_b}{N_0} \right)$

si FIJAMOS una BER
ej 10^{-5}

Entonces podemos dibujar puntos para varias modulaciones

BPSK $\Rightarrow \eta = 1 \text{ bps/Hz}$

QPSK $\Rightarrow \eta = 2 \text{ bps/Hz}$



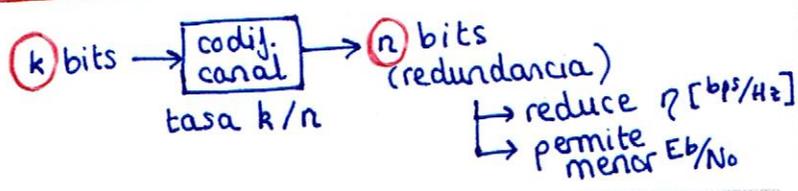
Límite de Shannon: $\frac{E_b}{N_0} \geq \frac{2^\eta - 1}{\eta}$

si $\eta \rightarrow 0$
 $E_b/N_0 \geq \ln(2)$

zona muy inmune al ruido: (FSK, códigos convolucionales, ...)

(limitado en potencia P)

Codificación de canal

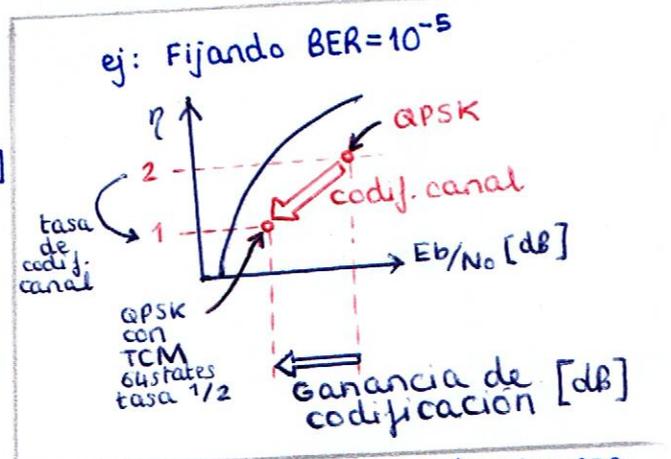


Para ser justos al ver ganancias de codif. habría que comparar QPSK sin codif $\rightarrow \eta = 2$

8-PSK con TCM de 64 estados $\rightarrow \eta = 2$

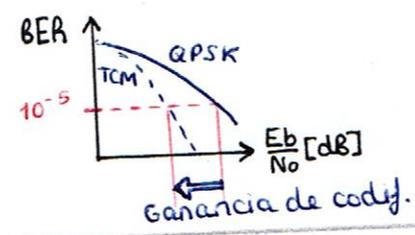
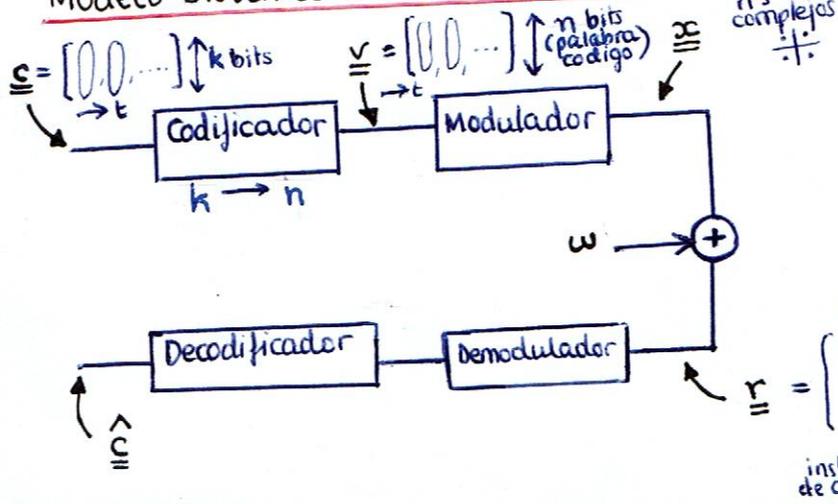
tasa $2/3$

Y en efecto EXISTE ganancia



se puede representar para todas las BER en las típicas curvas E_b/N_0 vs. BER (pero entonces no vemos la eficiencia)

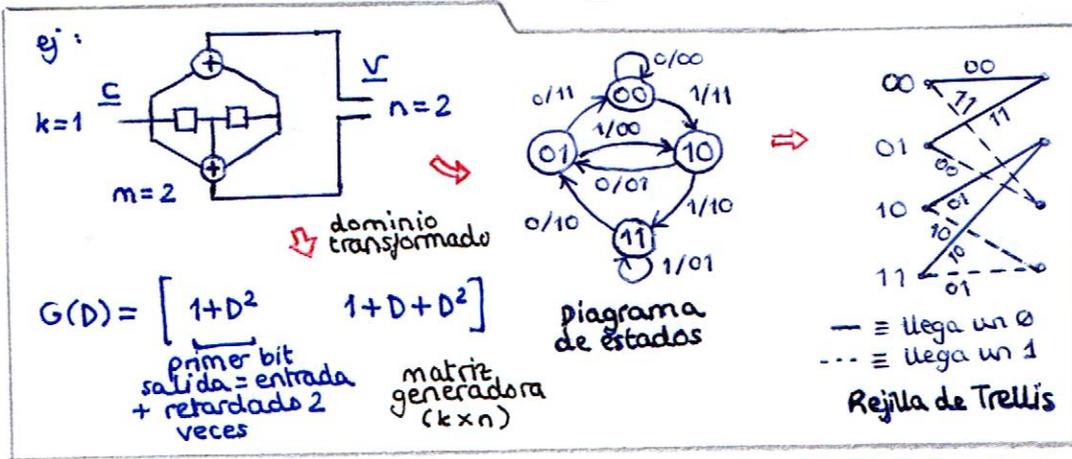
Modelo sistema de Comunicación



2. CODIFICACIÓN CONVOLUCIONAL

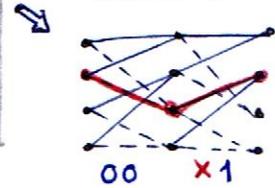
- Introdutor Elias 1954
- Viterbi plantea decodificación ML 1967
- ↳ se usa en telefonía móvil, WLAN y Trellis coded Modulation

CC (n bits salida, k bits entrada, m memoria)
 ↳ 2^m estados



Puncturing (Perforado)

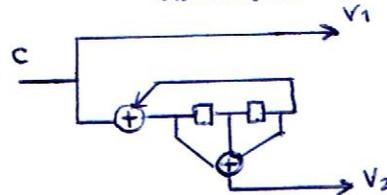
Eliminar algunos bits de v
 ↳ mejora la tasa binaria
 ↳ empeora prestaciones
 se puede alternar según haya mayor o menor E_b/N_0
 ej: eliminar 3^{er} bit de cada 4



• CC sistemáticos: **CCS**
 Los primeros k bits de salida son igual a los de entrada
 $G(D) = \begin{bmatrix} I & P(D) \end{bmatrix}$

• CC recursivo: **CCSR** ← recursivo sistemático

ej: CCSR (2,1,2)



$$G(D) = \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix}$$

3. DECODIFICACION BASADA EN ENREJADO

$$P(v|c) = P(c|v) \cdot \frac{P(v)}{P(c)}$$

↑ lo que maximiza MAP ↑ lo que maximiza ML ⇒ iguales si símbolos equiprobables

Algoritmo de Viterbi

Mantener sólo el camino con métrica máxima de entre los que entran a un estado (superviviente)

- si T es muy grande, tomar decisión cada D pasos del enrejado (recomendado $D = 6 \cdot (m+1)$)

maximizar $P(c|v)$ ↔ canal gaussiano ↔ Minimizar distancia euclídea entre c y z

↕

Maximizar la métrica del camino

$$\mu_t = \sum_{i=1}^t b_i = \mu_{t-1} + b_t$$

↑ acumular

métrica de cada rama en el instante t :

$$b_t = \sum_{i=0}^{n-1} r_{ti} x_{ti}$$

nota: bit 1 ⇒ +1, bit 0 ⇒ -1

↑ para cada transición dentro del enrejado



Algoritmo MAP → BCJR

Para cada bit decodificado se calcula el LLR (log-likelihood ratio)

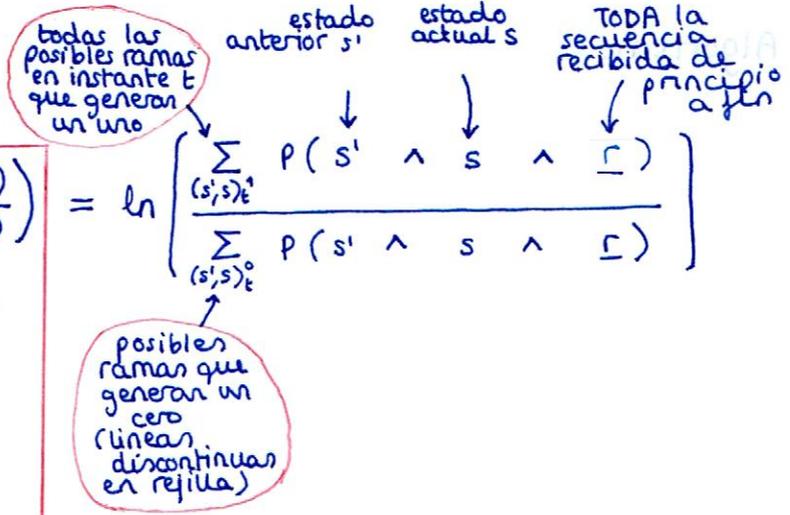
LLR

$$L(c_t | \underline{r}) = \ln \left(\frac{P(c_t=1 | \underline{r})}{P(c_t=0 | \underline{r})} \right) = \ln \left(\frac{\sum_{(s',s)_t^1} P(s' \wedge s \wedge \underline{r})}{\sum_{(s',s)_t^0} P(s' \wedge s \wedge \underline{r})} \right)$$

↑ bit actual ↑ TODA la secuencia recibida $t \in [1, \dots, T]$ ↑ proba conjunta

escojo 0 escojo 1 (hard) salida soft

el bit era seguro un 0 no se el bit era seguro un 1



Pudiendo descomponerse en el tiempo

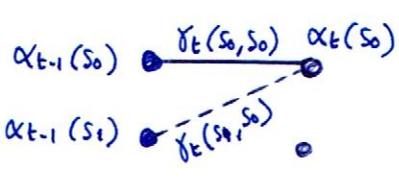
$$P(s' \wedge s \wedge \underline{r}) = P(s' \wedge \underline{r}_{j < t}) \cdot P(\{r_t \wedge s\} | s') \cdot P(\underline{r}_{j > t} | s)$$

transiciones pasadas (sólo dependen de s') transición actual ($s' \rightarrow s$) transiciones futuras (sólo dependen de s)

$$= \alpha_{t-1}(s') \cdot \gamma_t(s', s) \cdot \beta_t(s)$$

② α : FORWARD

El α de cada estado se calcula con los α de los estados que llevan a el

$$\alpha_t(s_0) = \gamma_t(s_0, s_0) \cdot \alpha_{t-1}(s_0) + \gamma_t(s_1, s_0) \cdot \alpha_{t-1}(s_1) + \dots$$


① γ : se calcular inicialmente para CADA rama

$$P(r_t | \{s' \wedge s\}) = \frac{1}{(\sigma \sqrt{2\pi})^n} e^{-\frac{1}{2\sigma^2} \|r_t - x\|^2}$$

$$= \frac{1}{(\sigma \sqrt{2\pi})^n} e^{-\frac{1}{2\sigma^2} (n + \|r_t\|^2 - 2r_t x)}$$

simbolos complejos

$$\gamma_t(s', s) = P(\{r_t \wedge s\} | s') = \frac{P(s | s') \cdot P(r_t | \{s' \wedge s\})}{P(c_t)}$$

prob. del bit que causa transición de s' a s

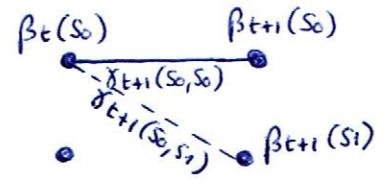
Las constantes se cancelan al calcular los LLR, por tanto

$$\gamma_t(s', s) = P(c_t) \cdot e^{\frac{r_t \cdot x_t}{\sigma^2}}$$

información a priori (tb se cancela si no se tiene) lo que usaba viterbi necesito info del canal (se cancela si no hay info a priori)

③ β : BACKWARD

El β de cada estado se calcula con los β de los estados que salen de el

$$\beta_t(s_0) = \gamma_{t+1}(s_0, s_0) \cdot \beta_{t+1}(s_0) + \gamma_{t+1}(s_0, s_1) \cdot \beta_{t+1}(s_1) + \dots$$


④ calcular los LLR

Modificaciones

Algoritmo Max-Log-MAP \leftarrow subóptima
 \leftarrow mucho más simple

Trabajar con el logaritmo de las métricas:

$$\left. \begin{aligned} \bar{\gamma}_t(s',s) &= \ln(P(c_t)) + \frac{1}{\sigma^2} r_t \cdot x_t \\ \bar{\alpha}_t(s) &= \ln\left(\sum_{s'} e^{\bar{\alpha}_{t-1}(s') + \bar{\gamma}_t(s',s)}\right) \quad \text{c.I. } \alpha_0(s_0) = 0 \\ &\quad \alpha_0(s_i) = -\infty \\ \bar{\beta}_{t-1}(s') &= \ln\left(\sum_s e^{\bar{\beta}_t(s) + \bar{\gamma}_t(s',s)}\right) \quad \text{c.F. } \beta_T(s_0) = 0 \\ &\quad \beta_T(s_i) = -\infty \end{aligned} \right\} L(c_t | \Omega) = \ln \left[\frac{\sum_{(s',s)_t^1} e^{\bar{\alpha}_{t-1}(s') + \bar{\gamma}_t(s',s) + \bar{\beta}_t(s)}}{\sum_{(s',s)_t^0} e^{\bar{\alpha}_{t-1}(s') + \bar{\gamma}_t(s',s) + \bar{\beta}_t(s)}} \right]$$

se introduce la aproximación: $\ln(e^{\delta_1} + e^{\delta_2} + \dots + e^{\delta_n}) \approx \max(\delta_i)$

Entonces queda:

$$L(c_t | \Omega) \approx \max_{(s',s)_t^1} [\bar{\alpha}_{t-1}(s') + \bar{\gamma}_t(s',s) + \bar{\beta}_t(s)] - \max_{(s',s)_t^0} [\bar{\alpha}_{t-1}(s') + \bar{\gamma}_t(s',s) + \bar{\beta}_t(s)]$$

\leftarrow Nos quedamos sólo con la transición máxima (igual que Viterbi)

$$\begin{aligned} \bar{\alpha}_t(s) &\approx \max_{s'} [\bar{\alpha}_{t-1}(s') + \bar{\gamma}_t(s',s)] & \bar{\gamma}_t(s',s) &= \underbrace{\ln(P(c_t))}_{\text{info a priori}} + \underbrace{\frac{1}{\sigma^2} r_t \cdot x_t}_{\text{info del canal}} \\ \bar{\beta}_{t-1}(s) &\approx \max_s [\bar{\gamma}_t(s',s) + \bar{\beta}_t(s)] & &= c_t' \frac{L_a(c_t)}{2} + \frac{L_c}{2} \sum_{i=0}^{n-1} r_{ti} x_{ti} \end{aligned}$$

- No es más que un doble Viterbi
- Misma salida HARD que Viterbi pero incluye salida SOFT

Algoritmo Log-MAP \leftarrow óptimo (resultado idéntico al ideal)
 \leftarrow menor coste computacional

Usar la igualdad: $\ln(e^{\delta_1} + e^{\delta_2}) = \underbrace{\max(\delta_1, \delta_2)}_{\max^*(\delta_1, \delta_2)} + \ln(1 + e^{-|\delta_2 - \delta_1|})$ \rightarrow La corrección es TABULABLE

Recursivamente:

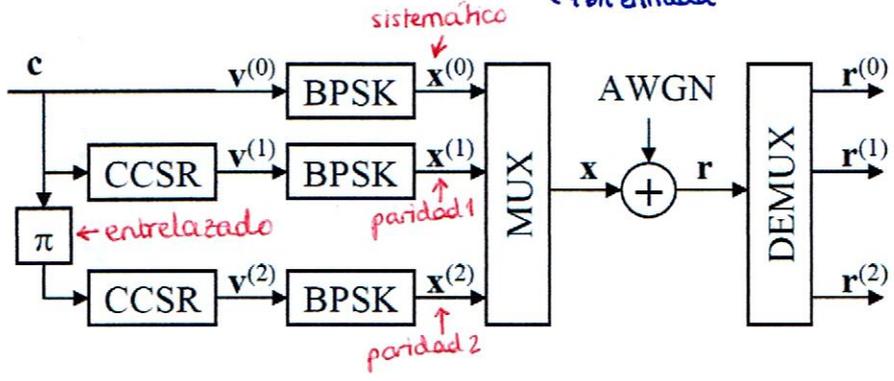
$$\begin{aligned} &\ln(e^{\delta_1} + e^{\delta_2} + e^{\delta_3}) \\ &= \max^*(\delta_1, \max^*(\delta_2, \delta_3)) \\ &= \max(\delta_1, \max(\delta_2, \delta_3) + \ln(1 + e^{-|\delta_2 - \delta_3|})) \\ &\quad + \ln(1 + e^{-|\max(\delta_2, \delta_3) + \ln(1 + e^{-|\delta_2 - \delta_3|}) - \delta_1|}) \end{aligned}$$

Shannon dijo que se necesitan bloques grandes y codif. aleatoria (concatenar Reed solomon + convolucional) hasta los años 90

4. TURBOCÓDIGOS Y DECODIFICACIÓN ITERATIVA

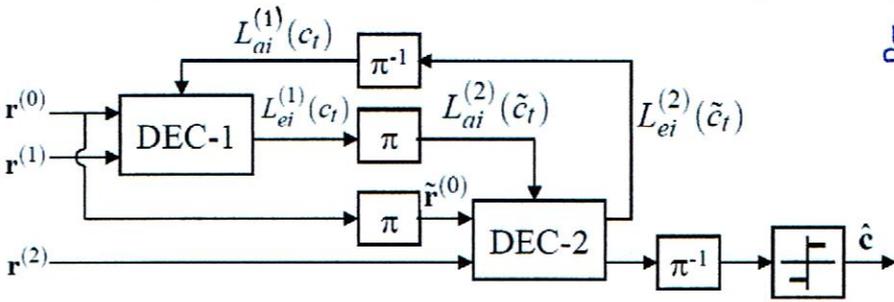
- Usar 2 CC en paralelo con mismos datos de entrada pero en distinto orden (entrelazado)
 - Decodificación ML inabordable
- se suelen usar 2 CCSR (n, 1, m)
 - 1 bit entrada

si n=2, codificador total de tasa 1/3 (perforando par/impar da 1/2)



- Terminación de C.C.
- Añadir bits de cierre \Rightarrow Tx bits y paridad
 - Solución óptima: TAIL BITTING (terminación circular)
 - ahorra bits de cierre
 - protección uniforme de todos los bits

- Decodificación \rightarrow óptima no factible
- \rightarrow iterativa con dos decodificadores BCJR SISO (soft input soft output)
 - se pasan info extrínseca soft entre ellos
 - lo que añaden al LLR sobre lo ya conocido como info a priori



- Cada decodificador SISO usa la info a priori que le pasa el otro para calcular la métrica de rama:

$$\bar{\gamma}_t(s', s) = \ln \left[P(c_t) \cdot e^{\frac{1}{\sigma^2} \sum r_{ti} \cdot x_{ti}} \right] = \text{dem en trasp}$$

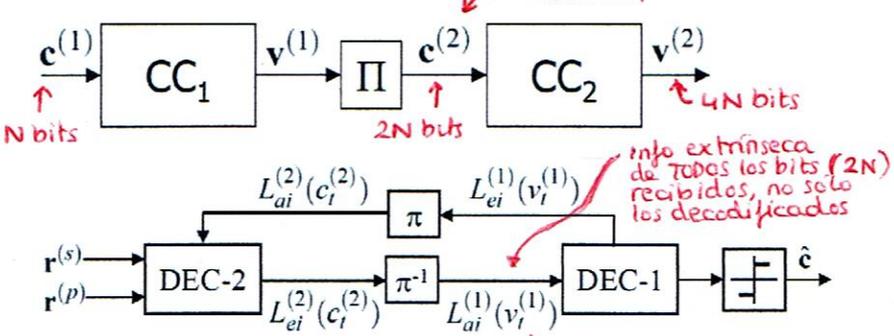
$$c_t \cdot \frac{L_a(c_t)}{2} + \frac{L_c}{2} r_t \cdot x_t = \bar{\gamma}_t(s', s)$$

$\underbrace{\hspace{10em}}_{\text{info a priori}} \quad \underbrace{\hspace{10em}}_{\text{info del canal}} \quad L_c = \frac{2}{\sigma^2} = 4 \frac{E_s}{N_0}$

- Y descompone en partes los LRR que calcula de cada bit para pasarle al otro sólo la parte extrínseca

$$L(c_t | r_t) = \underbrace{L_a(c_t)}_{\text{a priori}} + \underbrace{L_c \cdot r_t}_{\text{canal primer bit = sistemático}} + \underbrace{L_e(c_t)}_{\text{extrínseca despear}}$$

TURBOCÓDIGOS SERIE



DEC-1 no recibe nada del canal! Trabaja sólo con la info a priori que le pasa el otro

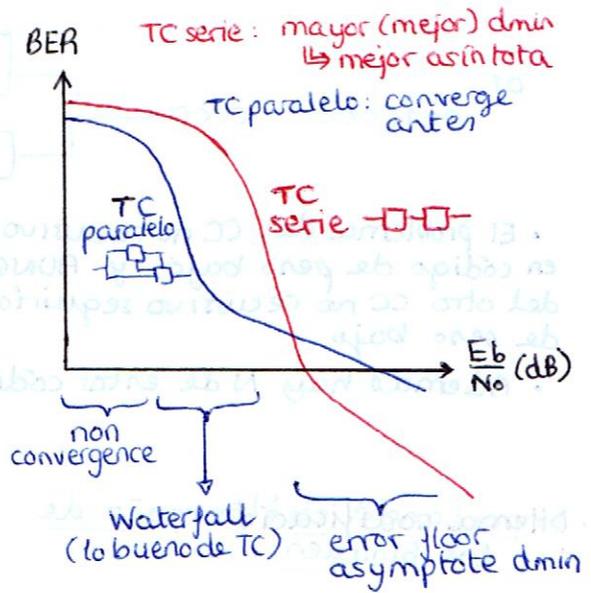
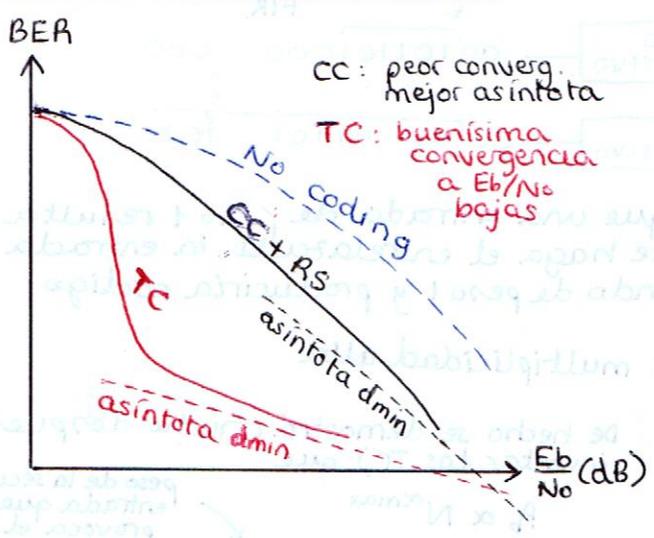
Serie \rightarrow

- Algoritmo decodificación iterativa:
- Inicializar $L_{ai}^{(1)} = 0$
 - Iteraciones $i=1, \dots, I$
 - DEC-2:
 - Obtener información a priori: $L_{ai}^{(2)} = L_{ei}^{(1)}$
 - Calcular los LLR para $t=1, \dots, 2N$
 - Obtener la información extrínseca para $t=1, \dots, 2N$
 - no se sustraen la información del canal porque DEC-1 no la tiene
 - DEC-1:
 - Obtener información a priori: $L_{ai}^{(1)} = L_{ei}^{(2)}$
 - Calcular los LLR de todos los bits codificados para $t=1, \dots, N$
 - Obtener la información extrínseca para $t=1, \dots, N$
 - Se emplea la siguiente métrica de soft
 - En la última iteración se obtienen los bits decodificados

no hace falta restarle $L_c \cdot r_t$

U.4 Prestaciones de los TC

se comparan en AWGN (modela bien al espacio)



Comportamiento asintotico de TC y CC

$$P_b \approx M_{min} \cdot \bar{w}_{min} \cdot \frac{1}{N} Q\left(\sqrt{2 \cdot d_{min} \cdot R \cdot \frac{E_b}{N_0}}\right)$$

↑
multiplicidad de códigos a la distancia libre (cuántos códigos tienen d_{min})

↑
tamaño del bloque (nº de bits entre bits de terminación)

↑
distancia libre (minima distancia entre códigos posibles) = nº de unos del código con menor nº de unos

cuanto menor sea, más fácil es confundir los códigos

R: tasa
 \bar{w}_{min} : peso promedio de secuencias de info que dan lugar a códigos con peso d_{min}

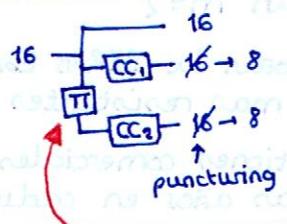
Zona waterfall
TC logran buenos resultados a bajas E_b/N_0 gracias a su baja multiplicidad efectiva $\equiv \frac{M_{min}}{N}$

Error floor: - pendiente mayor a mayores d_{min}
CC(2,1,14) el mayor CC jamás usado tenía $d_{min}=18$ (muy buena!!) PERO tenía muchos códigos a d_{min} (más a E_b/N_0 baja)
- puede bajarse el nivel del error floor

Spectral Thinning

ej:
100001 → CC(2,1,4) → código con $d=d_{min}=6$
↑
este código y TODAS sus N-5 versiones retrasadas generan código con d_{min}
↓
Multiplicidad alta para pesos bajos

Si se usa el mismo CC en un TC:



• Gracias al entrelazado aleatorio, aunque la entrada a C1 de lugar a un código de peso bajo, la entrada a C2 YA NO tiene porque dar lugar a un código de peso bajo

El entrelazador empareja la mayor parte de los códigos de peso bajo del primer codificador con códigos de peso alto en el segundo
Para N grande el efecto es mayor

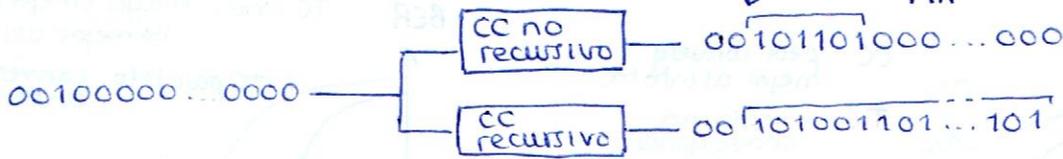
TABLE 16.1: Weight spectra of two (32, 12) codes.

(a) Terminated convolutional		(b) Parallel concatenated	
Weight	Multiplicity	Weight	Multiplicity
0	1	0	1
1	0	1	0
2	0	2	0
3	0	3	0
4	0	4	0
5	0	5	1
6	11	6	4
7	12	7	8
8	23	8	16
9	38	9	30
10	61	10	73
11	126	11	144
12	200	12	210
13	332	13	308
14	425	14	404

→ mala suerte ha bajado la d_{min}
→ pero se ha reducido la multiplicidad de pesos bajos

Dilema codificación:
Convergencia vs. d_{min}
Los TC han resuelto el problema de la convergencia

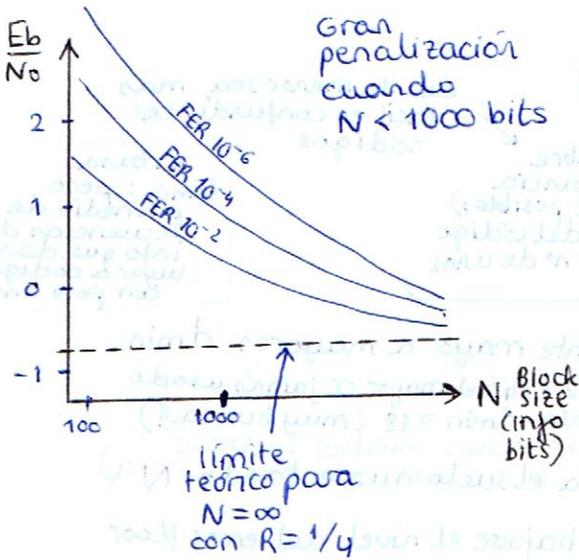
Importancia de usar CC recursivos



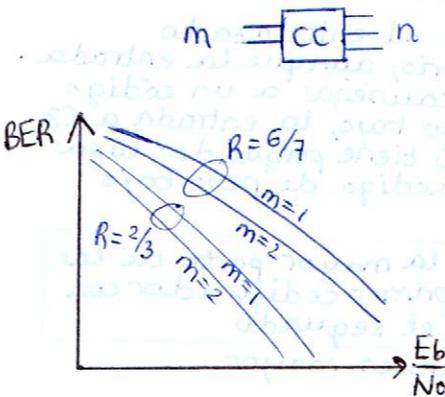
- El problema del CC no recursivo es que una entrada de peso 1 resulta en código de peso bajo, y AUNQUE se haga el entrelazado la entrada del otro CC no recursivo seguiría siendo de peso 1 y produciría código de peso bajo
- Además hay N de estos códigos: multiplicidad alta

De hecho se demostró (mucho después de inventar los TC) que $P_b \propto N^{\alpha_{max}}$
 $\alpha_{max} = 1 - w_{min}$
 peso de la secuencia de entrada que provoca el código de distancia mínima
 sólo hay ganancia si $w_{min} > 1$

Importancia del tamaño de los bloques



TC con codificadores no binarios $m \neq 1$



- Todos los estándares que están surgiendo usan $m=2$
- Mejoran la BER vs E_b/N_0
- Son más resistentes a la aproximación max-log-MAP
- cuestiones comerciales: la patente tardará muchos más años en caducar

Prestaciones estructura serie

- Las entradas al CC interno son secuencias cuyo w_{min} es igual al d_{min} del CC externo
- ⇒ d_{min} mayores
- ⇒ se reduce la multiplicidad de pesos bajos

Análisis por cotas:

$P_b \propto N^{\alpha_{max}} \rightarrow \alpha_{max} = - \left\lfloor \frac{d_{min} + 1}{2} \right\rfloor$

mejores resultados en zona de BER baja (suelo de error)

serie: mejor suelo de error
 paralelo: converge antes

Diagramas EXIT

• Curva que relaciona info a priori de un bit $I_a \equiv I_a(C_t)$ con info extrínseca obtenida $I_e \equiv I_e(C_t)$

la curva es única para un decodificador BCJR dado

Al iterar en un TC interesa $I_e > I_a$ (significa que ha mejorado la información sobre el bit)

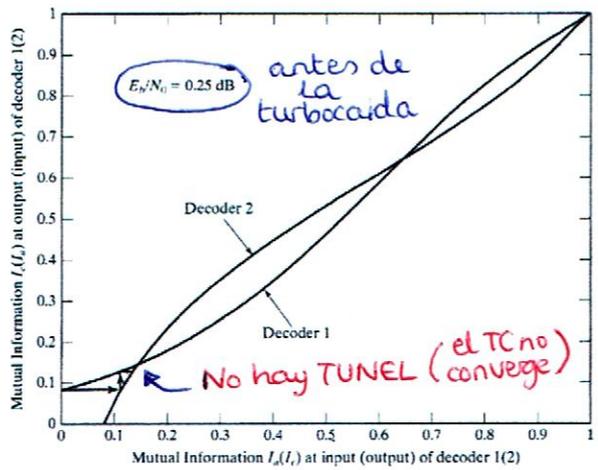
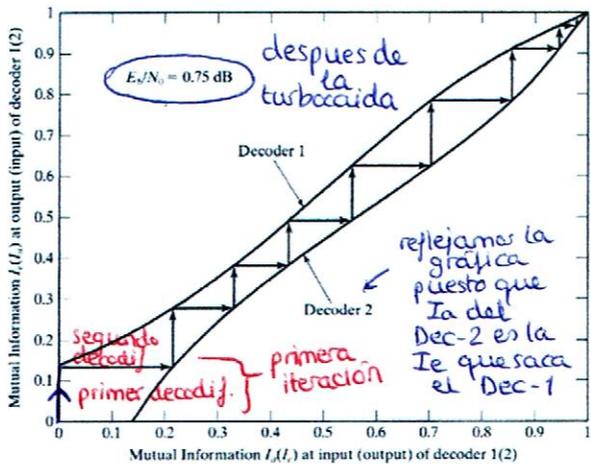
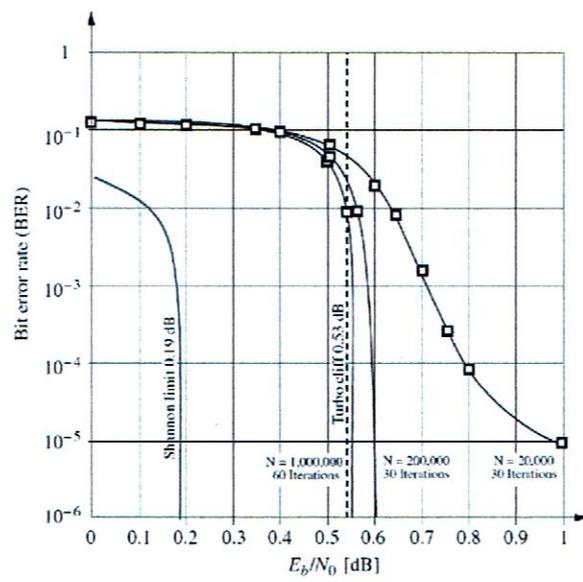
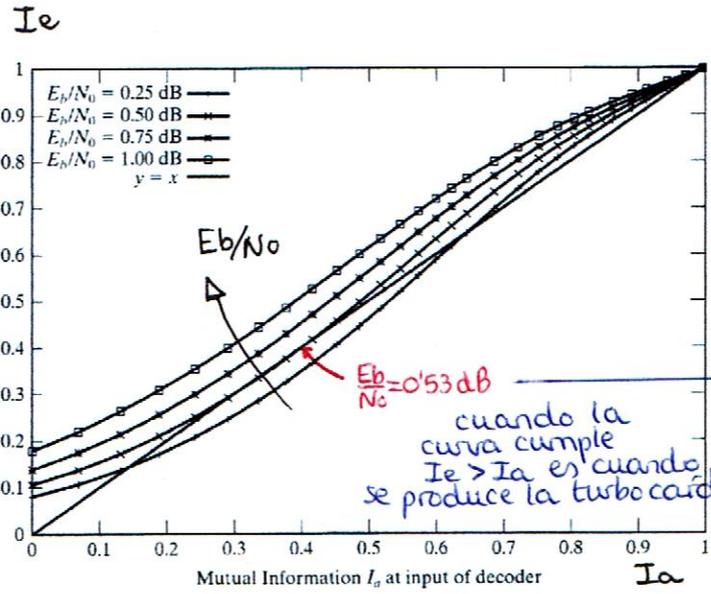
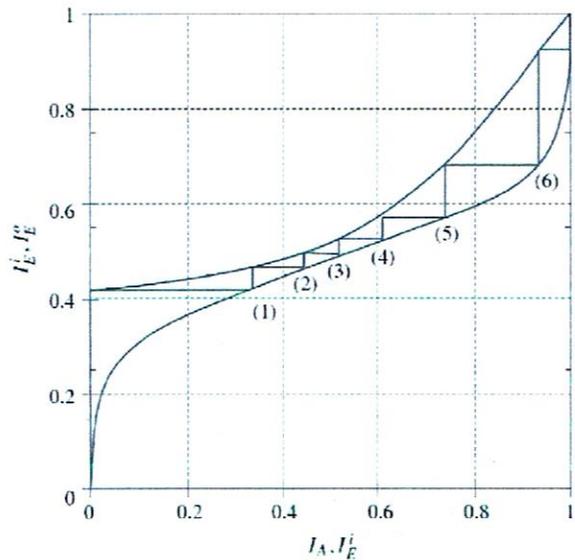


Diagrama EXIT para TC serie:



Curvas no simétricas ya que el CC interno y el externo no son el mismo

4.5 Aplicaciones de los turbocódigos

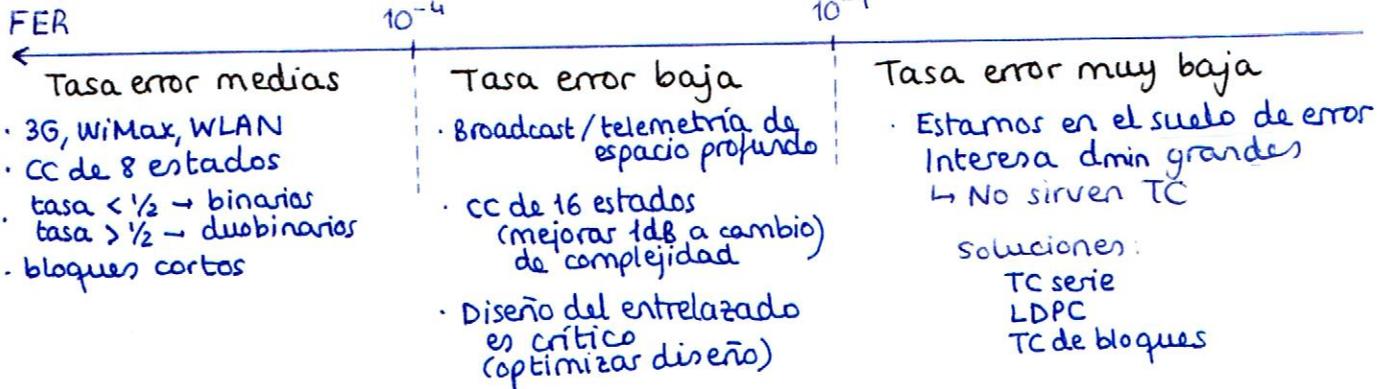
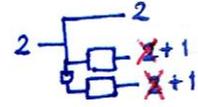
• TC usados en numerosos estándares

• Codificadores habituales:

• CCSR binarios de tasa $\frac{1}{2}$ → TC de tasa $\frac{1}{3}$ perforado → tasa $\frac{1}{2}$ o mayor de 8 o 16 estados

• CCSR no-binarios de tasa $\frac{2}{3}$ → TC de tasa global $\frac{1}{2}$

- mejor convergencia
- menos sensible a perforado y a decodificación subóptima
- facilita diseño entrelazado
- decodificación más compleja



4.6 Aspectos Prácticos

• Tamaño de bloque:

- Debe ser grande para mejorar las prestaciones, en muchas aplicaciones es necesario que sea pequeño para tener una latencia baja
- Hay estándares donde se usan varios tamaños: el problema surge en el diseño del entrelazado

• Cuantificación:

- Los TC funcionan bien con pocos bits de precisión interna pues tienen buenas prestaciones a SNR bajas: son resistentes a trabajar con ruido de cuantificación
- Distinguir: LLR entrada (3 a 6), LLR extrínseco (entrada+1), recursiones (7 a 10)

• Criterios de parada:

- Habitualmente se usa número fijo, reducirlo mejora consumo
- Criterios: comprobación mediante CRC, signos LLR en entrada y salida iguales (en 2 iteraciones), LLR mínimo supera umbral

• Enventanado:

- Método reducción de latencia: se divide el bloque en varios subbloques para no esperar hasta final para la recursión inversa
- La inicialización de los α se basa en la info. del bloque anterior
- La inicialización de los β necesita empezar con distribución uniforme D bits después del fin de ventana (similar a profundidad de decisión en Viterbi)

• Paralelización:

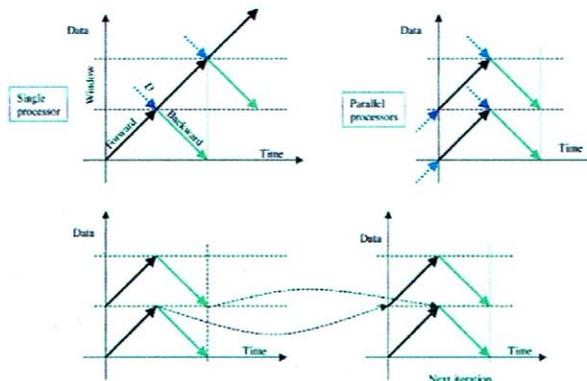
- Es necesaria para lograr velocidades grandes. El problema es la naturaleza iterativa de los TC
- Soluciones básicas:
 - Aumentar el número de decodificadores para trabajar con bloques distintos al mismo tiempo
 - Utilizar un array lineal de $2 \cdot N_{dec}$ decodificadores SISO, cuando uno termina una iteración pasa el resultado al siguiente. Problemas: demasiada latencia y uso de muchos recursos (replicar memoria)
- Uso de enventanado: dividir bloque en P ventanas
 - P decod. SISO trabajan en paralelo con todo el bloque
 - La inicialización se puede hacer trabajando con D bits antes, o bien con el resultado de la iteración anterior, si la ventana es larga a penas hay degradación de las prestaciones
 - Pueden aparecer problemas de colisión en la segunda parte de la iteración (cuando se leen las memorias entrelazadas). Soluciones:
 - Hay entrelazadores diseñados para evitar colisión al paralelizar la decodificación por ventanas: DVB-RCS y WiMax
 - Diseño de la partición de las memorias y su acceso mediante buffers para evitar las colisiones

• Tipo de decodificador:

- La solución más sencilla es max-Log-Map, problema es que degrada las prestaciones (0,5 dB / 0,1 dB). Soluciones:
 - Log-Map con cuantificación de la corrección
 - Max-Log-Map con reescalado (reducción de los valores extrínsecos sobrestimados (degradación de sólo 0,2 / 0,05 dB)

• Perforado:

- En principio sólo se perforan los bits de paridad: mantiene las prestaciones en la zona de convergencia (SNR bajas) pero empeora el suelo de error (SNR altas)
- Si se quiere mantener buenas prestaciones en SNR altas se pueden perforar los bits sistemáticos



• Diseño del entrelazado:

- Buenas prestaciones: pseudoaleatorio evite d_{min} bajas
- Permitan decodificación paralela sin colisión
- Cálculo de posición en tiempo de ejecución
- Faciliten trabajar con bloques de diferentes tamaños

i.e. si el \oplus del resto de bits conectados es 0, entonces el tuyo también debería ser 0 para que el \oplus total de bits conectados fuese 0.

y si el \oplus del resto de bits conectados es 1, entonces el tuyo también debería ser 1 para que el \oplus total de bits conectados fuese 0

Decodificación por paso de mensaje: HARD

- Se transmite $\mathbf{v} = [1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1]$ y canal cambia $v_2 = 1$

- Algoritmo decodificación:

- Paso 1: los nodos bit envían a los nodos comprobación el bit que creen correcto (en primera iteración el bit recibido)
- Paso 2: los nodos comprobación devuelven el resultado de aplicar la ecuación de paridad a cada bit recibido usando el resto de bits conectados. Ej: $c_1 \rightarrow v_2; v_4 \oplus v_5 \oplus v_6 = 0, c_2 \rightarrow v_2; v_1 \oplus v_5 \oplus v_6 = 0$
 - Si se cumplen todas las ecuaciones de chequeo la decodificación termina. El bit de salida de c_m y el existente en v_n deben ser iguales para que su suma de 0
- Paso 3: los nodos bit toman decisión con la información recibida, usando por ejemplo votación por mayoría.
 - Ej: v_2 tiene un 1 y recibe 0 de c_1 y 0 de c_2 , por lo que cambia a 0
- Paso 4: ir a paso 1

Decodificación por paso de mensaje: SOFT

- En el receptor se tiene la señal $\mathbf{r} = [r_1, r_2, \dots, r_n]$ tras pasar por el canal AWGN la palabra código $\mathbf{v} = [v_1, v_2, \dots, v_n]$:

$$\mathbf{r} = 2\mathbf{v} - 1 + \mathbf{w}$$

- El detector que minimiza la prob. de error para el bit n -ésimo calcula el LLR *a posteriori*:

$$L(v_n | \mathbf{r}) = \lambda_n = \frac{\log P(v_n = 1 | \mathbf{r})}{\log P(v_n = 0 | \mathbf{r})}$$

- Descomponiendo $\mathbf{r} = [r_n, \{r_{i \neq n}\}]$ y aplicando Bayes queda:

$$\lambda_n = \log \frac{P(r_n | v_n = 1)}{P(r_n | v_n = 0)} + \log \frac{P(v_n = 1 | \{r_{i \neq n}\})}{P(v_n = 0 | \{r_{i \neq n}\})}$$

- Teniendo en cuenta que el canal es AWGN el LLR queda:

$$\lambda_n = \frac{2}{\sigma^2} r_n + \log \frac{P(v_n = 1 | \{r_{i \neq n}\})}{P(v_n = 0 | \{r_{i \neq n}\})}$$

• El primer término es la información intrínseca (contribución del canal) y el segundo es la información extrínseca (generada por el resto de bits recibidos)

- El cálculo de la información extrínseca se hace de manera similar a la decodificación HARD utilizando el paso de mensajes: es el resultado obtenido por los nodos de comprobación conectados al bit n

- Regla de la tanh

• Se tiene un conjunto de bits $\{v_1, v_2, \dots, v_p\}$ independientes y con probabilidades a priori definidas mediante LLR:

$$\lambda_i = \log \frac{P(v_i = 1)}{P(v_i = 0)}$$

• Dada la ecuación de paridad

$$\psi = \sum_{i=1}^p v_i$$

• Se puede demostrar que el LLR de la ecuación de paridad λ_ψ está relacionado con los LLR a priori de bit según:

$$\lambda_\psi = -2 \tanh^{-1} \left(\prod_{i=1}^p \tanh \left(\frac{-\lambda_i}{2} \right) \right)$$

Algoritmo de decodificación SOFT:

• Definimos los conjuntos de índices: $\mathcal{M}_n = \{m : H_{mn} = 1\}$

$\mathcal{N}_m = \{n : H_{mn} = 1\}$

• Definimos $u_{mn}^{(l)}$ como el mensaje entre nodo comprobación m y el nodo de bit n en la iteración l

• Inicialización para todos los nodos: $u_{mn}^{(0)} = 0 \quad \lambda_n^{(0)} = \frac{2}{\sigma^2} r_n$

- Iteración:

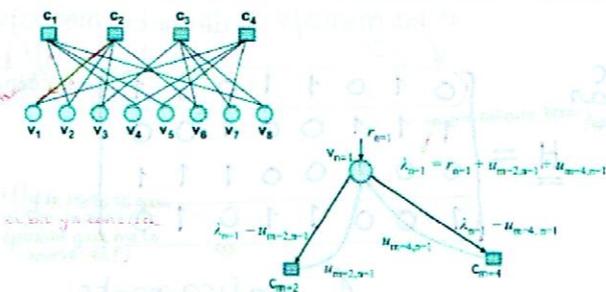
• Nodos comprobación: para $m \in \{1, \dots, M\} \quad n \in \mathcal{N}_m$

$$u_{mn}^{(l)} = -2 \tanh^{-1} \left(\prod_{i \in \mathcal{N}_m, i \neq n} \tanh \left(\frac{\lambda_i^{(l-1)} - u_{mi}^{(l-1)}}{2} \right) \right)$$

• Nodos bit: para $n \in \{1, \dots, N\}$

$$\lambda_n^{(l)} = \frac{2}{\sigma^2} r_n + \sum_{m \in \mathcal{M}_n} u_{mn}^{(l)}$$

- Interpretación gráfica de la decodificación SOFT:

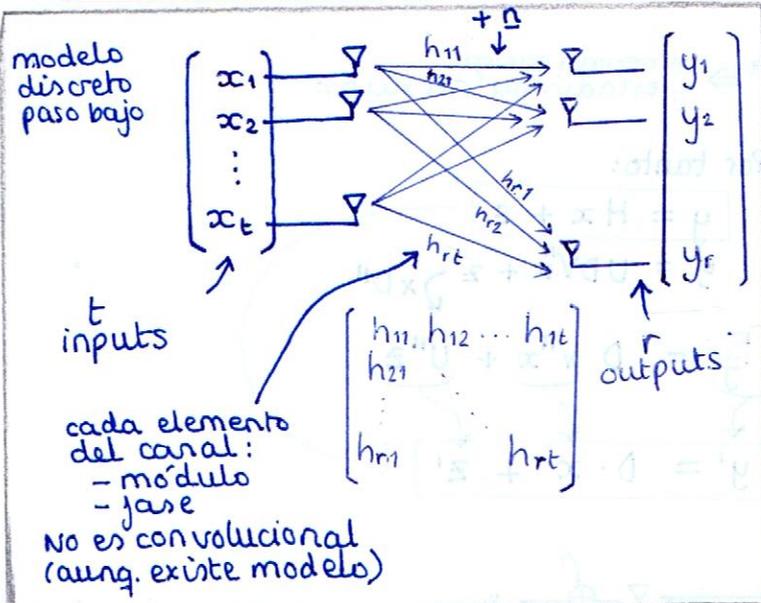


$$u_{m-2,n-1} = -2 \tanh^{-1} \left(\tanh(\lambda_2 - u_{2,2}) \cdot \tanh(\lambda_3 - u_{2,3}) \cdot \tanh(\lambda_6 - u_{2,6}) \right)$$

PARTE 2. MIMO

Multiple Input
Multiple output

1. MODELOS DE CANAL MIMO



$$\vec{y}_{rx1} = \underline{H}_{rxt} \cdot \vec{x}_{tx1} + \vec{n}_{rx1}$$

si considero varios tiempos de simbolo: (trama N simbolos)

$$\underline{Y}_{rxN} = \underline{H}_{rxt} \underline{X}_{txN} + \underline{Z}_{rxN}$$

Slow fading: mantengo misma matriz H durante toda la trama

Fast fading: cada vez se selecciona una realizacion de H (i.e. es una V.A)

Ventajas intuitivas inmediatas

- t transmisores: MULTIPLEXADO: ganancia en regimen binario
- r receptores: DIVERSIDAD: ganancia en potencia
 - si se estima el canal: beamforming
 - CSI: channel state information

Detección maxima verosimilitud \iff minimizar $\|Y - HX\|^2$ (escoger la mejor X de entre las posibles)

2. CAPACIDAD CANALES MIMO

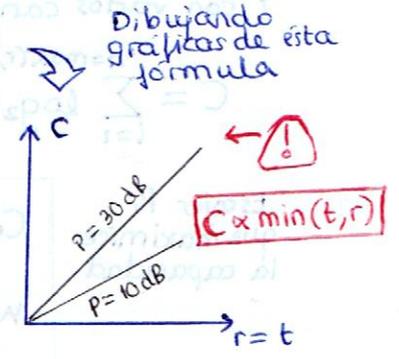
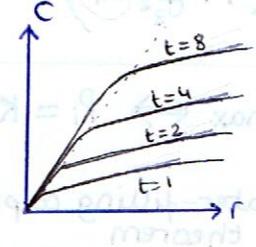
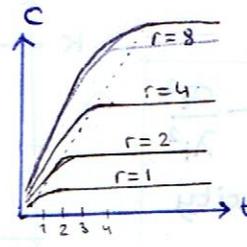
canal SISO: $y = \alpha \cdot x + z \rightarrow C = \log_2(1 + \frac{P}{\sigma_z^2} |\alpha|^2)$

canal MIMO $\begin{bmatrix} y \\ \vdots \\ y_r \end{bmatrix} = \underline{H}_{rxt} \begin{bmatrix} x \\ \vdots \\ x_t \end{bmatrix} + \begin{bmatrix} z \\ \vdots \\ z_r \end{bmatrix} \rightarrow C = \log_2 \left[\det \left[\underline{I}_{rxr} + \frac{1}{\sigma_z^2} \underline{H} \cdot \underline{Q} \cdot \underline{H}^H \right] \right]$
covariante de señal tx $\underline{Q} = E\{\vec{x} \cdot \vec{x}^T\}$

ejemplo:
 sencillo y justo: $\underline{Q} = \frac{P}{t} \underline{I}$ (se reparte pot. entre antenas)
 canales incorreladas: $\underline{H} \cdot \underline{H}^H = \underline{I}$
 entonces $\hookrightarrow C = r \log_2(1 + \frac{P/t}{\sigma_z^2})$ si $r \geq t$
 Capacidad aumenta linealmente con el numero de antenas

El canal varia con el tiempo (H es V.A)
 Ergodico: podemos estimar capacidad con la media estadistica

$$C = E \left\{ \log_2 \left[\det \left[\underline{I} + \frac{1}{\sigma_z^2} \underline{H} \underline{Q} \underline{H}^H \right] \right] \right\}$$

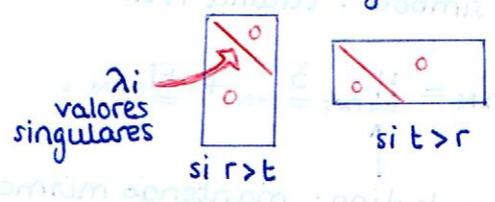


Interpretación de MIMO usando descomposición SVD ← truco matemático MUY revelador

$$H_{r \times t} = U_{r \times r} D_{r \times t} V_{t \times t}^H$$

unitaria: $A^{-1} = (A^t)^*$ ⇒ conservan energía y estadística del ruido

diagonal rectangular



Por tanto:

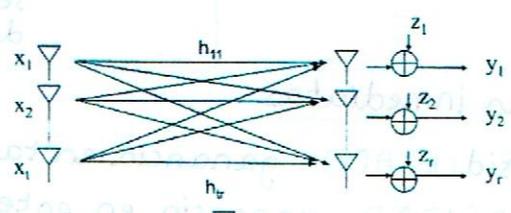
$$y = Hx + z$$

$$y = UDV^H x + z \quad \xrightarrow{\times U^H}$$

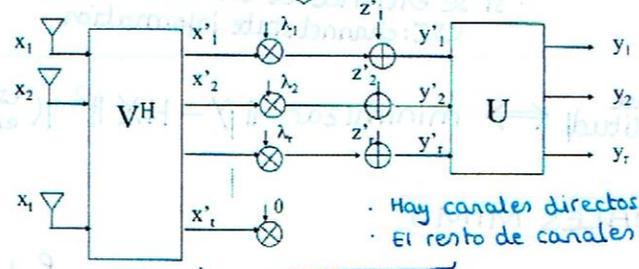
$$U^H y = D V^H x + U^H z$$

$$y' = D \cdot x' + z'$$

La interpretación de la descomposición SVD es la siguiente:



• $t > r = n$



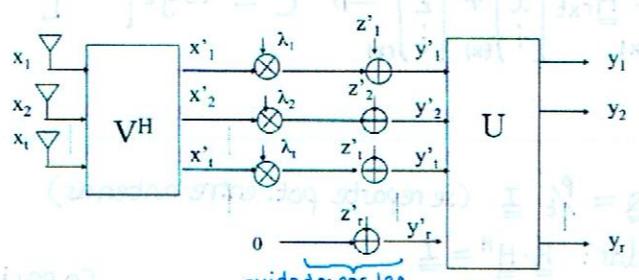
• Hay canales directos hasta 'r'
• El resto de canales son cero

el canal ahora es una matriz diagonal

¿La relación SNR nueva de que dependerá?

- El ruido sigue igual
- Dependerá de las λ que son los valores singulares de H

• $t > r = n$



cuidado: por las entradas a cero entra ruido!

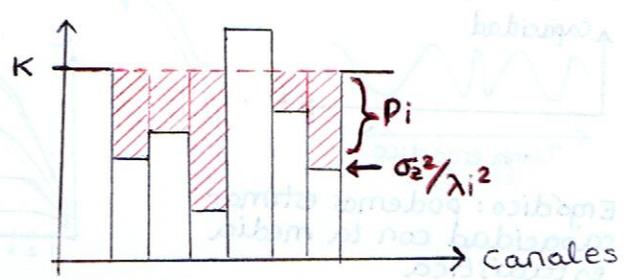
Y con varios canales SISO es trivial calcular la capacidad sumando:

$$C = \sum_{i=1}^{n=\max(r,t)} \log_2 \left(1 + \frac{P_i}{\sigma_z^2} \lambda_i^2 \right)$$

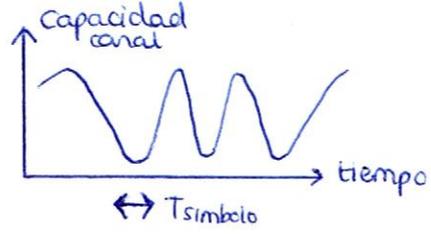
Escojer P_i que maximice la capacidad

$$C_{max} \iff P_i = K - \frac{\sigma_z^2}{\lambda_i^2}$$

Water-filling capacity theorem



Capacidad en canales no ergódicos



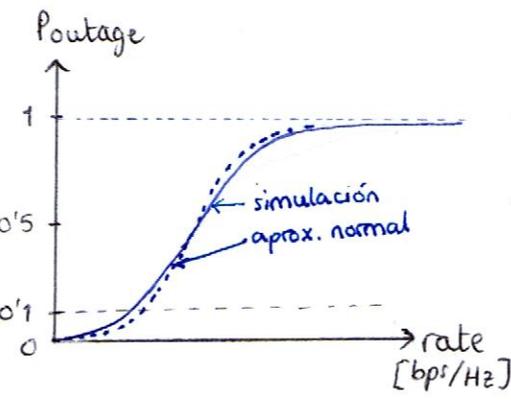
- La capacidad del canal puede ser muy baja durante toda la longitud del "codeword"
- No vale utilizar la 'media' de la capacidad para prever la capacidad
ej: i.e. aunque la media sea 15 bps/Hz, algunos codewords podrán tener sólo 5 bps/Hz

Hay que definir

Outage Probability: probabilidad de que el canal esté por debajo de un umbral de capacidad durante un codeword

$$P_{out} = Pr \{ C(H) < \rho \}$$

siendo $C(H) = \log_2 \left[\det \left[I + \frac{P/t}{\sigma^2} H H^H \right] \right]$ V.A.



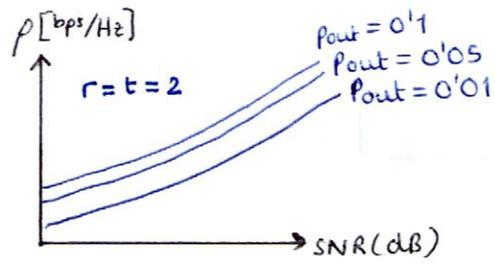
• Puede calcularse de forma exacta por simulación

• Puede estimarse:

suponer $C(H)$ gaussiana $\rightarrow P_{out} = Q \left(\frac{C_{media} - \rho}{\sigma_c} \right)$

↖ bps/Hz mínimo

otras graficas:



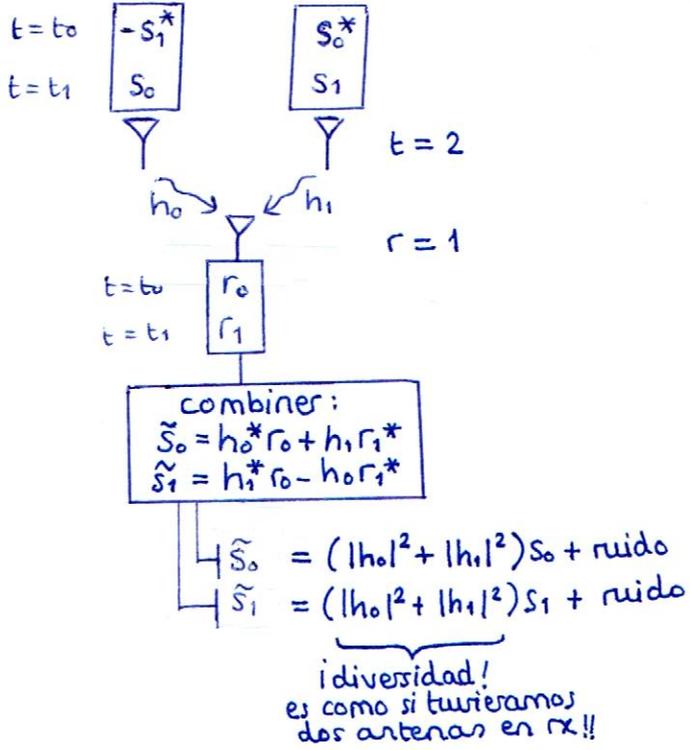
curiosamente las curvas para distintas P_{out} se van juntando según aumenta $r=t$

3. CÓDIGOS ESPACIO-TIEMPO

¿Cómo me apañó para tx símbolos por varias antenas? ¿cada vez un símbolo? ¿todos a la vez? ¿hago sumas y restas?

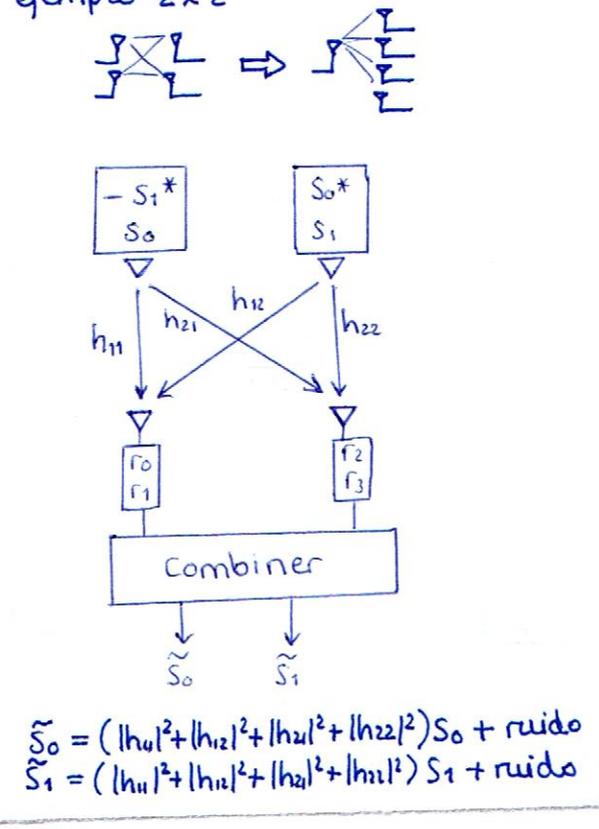
Soluciones ingeniosas: códigos espacio-tiempo $\left\{ \begin{array}{l} \text{garancia por diversidad} \\ \text{garancia de codificación:} \\ \text{utilizar técnicas como en Parte 1} \\ \text{de la asignatura} \end{array} \right.$

→ Códigos de Alamouti → "convierten" el nº de antenas en tx a antenas en rx



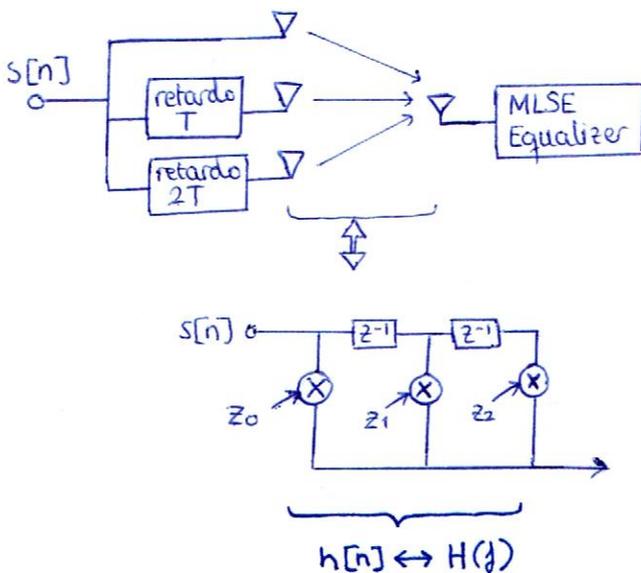
- obtienen ganancia por diversidad pero NO por codificación

ejemplo 2x2



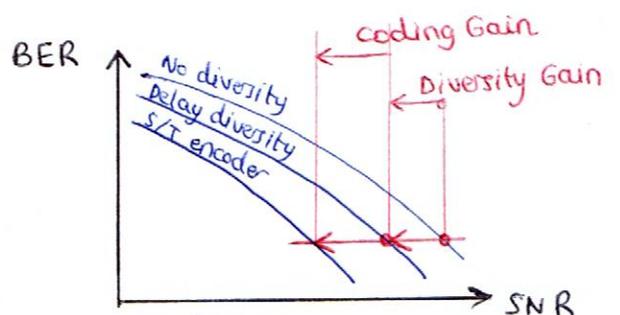
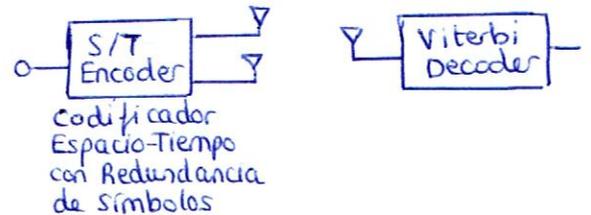
→ Otro ejemplo: convierte canal MIMO no convolucional (fast fading) (módulo y fase)

→ en canal SISO convolucional (Frequency selective fading) (i.e. selectivo en frecuencia con $h(t) \leftrightarrow H(f)$)

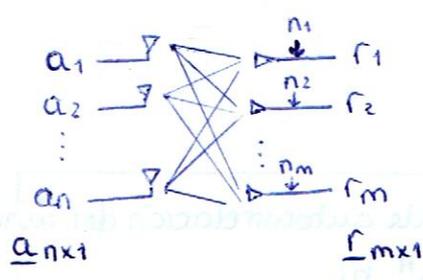


- Delay Diversity

→ Otros: ganancia codificación



4. CANALES MIMO SIN MEMORIA: DETECCIÓN



$$r = H a + n$$

(cambio nomenclatura)

nuido gaussiano blanco, componentes independientes, media nula, idénticamente distribuido, ...

$m \geq n$ y rango completo

vector r es una combinación lineal de las columnas de H

Objetivo: se tiene un alfabeto de posibles a

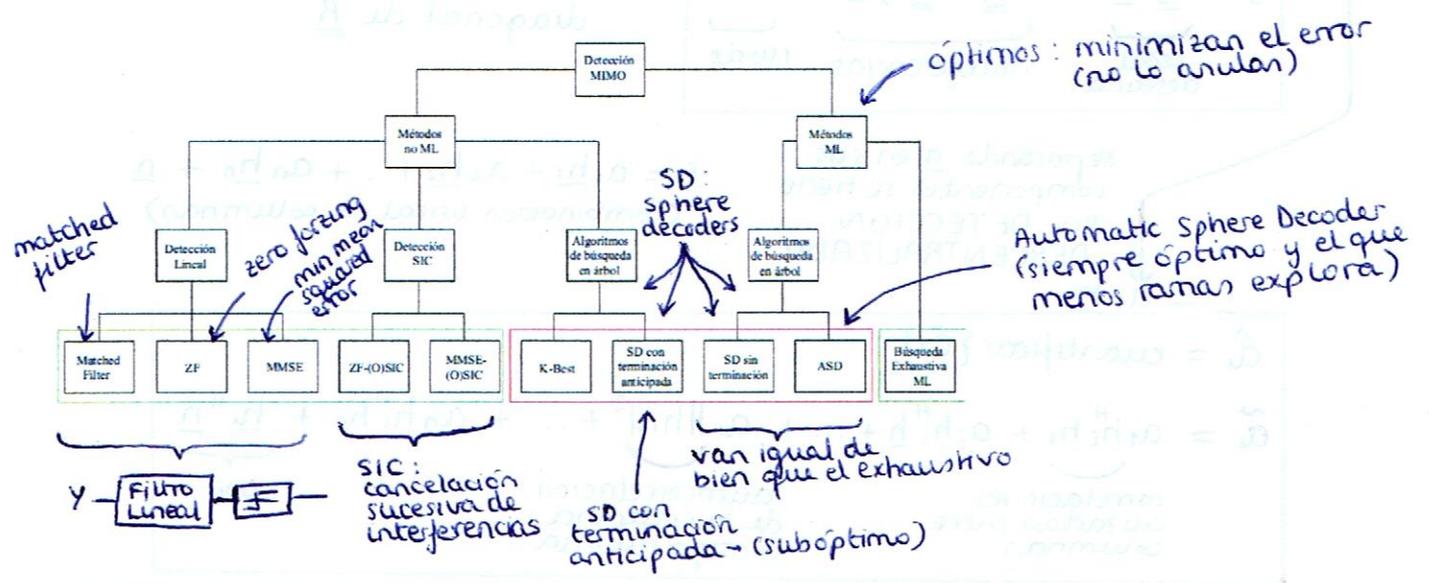
$$\text{Hallar la } a \text{ (del alfabeto) que minimice } \|r - H a\|^2$$

Exhaustivo:

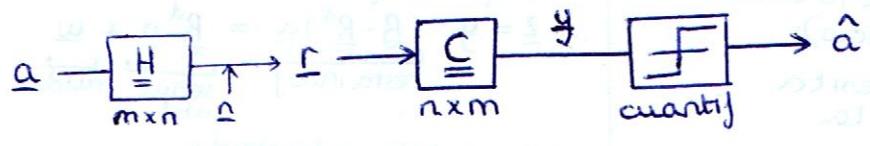
- MIMO 4x4 QPSK: 256 combin.
- MIMO 4x4 16QAM: 65 536
- MIMO 8x8 64QAM: 28 · 10¹²

↓
inviable

CLASIFICACIÓN DE ALGORITMOS



DETECTORES LINEALES



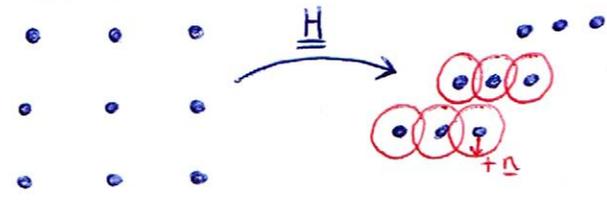
- Centralizado: detecta \hat{a} completo
- No centralizado: Detecta sola a_i

y_i es la suma del producto escalar entre las filas w_i^H de C y las columnas de H por a

$$y_i = a_1 w_i^H h_1 + a_2 w_i^H h_2 + \dots + a_i w_i^H h_i + \dots + a_n w_i^H h_n + w_i^H n$$

Nota: interpretación de $r = H \cdot a + n$

a (posibles)



→ H deforma la constelación y hace que el ruido sea peor en unas direcciones que otras

→ interesa que H deforme poco

Receptor Lineal Filtro Adaptado

$\underline{C} = \underline{H}^H$: filtro adaptado al canal \underline{H}

$y = \underline{H}^H \cdot \underline{H} \underline{r}$

$y = \underbrace{\underline{H}^H \cdot \underline{H}}_{\underline{R}} \underline{a} + \underbrace{\underline{H}^H}_{\underline{w}} \underline{n}$ $\underline{R} = \underline{H}^H \cdot \underline{H}$ matriz de autocorrelación canal
 $\underline{R}_{ij} = \underline{h}_i^H \cdot \underline{h}_j$
 $E\{\underline{w} \cdot \underline{w}^H\} =$ matriz de autocorrelación del ruido filtrado $= N_0 \underline{A}$

$y = \underbrace{\underline{R}^d}_{\text{señal deseada}} \underline{a} + \underbrace{(\underline{R} - \underline{R}^d)}_{\text{interferencias}} \underline{a} + \underbrace{\underline{w}}_{\text{ruido}}$

siendo \underline{R}^d la diagonal de \underline{R}

separando \underline{a} en sus componentes se tiene en DETECCIÓN DESCENTRALIZADA

$\underline{r} = a_1 \underline{h}_1 + a_2 \underline{h}_2 + \dots + a_n \underline{h}_n + \underline{n}$
 (combinación lineal de columnas)

$\hat{a}_i =$ cuantificar $\{\tilde{a}_i\}$
 $\tilde{a}_i = a_1 \underbrace{\underline{h}_i^H \underline{h}_1}_{\text{correlaciones cruzadas entre columnas}} + a_2 \underline{h}_i^H \underline{h}_2 + \dots + a_i \underbrace{\|\underline{h}_i\|^2}_{\text{autocorrelación de la columna consigo misma}} + \dots + a_n \underline{h}_i^H \underline{h}_n + \underbrace{\underline{h}_i^H \underline{n}}_{\underline{w}_i}$

Conclusiones filtro adaptado:

- Es óptimo en ausencia de interferencias (maximiza señal útil-a-ruido)
- El descentralizado sólo necesita conocer UNA columna de la matriz del canal
- su comportamiento es cercano al óptimo si las columnas de \underline{H} son ortogonales
- Es particularmente sensible al problema cerca-lejos (cuando $\|\underline{h}_j\| \gg \|\underline{h}_i\|$)

Cota: "Receptor mágico" que elimina interferencias
 Receptor mágico haría
 $\underline{z} = y - \underbrace{(\underline{R} - \underline{R}^d)}_{\text{resta interf.}} \underline{a} = \underbrace{\underline{R}^d}_{\text{señal}} \underline{a} + \underbrace{\underline{w}}_{\text{ruido}} \underline{w}_i$

cota de filtro adaptado
 $\boxed{SNR_{MFB, i} = \frac{\|\underline{h}_i\|^2}{N_0}}$ $\xrightarrow{\text{MSE si } \|\hat{a}_i\|=1}$ $MSE = \frac{N_0}{\|\underline{h}_i\|^2}$

Receptor Lineal: Forzador de ceros (Zero Forcing) (ZF)

Objetivo: Eliminar completamente las interferencias

$$\underline{C} \cdot \underline{H} = \underline{I}$$

→ si $m=n \Rightarrow \underline{C} = \underline{H}^{-1}$ (solución única)

→ si $m > n \Rightarrow \underline{C} = (\underline{H}^H \cdot \underline{H})^{-1} \cdot \underline{H}^H$

pseudoinversa \underline{H}^+

si $m > n$, múltiples soluciones. Se escoge la que minimice el error cuadrático medio

$$MSE = E\{\|\underline{C}\underline{r} - \underline{a}\|^2\}$$

si $\underline{C} = \underline{W} \cdot \underline{H}^H$

$$\Rightarrow \underline{W}(\underline{H}^H \cdot \underline{H}) = \underline{I}$$

$$\Leftrightarrow \underline{W} = (\underline{H}^H \cdot \underline{H})^{-1}$$

Interpretación:

El objetivo era minimizar $\|\underline{r} - \underline{H}\underline{a}\|$
Lo primero que diría un matemático es:

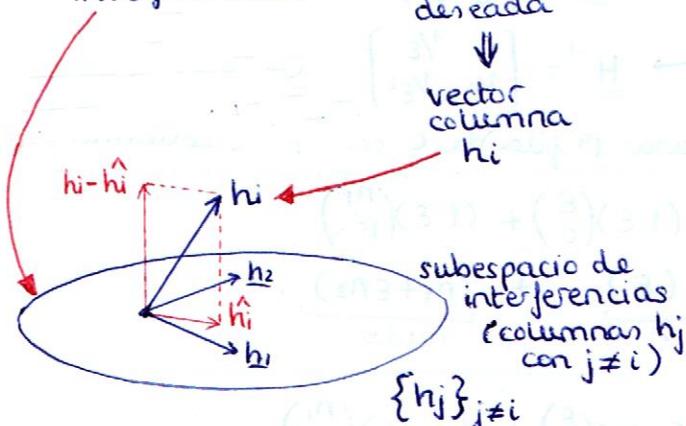
$$\tilde{\underline{a}} = \underline{H}^+ \cdot \underline{r} = \underbrace{\underline{H}^+ \underline{H}}_{\underline{I}} \underline{a} + \underbrace{\underline{H}^+ \underline{n}}_{\text{problema: término incontrolado}}$$

El ruido se puede disparar

otro problema: \hat{a} no puedo tomar cualquier valor, debe pertenecer al codebook (hay que cuantificar) $\hat{a} = \text{cuantif}\{\tilde{\underline{a}}\}$

Interpretación geométrica:

$$\underline{r} = \underbrace{a_1 \underline{h}_1 + a_2 \underline{h}_2 + \dots + a_i \underline{h}_i + \dots + \underline{n}}_{\text{interferencias}} + \underbrace{\dots}_{\text{señal deseada}}$$



$\hat{h}_i \equiv$ proyección de h_i sobre $\{h_j\}_{j \neq i}$

$h_i - \hat{h}_i =$ vector perpendicular al subespacio de interferencias

$$\begin{bmatrix} \underline{C} \\ \underline{C} \\ \underline{C} \end{bmatrix} \cdot \begin{bmatrix} \underline{H} \\ \underline{H} \\ \underline{H} \end{bmatrix} = \underline{a}$$

filas de \underline{C}

Sabemos que el ZF me quita las interferencias $\Leftrightarrow \underline{w}_i^H \cdot \underline{h}_j = 0$

Por tanto se deduce que: \underline{w}_i es ortogonal al subespacio de interf.

$$\underline{w}_i = \text{cte} \cdot (\underline{h}_i - \hat{h}_i)$$

$$\|\underline{w}_i \cdot \underline{h}_i\| = 1 \Rightarrow \|\underline{w}_i\| \cdot \|\underline{h}_i - \hat{h}_i\| = 1 \Rightarrow \frac{1}{\|\underline{h}_i - \hat{h}_i\|^2}$$

si $\|\underline{h}_i - \hat{h}_i\|$ es pequeña, habrá "amplificación del ruido"

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \underline{C} \\ \underline{C} \\ \vdots \\ \underline{w}_i^H \\ \vdots \end{pmatrix} \cdot \begin{pmatrix} \underline{H} \\ \underline{H} \\ \vdots \\ h_i \\ \vdots \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{pmatrix}$$

si nos fijamos sólo en y_i (descentraliz.)

$$y_i = [-w_i^H] \cdot \left[a_1 h_1 \quad a_2 h_2 \quad \dots \quad a_i h_i \quad \dots \right]$$

$$y_i = 0 \cdot a_1 + 0 \cdot a_2 + \dots + \underline{1} \cdot a_i + \dots + \underline{w}_i^H \underline{n}$$

escoger \underline{w}_i

$$\|\underline{w}_i\| = \frac{1}{\|\underline{h}_i - \hat{h}_i\|}$$

• Prestaciones Zero-Forcing

$$\underline{y} = \underbrace{\underline{H}^H \underline{H}}_{\underline{I}} \underline{a} + \underbrace{\underline{H}^H \underline{n}}_{\tilde{\underline{n}}} \implies \underline{y} = \underline{a} + \tilde{\underline{n}}$$

$$y_i = \underbrace{0 \cdot a_1 + \dots + 1 \cdot a_i + \dots + 0 \cdot a_n}_{\text{anula interj}} + \underbrace{w_i \cdot n}_{\text{deseada}}$$

$$\|w_i\| = \frac{1}{\|h_i - \hat{h}_i\|}$$

amplifica ruido si $\|h_i - \hat{h}_i\| \ll 1$

Ruido filtrado.
Matriz de correlación
 $E\{\tilde{\underline{n}} \cdot \tilde{\underline{n}}^H\} = N_0 \cdot \underline{R}^{-1}$
Inversa de la matriz de correlación del canal

Para el símbolo i : $MSE_i = N_0 (\underline{R}^{-1})_{ii} \stackrel{\text{se puede demostrar}}{=} \frac{N_0}{\|h_i - \hat{h}_i\|^2}$

Comparando con el MF con cancelación ideal

$$\frac{MSE_{MF,i}}{MSE_i} = \frac{N_0 / \|h_i\|^2}{N_0 / \|h_i - \hat{h}_i\|^2} = 1 - \frac{\|\hat{h}_i\|^2}{\|h_i\|^2}$$

• El MZ se acerca al MF con cancelación perfecta de interferencias cuando \hat{h}_i es ortogonal a $\text{span}\{h_j\}_{j \neq i}$

ejemplo: MF y ZF

canal $\underline{H} = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix}$

$$\underline{H}^H = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix} = \underline{C}_{MF}$$

$$\underline{H}^{-1} = \begin{bmatrix} 0 & 1/\epsilon \\ 1/\epsilon & 1/\epsilon^2 \end{bmatrix} = \underline{C}_{ZF}$$

Decodifiquemos solo y_1 (usar 1ª fila de \underline{C} con cada columna de \underline{H})

$$y_1^{MF} = a_1 (1 \ \epsilon) \cdot \begin{pmatrix} 1 \\ \epsilon \end{pmatrix} + a_2 (1 \ \epsilon) \cdot \begin{pmatrix} \epsilon \\ 0 \end{pmatrix} + (1 \ \epsilon) \cdot \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

$$= a_1 (1 + \epsilon^2) + \underbrace{a_2 (\epsilon)}_{\text{interj}} + \underbrace{(n_1 + \epsilon n_2)}_{\text{ruido}}$$

$$y_1^{ZF} = a_1 (0 \ 1/\epsilon) \cdot \begin{pmatrix} 1 \\ \epsilon \end{pmatrix} + a_2 (0 \ 1/\epsilon) \cdot \begin{pmatrix} \epsilon \\ 0 \end{pmatrix} + (0 \ 1/\epsilon) \cdot \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

$$= a_1 \cdot 1 + \underbrace{a_2 \cdot 0}_{\text{sin interj.}} + \underbrace{\frac{1}{\epsilon} n_2}_{\text{amplifica ruido si } \epsilon \downarrow}$$

MMSE $\rightarrow \underline{C} = \left(\begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix}^H \begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix} + N_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix}^H$

pag. siguiente

$$= \left(\begin{bmatrix} 1 + \epsilon^2 & \epsilon \\ \epsilon & \epsilon^2 \end{bmatrix} + N_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix} = \begin{bmatrix} 1 + \epsilon^2 + N_0 & \epsilon \\ \epsilon & \epsilon^2 + N_0 \end{bmatrix}^{-1} \begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix}$$

$$= \frac{1}{(1 + \epsilon^2 + N_0)(\epsilon^2 + N_0) - \epsilon^2} \begin{bmatrix} \epsilon^2 + N_0 & -\epsilon \\ -\epsilon & 1 + \epsilon^2 + N_0 \end{bmatrix} \cdot \begin{bmatrix} 1 & \epsilon \\ \epsilon & 0 \end{bmatrix} = \frac{1}{\epsilon^4 + N_0(1 + 2\epsilon^2 + N_0)} \begin{bmatrix} N_0 & \epsilon^3 + N_0 \epsilon \\ \epsilon^3 + N_0 \epsilon & -\epsilon^2 \end{bmatrix}$$

más sencillo con la otra fórmula para cada fila

$$\underline{w}_i^H = \left(\underline{h}_i \underline{h}_i^H + \sum_{j \neq i} \underline{h}_j \underline{h}_j^H + N_0 \underline{I} \right)^{-1} \underline{h}_i$$

$$= \left(\begin{bmatrix} 1 \\ \epsilon \end{bmatrix} \begin{bmatrix} 1 & \epsilon \end{bmatrix} + \begin{bmatrix} \epsilon \\ 0 \end{bmatrix} \begin{bmatrix} \epsilon & 0 \end{bmatrix} + N_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1 \\ \epsilon \end{bmatrix} = \frac{1}{\frac{\epsilon^4}{N_0} + 1 + 2\epsilon^2 + N_0} \begin{bmatrix} 1 \\ \epsilon + \frac{\epsilon^3}{N_0} \end{bmatrix}$$

Detector lineal MMSE (Minimum Mean Squared Error)

Como siempre: Matched-Filter: maximiza señal-útil-a-ruido
sin importarle las interferencias
Zero-Forcing: anula las interferencias
sin importarle el ruido

MMSE: compromiso ruido e interferencias
 \underline{C} que minimiza $\{\|\underline{C} \cdot \underline{r} - \underline{a}\|^2\}$ (que se parezca lo máximo posible)

se obtiene:

$$\underline{C} = \left(\underline{H}^H \underline{H} + \underbrace{N_0 \underline{I}}_{\substack{\text{añade} \\ \text{esto al} \\ \text{ZF}}} \right)^{-1} \underline{H}^H$$

← expresión mejor para implementación

expresión mejor para análisis

$$\underline{C} = \underline{H}^H (\underline{H} \underline{H}^H + N_0 \underline{I})^{-1}$$

- Cuando $N_0 \rightarrow \infty \Rightarrow$ MMSE tiende a MF ($\underline{w}_i \rightarrow \frac{h_i}{N_0}$)
- Cuando interferencias tienden a cero $\|h_j\|_{j \neq i} = 0 \Rightarrow$ MMSE tiende a MF ($\underline{w}_i \rightarrow \frac{h_i}{\|h_i\|^2 + N_0}$)
- Cuando $N_0 \rightarrow 0 \Rightarrow$ MMSE tiende a ZF

→ Detector descentralizado

$$y_i^{MMSE} = \underline{w}_i^H \cdot \underline{r} \quad \text{siendo } \underline{w}_i^H \text{ la fila } i\text{-ésima de } \underline{C}$$

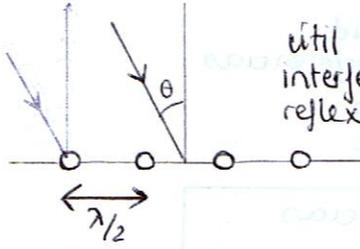
$$\underline{w}_i = \left(\underline{h}_i \underline{h}_i^H + \sum_{j \neq i} \underline{h}_j \underline{h}_j^H + N_0 \underline{I} \right)^{-1} \underline{h}_i$$

$$\underline{w}_i = (\underline{H} \underline{H}^H + N_0 \underline{I})^{-1} \underline{h}_i$$

ejemplo: $\underline{H} = \begin{bmatrix} 2 & A_2 \\ 1 & 0 \end{bmatrix}$ Calcular MMSE descentralizado para el primer símbolo:

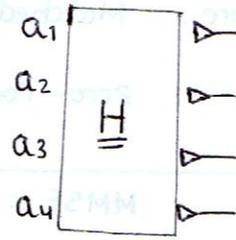
$$\begin{aligned} \underline{w}_1 &= \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \end{bmatrix} + \begin{bmatrix} A_2 \\ 0 \end{bmatrix} \begin{bmatrix} A_2 & 0 \end{bmatrix} + N_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \left(\begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} A_2^2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} N_0 & 0 \\ 0 & N_0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 4+A_2^2+N_0 & 2 \\ 2 & 1+N_0 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \frac{1}{(4+A_2^2+N_0)(1+N_0)-4} \begin{bmatrix} 1+N_0 & -2 \\ -2 & 4+A_2^2+N_0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \frac{1}{5N_0+N_0^2+A_2^2(1+N_0)} \begin{bmatrix} 2N_0 \\ A_2^2+N_0 \end{bmatrix} \end{aligned}$$

problema: Array lineal visto como MIMO



util $\theta = 15^\circ$
 interferencias $\theta = 30^\circ, 70^\circ, 80^\circ$
 reflexión util $\theta = 50^\circ, \alpha = 0.19$

Modelo MIMO



Steering vector $\underline{V}(\theta) = \begin{bmatrix} 1 \\ e^{j\pi \text{sen} \theta} \\ e^{j2\pi \text{sen} \theta} \\ e^{j3\pi \text{sen} \theta} \end{bmatrix}$

Se puede deducir el modelo MIMO como:

$$\underline{H} = \begin{bmatrix} \underline{V}(15^\circ) + 0.19 \underline{V}(50^\circ) & \underline{V}(30^\circ) & \underline{V}(70^\circ) & \underline{V}(80^\circ) \end{bmatrix}$$

$\underline{h}_1 \qquad \underline{h}_2 \qquad \underline{h}_3 \qquad \underline{h}_4$

Detector descentralizado; nos interesa primera fila (\underline{w}_1^H) de \underline{C}

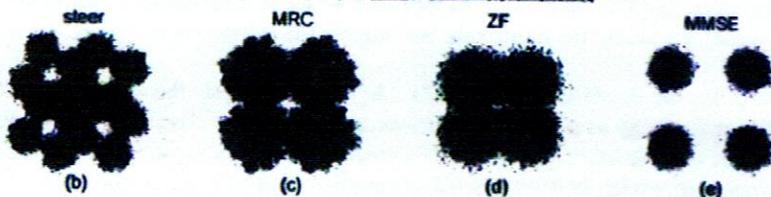
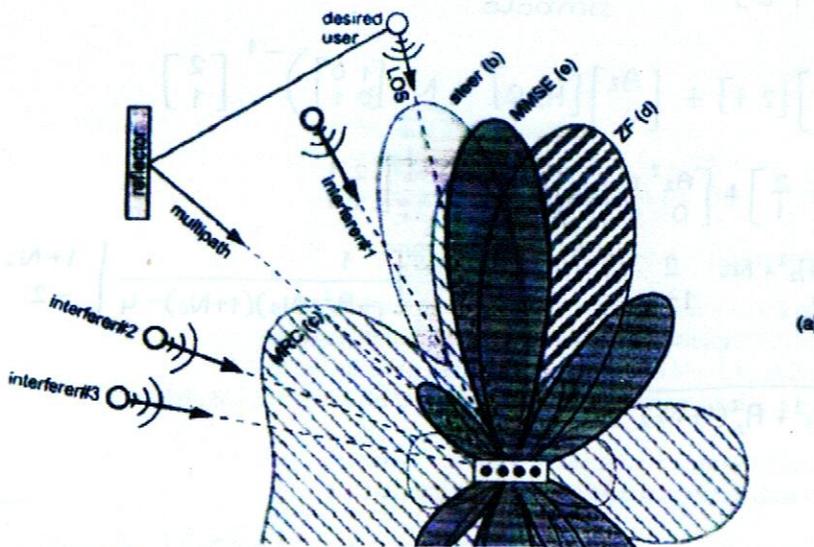
$$\underline{y} = \underline{w}_1^H \cdot \underline{r}$$

→ Beamsteering detector: $\underline{w}_1 = \underline{V}(15^\circ)$

→ Matched Filter $\underline{w}_1 = \underline{h}_1 \equiv$ Max ratio combiner

→ Zero Forcing $\underline{w}_1 = (\underline{h}_1 \underline{h}_1^H + \sum_{2,3,4} \underline{h}_j \underline{h}_j^H)^{-1} \cdot \underline{h}_1$

→ MMSE $\underline{w}_1 = (\underline{h}_1 \underline{h}_1^H + \sum \underline{h}_j \underline{h}_j^H + N_0 \underline{I})^{-1} \cdot \underline{h}_1$



Detección con decisión realimentada (Decision Feedback)

- Realimentar las decisiones tomadas sobre algunos símbolos para eliminar la interferencia de éstos sobre los restantes y mejorar su detección
- Se aplica tanto al ZF como al MMSE

◦ ZF-DF : BLAST nulling and cancellation detector / Generalised DFE

→ Interpretación SIC : Successive Interference Cancellation

Dos pasos

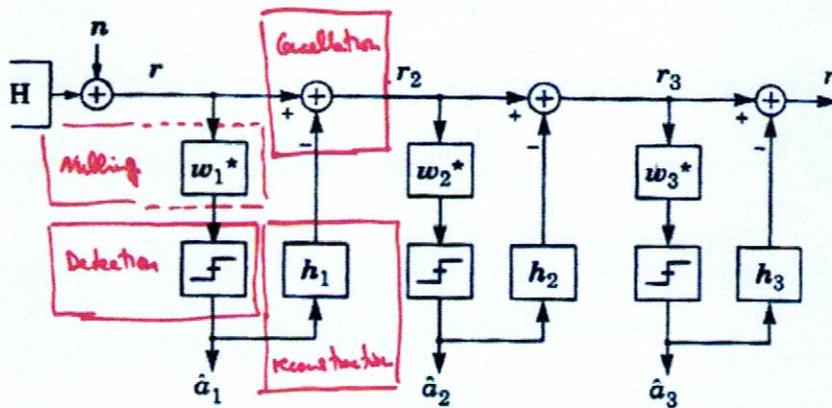
① Nulling: se detecta $\hat{a}_1 = \text{quantice} \{ \underline{w}_1^H \cdot \underline{r} \}$
 ↑ símbolo mas cercano ↑ se escoge para que elimine interferencias del resto (ZF)
 Es el "nulling vector"

② Cancellation: se elimina la contribución de \hat{a}_1 sobre \underline{r}

$$\underline{r}_2 = \underline{r} - \hat{a}_1 \underline{h}_1$$

③ Volver a ① para \hat{a}_{i+1}

orden de la detección
 interesa detectar primero los símbolos que llegan mejor (encontrar matriz de permutación)



10-13. The MIMO DF detector consists of n stages, one for each channel input. Each stage consists of four steps: nulling, detection, reconstruction, and cancellation.

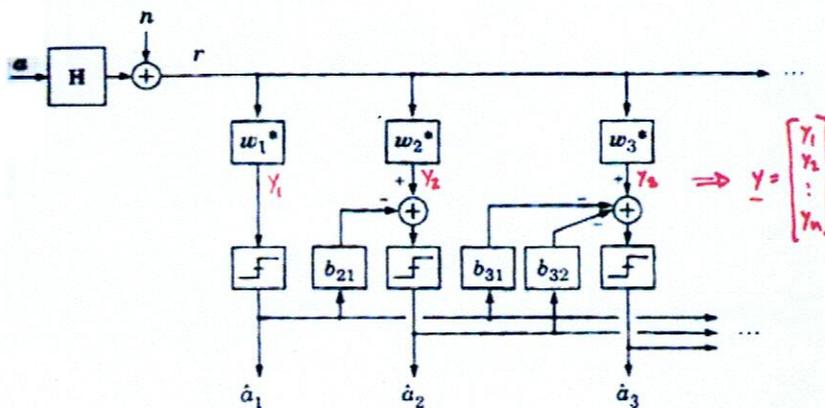


Fig. 10-14. An alternative but equivalent implementation of the DF detector of Fig. 10-13.

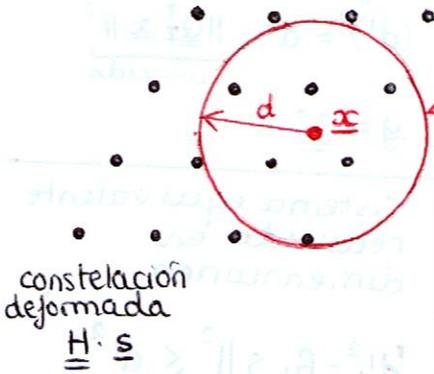
$b_{ij} = \underline{a}_i^H \underline{h}_j$

MIMO detection: SPHERE DECODING

(nueva nomenclatura ... otra vez!)

→ Concepto: radio de búsqueda

$$\underline{x} = \underline{H} \underline{s} + \underline{n}$$



Se busca solución solo a una distancia d del vector recibido \underline{x}

$d \uparrow \uparrow$ → tiende a búsqueda exhaustiva
 $d \downarrow \downarrow$ → ninguna posible solución

Elección de 'd'

- Radio de Babai: usar distancia a la solución subóptima ZF (y seguro que la óptima será menor o igual)
- Radio adaptativo
- Varianza del ruido escalada

→ Concepto: búsqueda en árbol

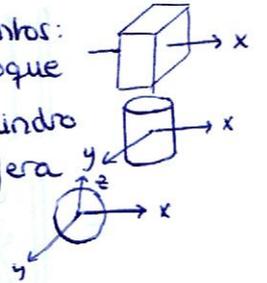
En lugar de calcular la distancia a cada solución directamente

ej: 3 dimensiones x, y, z

$$dx^2 + dy^2 + dz^2 \leq d^2$$

se va calculando por dimensiones, de forma que vamos haciendo una distancia acumulada que podemos ir comparando con el radio d para ir descartando

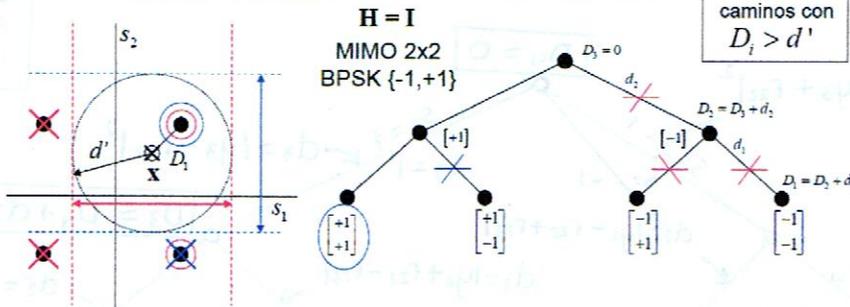
- Primero calculo: $D_3 = dx^2 \rightarrow D_3 \leq d^2 \rightarrow$ bloque
- Segundo calculo: $D_2 = dx^2 + dy^2 \rightarrow D_2 \leq d^2 \rightarrow$ cilindro
- Tercero calculo: $D_1 = dx^2 + dy^2 + dz^2 \rightarrow D_1 \leq d^2 \rightarrow$ esfera



distancias acumuladas
 ej: $D_2 = D_1 + dy^2$
 $D_3 = D_2 + dz^2$

A cada paso calculamos distancia en una nueva dimensión, la sumamos a la distancia acumulada, y descartamos puntos que caigan fuera

ANALOGÍA ESFERA ↔ ÁRBOL



d_i : distancia Euclídea parcial del nivel i , pesos de las ramas
 D_i : distancia Euclídea acumulada desde la raíz hasta el nivel i
 d' : Radio de búsqueda de la esfera

Solución: Camino con menor D_i

Desarrollo matemático

$$\|x - Hs\|^2 \leq d^2 \Rightarrow$$

$$H = Q \cdot R = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

unitaria triangular superior

Recuerda: QR: ir consiguiendo ceros en la matriz con transform. unitarias que voy agrupando en Q

$$\Rightarrow \|x - QBs\|^2$$

$$= \|x - [Q_1 \ Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} s\|^2$$
x Q no cambia norma

$$= \left\| \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} x - \begin{bmatrix} R_1 \\ 0 \end{bmatrix} s \right\|^2$$

$$= \underbrace{\|Q_1^T x - R_1 s\|^2}_{y} + \|Q_2^T x\|^2$$

Definiendo

$$(d')^2 = d^2 - \underbrace{\|Q_2^T x\|^2}_{\text{conocido}}$$

$$y = Q_1^T x$$

Sistema equivalente reducido en dimensiones

$$\|y - R_1 s\|^2 \leq d'^2$$

↑ mang.



$$\sum_{i=1}^{n_T} \left| y_i - \sum_{j=i}^{n_T} r_{ij} s_j \right|^2 \leq d'^2$$

↑ se puede ir calculando cada término desde $n_T \dots 1$ e ir descartando

ejemplo: $H_{3 \times 3}$

$$\left\| \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} - \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \right\|^2 \leq d'^2$$

$$= \left\| \begin{bmatrix} y_1 - r_{11}s_1 - r_{12}s_2 - r_{13}s_3 \\ y_2 - r_{22}s_2 - r_{23}s_3 \\ y_3 - r_{33}s_3 \end{bmatrix} \right\|^2 = d_1 + d_2 + d_3 \leq d'^2$$

$$d_1 = |y_1 - r_{11}s_1 - r_{12}s_2 - r_{13}s_3|^2$$

$$d_2 = |y_2 - r_{22}s_2 - r_{23}s_3|^2$$

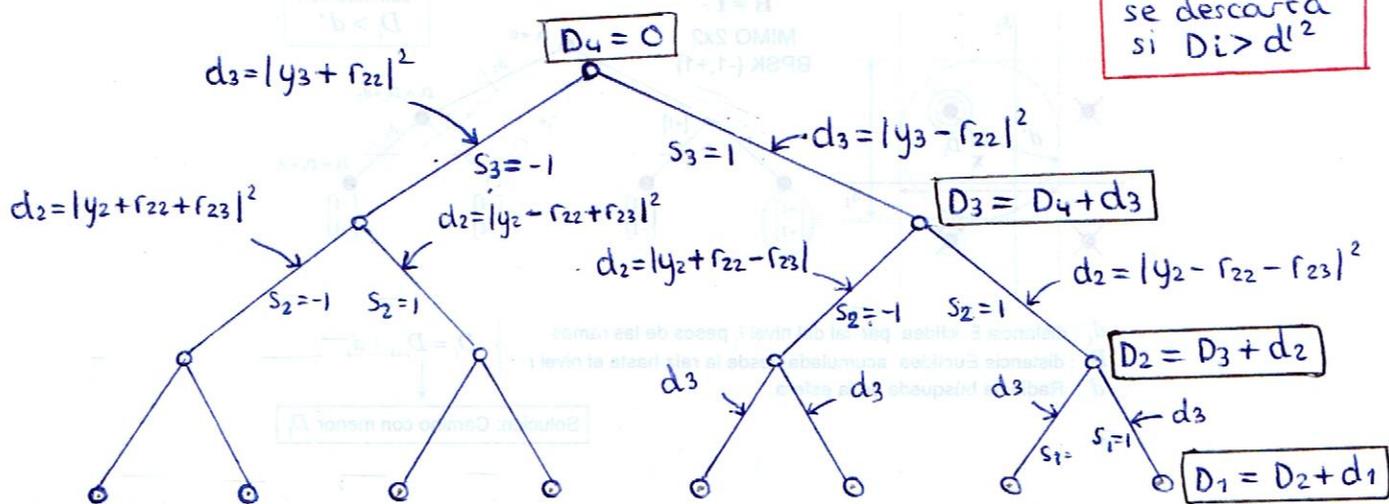
$$d_3 = |y_3 - r_{33}s_3|^2$$

$$D_3 = d_3$$

$$D_2 = D_3 + d_2$$

$$D_1 = D_2 + d_1$$

En cada nivel se descarta si $D_i > d'^2$



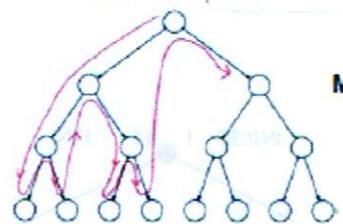
Algoritmos Depth-First

- i. Fincke-Pohst

Algoritmos Breadth-First

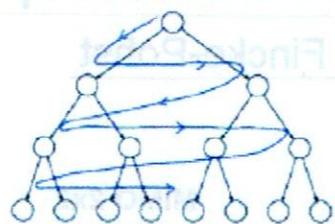
- i. K-Best
- ii. K-Best dinámico
- iii. Automatic Sphere Decoder

Búsqueda Depth-First (profundidad)



- Recorre de arriba a abajo de izqda a dcha
- Desordenada
- Necesita memoria del estado del árbol

Búsqueda Breadth-First (anchura)

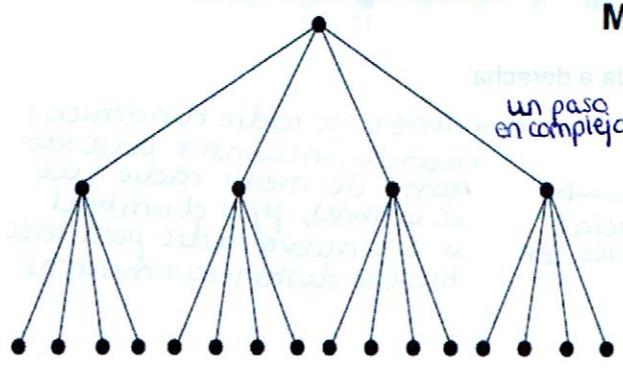


- Recorre por niveles
- Ordenada
- Estado del árbol conocido

MIMO 3x3
BPSK

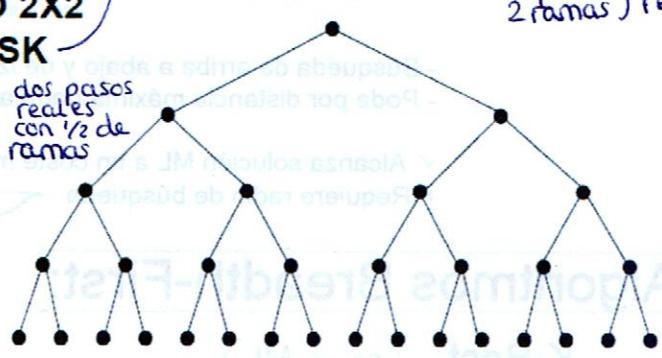
Modelo real del sistema

ÁRBOL COMPLEJO



MIMO 2X2
QPSK

ÁRBOL REAL



un paso en complejo

dos pasos reales con 1/2 de ramas

2 símbolos recibidos → 2 pasos } en complejo
4 ramas en cada paso

4 pasos } en real
2 ramas

Problema complejo: $\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n}$ → Problema real: $\mathbf{x}_r = \mathbf{H}_r\mathbf{s}_r + \mathbf{n}_r$

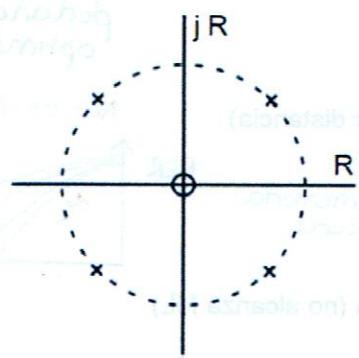
- Número de niveles será el doble
- Número de ramas desde un nodo será el \log_2 del original (las constelaciones pasan a expresarse en su forma real).

$$\mathbf{s}_r = \begin{bmatrix} R(\mathbf{s})^T & I(\mathbf{s})^T \end{bmatrix}^T$$

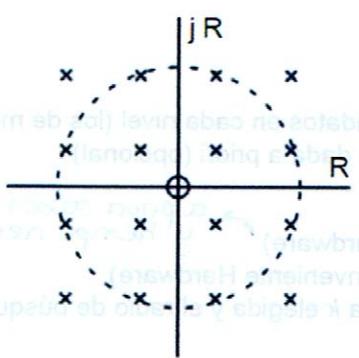
$$\mathbf{x}_r = \begin{bmatrix} R(\mathbf{x})^T & I(\mathbf{x})^T \end{bmatrix}^T$$

$$\mathbf{n}_r = \begin{bmatrix} R(\mathbf{n})^T & I(\mathbf{n})^T \end{bmatrix}^T$$

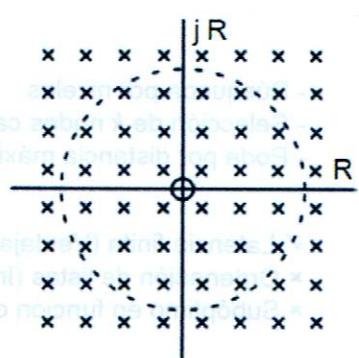
$$\mathbf{H}_r = \begin{bmatrix} R(\mathbf{H}) & -I(\mathbf{H}) \\ I(\mathbf{H}) & R(\mathbf{H}) \end{bmatrix}$$



QPSK



16-QAM



64-QAM

$$\chi = \frac{1}{\sqrt{2}} \{-1, +1\}$$

$$\chi = \frac{1}{\sqrt{10}} \{-3, -1, +1, +3\}$$

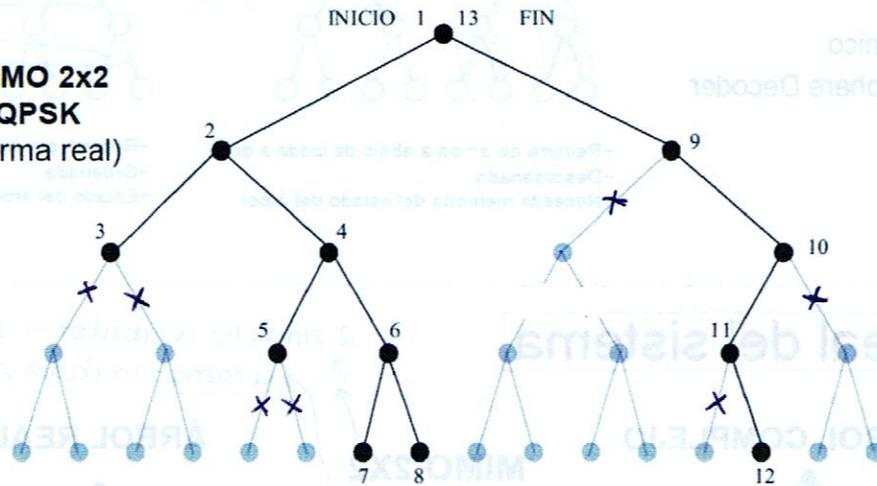
$$\chi = \frac{1}{\sqrt{42}} \{-7, -5, -3, -1, +1, +3, +5, +7\}$$

χ : denota la representación en forma real de las distintas constelaciones.

Algoritmos Depth-First:

Fincke-Pohst

MIMO 2x2
QPSK
(Forma real)



- Búsqueda de arriba a abajo y de izquierda a derecha
- Poda por distancia máxima dada a priori

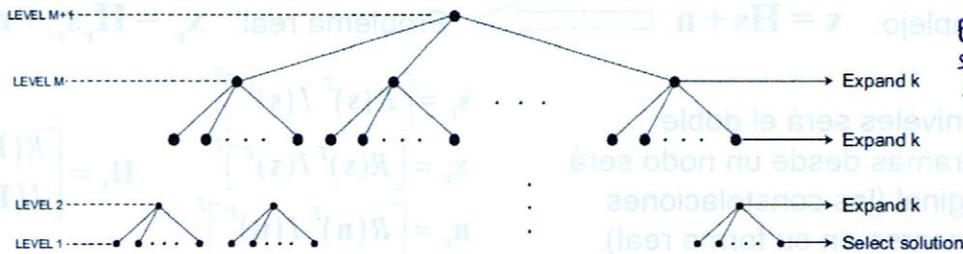
- ✓ Alcanza solución ML a un coste menor
- × Requiere radio de búsqueda

Conviene usar radio dinámico:
- cuando encuentre un nodo final de menor radio que el umbral, fijar el umbral a ese nuevo radio para sólo buscar distancias menores

inicial típico: ZF

Algoritmos Breadth-First:

K-Best (no es ML)



En cada nivel se expanden sólo los k mejores (y se poda el resto)
↓
no se garantiza que no estés podando el óptimo!

- Búsqueda por niveles
- Selección de k nodos candidatos en cada nivel (los de menor distancia)
- Poda por distancia máxima dada a priori (opcional)

- ✓ Latencia finita (Ventaja Hardware) → a priori saben la memoria y tiempo necesario
- × Ordenación de listas (Inconveniente Hardware)
- × Subóptimo en función de la k elegida y el radio de búsqueda (no alcanza ML)

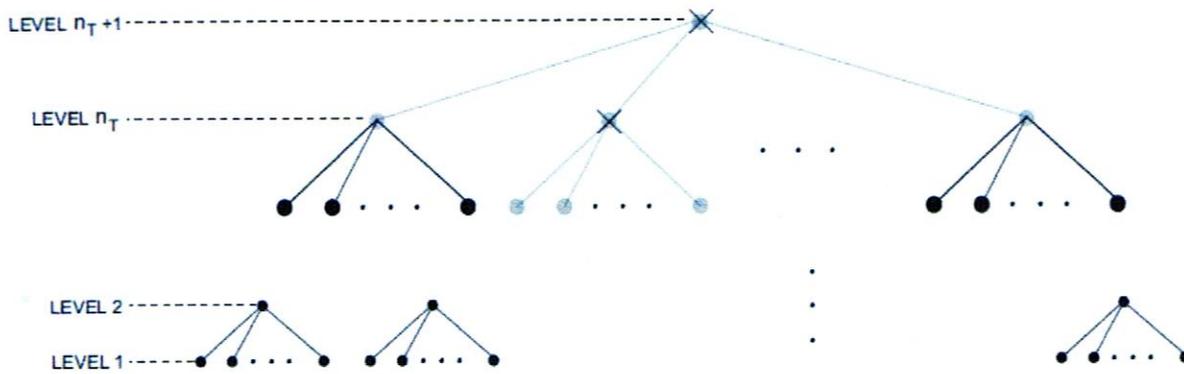


K-Best dinámico (no es ML)

- Misma filosofía que K-Best
- Utiliza valores de k variables por niveles
- Interesan valores de k mayores en niveles más altos

- ✓ Evita descartar la solución ML en etapas iniciales consiguiendo reducir coste frente a K-Best fijo
- × Más complicado de implementar que K-Best fijo

Automatic SD

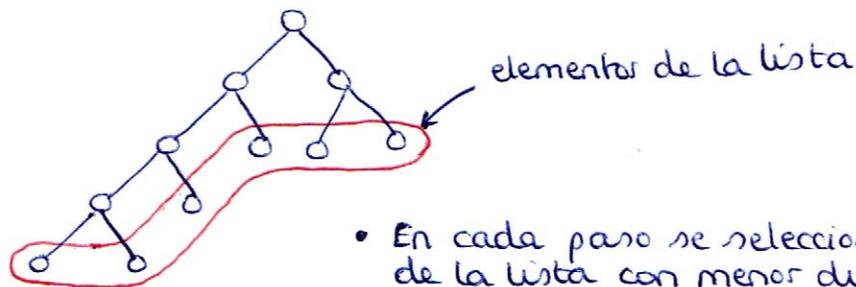


Pasos:

1. Inicializar lista con nodo raíz
 2. Expandirlo, eliminarlo de la lista e introducir hijos
 3. Seleccionar el nodo con menor d de la lista
 4. Volver al paso 2 con el nodo seleccionado
- } Proceder así hasta llegar al nivel inferior.
Nodo con menor d será la solución ML.

- ✓ Algoritmo óptimo: alcanza la solución ML
- ✓ No requiere radio de búsqueda
- ✓ El más eficiente computacionalmente de los óptimos
- * Requiere listas ordenadas (Inconveniente Hardware)

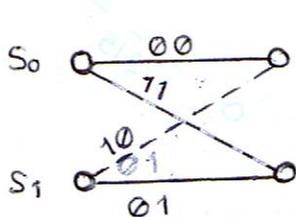
En cada paso se tiene una lista con el "frente de ondas"



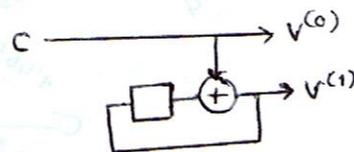
- En cada paso se selecciona el nodo de la lista con menor distancia y se expande, eliminando el nodo e incluyendo a los hijos en la lista
- Puesto que los hijos siempre tienen distancia acumulada mayor o igual que el padre, en cuanto 'toquemos' abajo del todo sabemos que es la mejor solución segura.

- Es ML
- Es el que menos ramas explora de los ML

Rejilla del primer codificador



— ≡ entra 0
 --- ≡ entra 1



Inicio de la decodificación Log-MAP

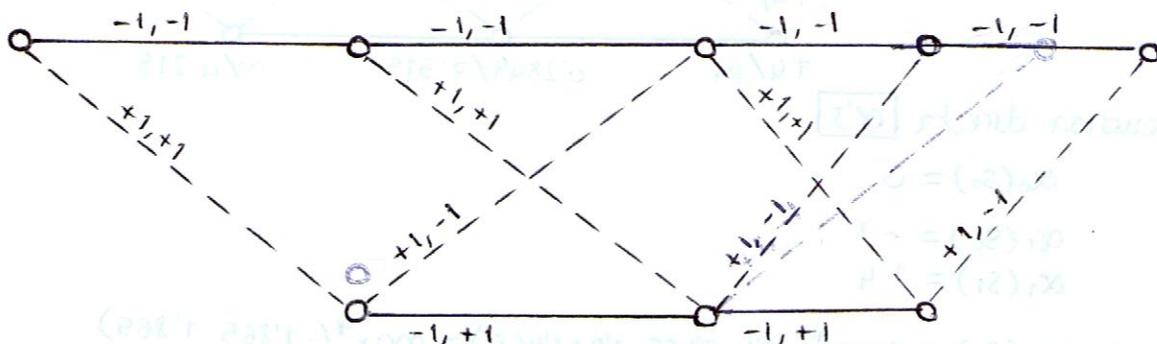
$$\begin{cases} L_c = 2 \\ L_a = [-0'4, -0'07, -0'8, 2'03] \end{cases}$$

$$\Gamma' = 0'8, 0'1$$

$$1, -0'5$$

$$-1'8, 1'1$$

$$1'6, -1'6$$



Primero:

cálculo de γ ramas

$$\gamma_1(0,0) = (-1) \cdot \frac{(-0'4)}{2} + \frac{2}{2}(-0'8 - 0'8) = -1'4$$

$$\gamma_1(0,1) = (1) \cdot \frac{-0'4}{2} + \frac{2}{2}(0'8 + 0'8) = 1'4$$

$$\gamma_2(0,0) = (-1) \cdot \frac{(-0'07)}{2} + (-1 + 0'5) = -0'465$$

$$\gamma_2(0,1) = (1) \cdot \frac{(-0'07)}{2} + (1 - 0'5) = 0'465$$

$$\gamma_2(1,0) = (1) \cdot \frac{(-0'07)}{2} + (1 + 0'5) = 1'465$$

$$\gamma_2(1,1) = (-1) \cdot \frac{(-0'07)}{2} + (-1 - 0'5) = -1'465$$

$$\gamma_3(0,0) = (-1)(-0'4) + (1'8 - 1'1) = 1'1$$

$$\gamma_3(0,1) = (1)(-0'4) + (-1'8 + 1'1) = -1'1$$

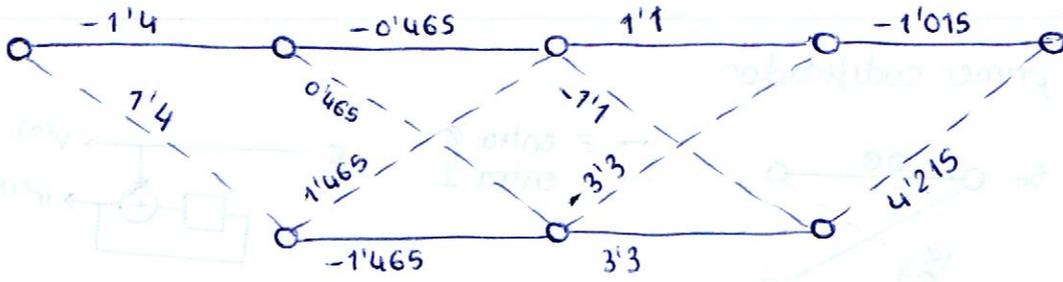
$$\gamma_3(1,0) = (1)(-0'4) + (-1'8 - 1'1) = -3'3$$

$$\gamma_3(1,1) = (-1)(-0'4) + (1'8 + 1'1) = 3'3$$

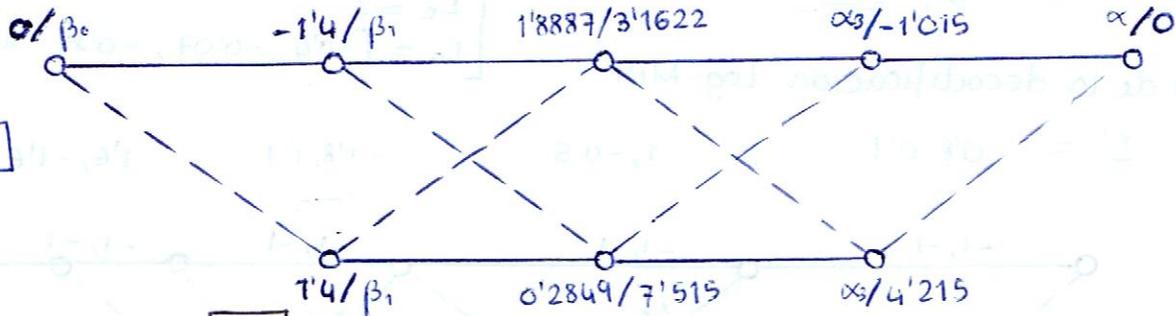
$$\gamma_u(0,0) = (-1)(1'015) + (-1'6 + 1'6) = -1'015$$

$$\gamma_u(1,0) = (1)(1'015) + (1'6 + 1'6) = 4'215$$

Y



α/β



Recursion directa α 's

$$\alpha_0(s_0) = 0$$

$$\alpha_1(s_0) = -1'4$$

$$\alpha_1(s_1) = 1'4$$

$$\begin{aligned} \alpha_2(s_0) &= \max^*(-1'4 - 0'465, 1'4 + 1'465) = \max^*(-1'865, 1'865) \\ &= \max(-1'865, 1'865) + \ln(1 + e^{-1'865 - (-1'865)}) \\ &= 1'865 + \ln(1 + e^{-3'73}) = 1'8887 \end{aligned}$$

$$\begin{aligned} \alpha_2(s_1) &= \max^*(-1'4 + 0'465, 1'4 - 1'465) = \max^*(-0'935, -0'065) \\ &= -0'065 + \ln(1 + e^{-0'87}) = 0'2849 \end{aligned}$$

$$\begin{aligned} \alpha_3(s_0) &= \max^*(1'8887 + 1'1, 0'2849 - 3'3) = \max^*(2'9887, -3'0151) \\ &= 2'9887 + \ln(1 + e^{-6'0038}) = 2'991 \end{aligned}$$

$$\alpha_3(s_1) =$$

no hace falta mas porque piden LLR en t=2

Recursion inversa β 's

$$\beta_4(s_0) = 0$$

$$\beta_3(s_0) = -1'015$$

$$\beta_3(s_1) = 4'215$$

$$\begin{aligned} \beta_2(s_0) &= \max^*(-1'015 + 1'1, 4'215 - 1'1) = \max^*(0'085, 3'115) \\ &= 3'115 + \ln(1 + e^{-3'03}) = 3'1622 \end{aligned}$$

$$\begin{aligned} \beta_2(s_1) &= \max^*(-1'015 - 3'3, 4'215 + 3'3) = \max^*(-4'315, 7'515) \\ &= 7'515 + \ln(1 + e^{-11'83}) = 7'515 \end{aligned}$$

Cálculo de los LLR en $t=2$

$$\begin{aligned}
 L(C_2, \underline{r}) &= \max_{(s', s)'}^* \left(\overbrace{-1'4 + 0'465 + 7'515}^{\alpha}, \overbrace{1'4 + 1'465 + 3'1622}^{\delta}, \overbrace{}^{\beta} \right) \\
 &\quad \text{lineas discontinuas} \\
 &= \max_{(s', s)'}^* \left(-1'4 - 0'465 + 3'1622, 1'4 - 1'465, 7'515 \right) \\
 &\quad \text{lineas continuas} \\
 &= \max^*(6'58, 6'0272) - \max^*(1'2972, 7'45) \\
 &= 6'58 + \ln(1 + e^{-0'5528}) - (7'45 + \ln(1 + e^{-6'1528})) \\
 &= 7'0345 - 7'4521 \\
 &= -0'4176
 \end{aligned}$$

Cálculo de la información extrínseca

$$L(C_2 | \underline{r}) = L_a(C_2) + L_c \cdot r_{t0} + L_e(C_t)$$

$$\begin{aligned}
 L_e(C_t) &= L(C_2 | \underline{r}) - L_a(C_2) - L_c \cdot r_{t0} \quad \leftarrow \text{el bit sistemático en } t=2 \\
 &= (-0'4176) - (-0'07) - 2(1) \\
 &= -2'3476
 \end{aligned}$$

COMUNICACIONES II
CONVOCATORIA ORDINARIA 18-junio-2008

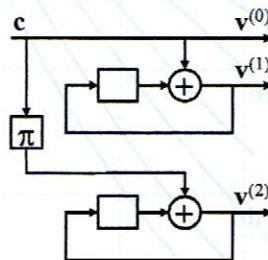
Nombre: Apellido 1º Apellido 2º

C1	C2	C3	C4	C5	C6

- La duración del examen es de 1,30 h.
- Responda en el espacio habilitado para cada cuestión.

Cuestión 1 (2,5 puntos)

Se tiene un sistema de comunicaciones con el siguiente codificador:



En recepción se plantea usar un decodificador iterativo empleando el algoritmo Log-MAP:

$$L(c_t | \mathbf{r}) = \max_{(s', s)_t}^* \left[\bar{\beta}_t(s) + \bar{\gamma}_t(s', s) + \bar{\alpha}_{t-1}(s') \right] - \max_{(s', s)_t}^* \left[\bar{\beta}_t(s) + \bar{\gamma}_t(s', s) + \bar{\alpha}_{t-1}(s') \right]$$

$$\bar{\alpha}_t(s) = \max_{s'}^* \left[\bar{\alpha}_{t-1}(s') + \bar{\gamma}_t(s', s) \right]$$

$$\bar{\beta}_{t-1}(s') = \max_s^* \left[\bar{\gamma}_t(s', s) + \bar{\beta}_t(s) \right]$$

$$\bar{\gamma}_t(s', s) = c_t' \frac{L_a(c_t)}{2} + \frac{L_c}{2} \sum_{i=0}^{n-1} r_{ti} x_{ti}$$

$$\max^*(\delta_1, \delta_2) = \max(\delta_1, \delta_2) + \ln \left(1 + e^{-|\delta_2 - \delta_1|} \right)$$

$$L(c_t | \mathbf{r}) = L_a(c_t) + L_c r_{t0} + L_e(c_t)$$

La entrada al primer decodificador es la secuencia de datos $\mathbf{r}' = [0,8 \ 0,1 ; 1 \ -0,5 ; -1,8 \ 1,1 ; 1,6 \ -1,6]$.

El canal tiene una SNR caracterizada por $L_c = 2$.

La información a priori es: $\mathbf{L}_a = [-0,4 ; -0,07 ; -0,8 ; 2,03]$.

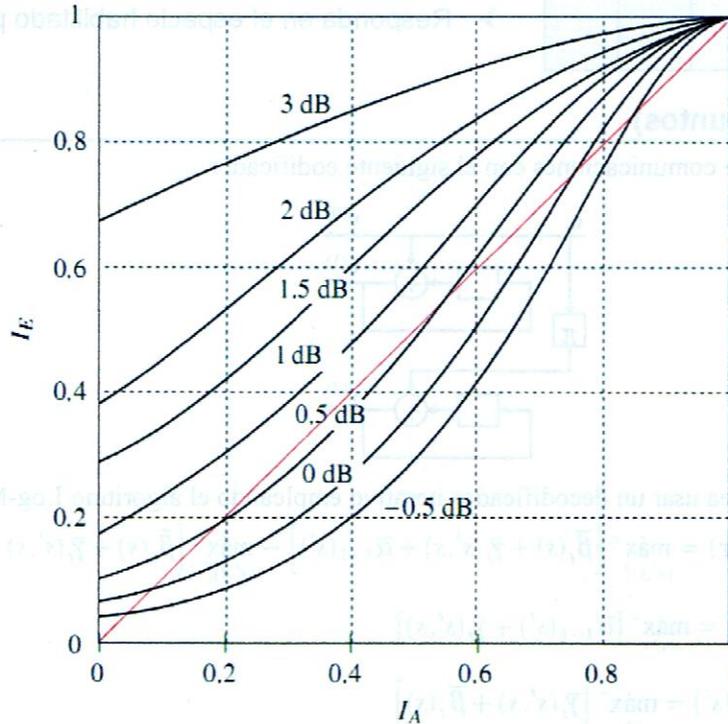
Se pide calcular la información extrínseca del bit en $t = 2$.

Contestar en HOJA aparte.

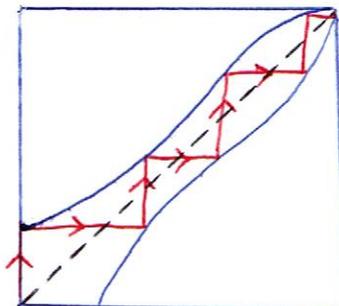


Cuestión 2 (1 punto)

Dada la siguiente figura en la que se representa la función EXIT que relaciona la información mutua a priori y la información mutua extrínseca para diferentes valores de E_b/N_0 . Indique para qué valores de relación señal ruido de las representadas en la figura el turbo decodificador convergerá (Razone su respuesta).



Puesto que los dos decodificadores de un TC hacen uso de la información ~~extrínseca~~ que extrínseca del otro como info a priori suya, se puede predecir la evolución de la información que se pasa en sucesivas iteraciones 'reflejando' la gráfica, para representar al otro codificador respecto a la línea $I_E = I_A$ y comprobar si el "túnel está abierto"

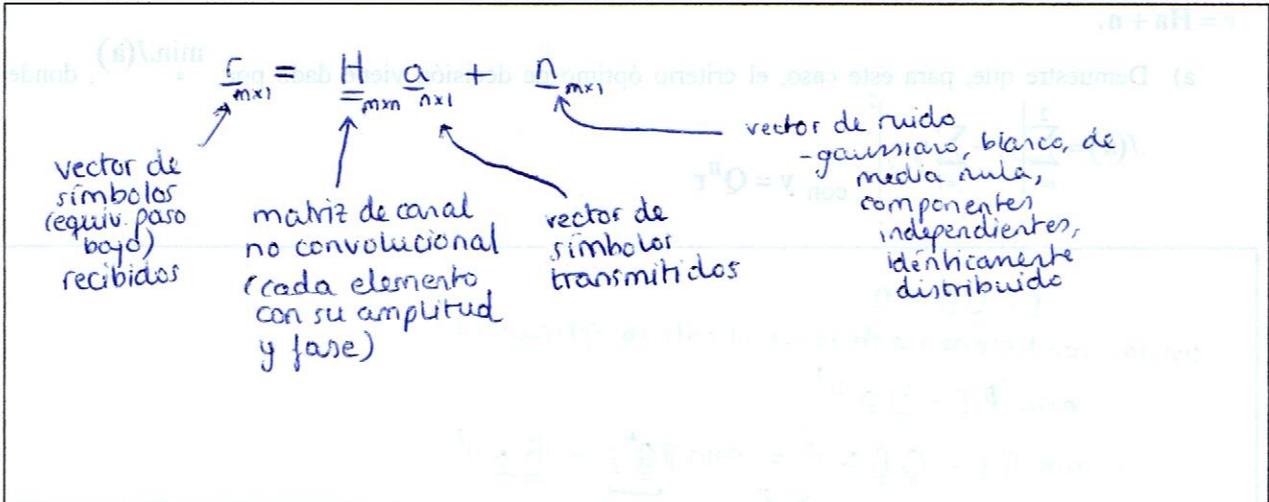


En las gráficas de la figura, el túnel está abierto para algún valor de E_b/N_0 entre 0.5 y 1 dB

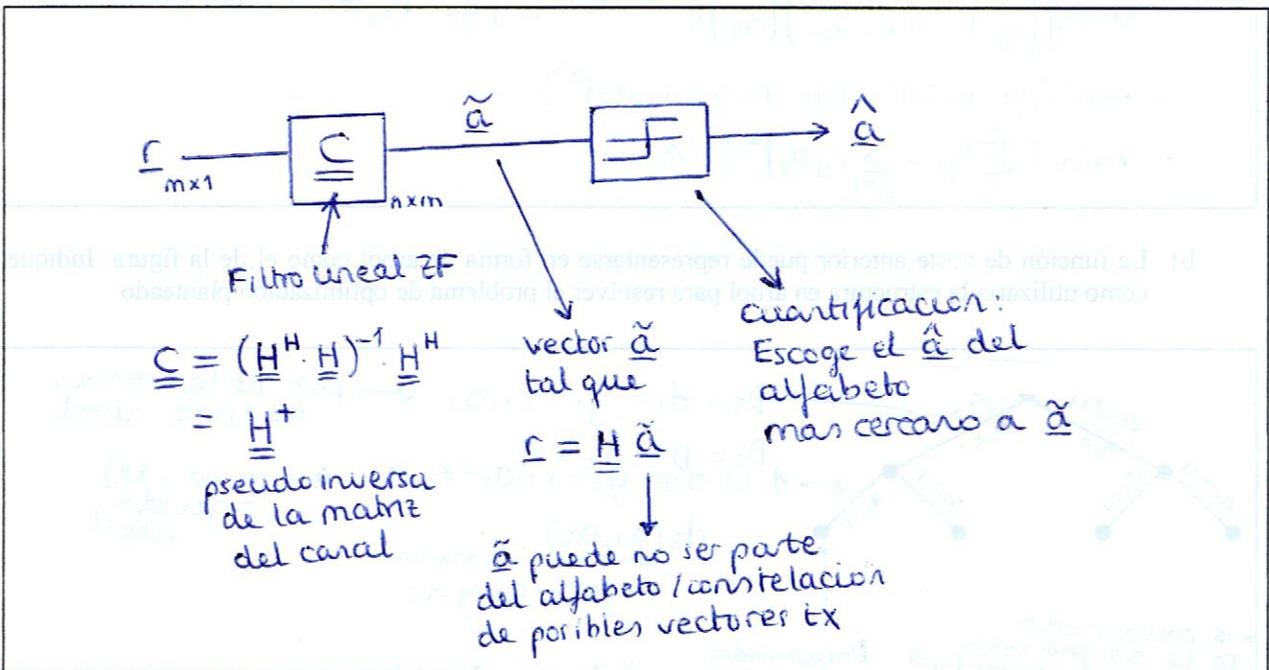
Cuestión 3 (2 puntos)

El modelo de sistema MIMO de comunicaciones digitales sobre canales gaussianos sin memoria se caracteriza por la siguiente relación en recepción: $\mathbf{r} = \mathbf{H}\mathbf{a} + \mathbf{n}$. Responda razonadamente a las siguientes cuestiones.

- a) Indique qué representan y cuáles son los modelos típicos que se utilizan para la matriz \mathbf{H} y los vectores \mathbf{a} y \mathbf{n} .



- b) Dibuje el esquema de un receptor lineal “forzador de ceros” (ZF) cuya entrada fuera \mathbf{r} y su salida una decisión sobre el vector \mathbf{a} transmitido. Explique el funcionamiento del detector dibujado. ¿Por qué se llama forzador de ceros?”



Cuestión 4 (2 puntos)

Se transmite un vector $\mathbf{a} = [a_1 \ a_2]^T$ por un canal de comunicaciones MIMO 2x2 sin memoria gaussiano, con matriz del canal dada por $\mathbf{H} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} r_{11} & 0 \\ r_{12} & r_{22} \end{bmatrix}$, donde \mathbf{Q} es una matriz unitaria y los elementos de la diagonal de \mathbf{R} son reales y positivos. Los símbolos $a_i \in \{1, -1\}$. Como en el ejercicio anterior, sea $\mathbf{r} = \mathbf{H}\mathbf{a} + \mathbf{n}$.

a) Demuestre que, para este caso, el criterio óptimo de decisión viene dado por: $\min_{\mathbf{a}} J(\mathbf{a})$, donde

$$J(\mathbf{a}) = \sum_{i=1}^2 \left| y_i - \sum_{j=1}^2 r_{ij} a_j \right|^2, \text{ con } \mathbf{y} = \mathbf{Q}^H \mathbf{r}$$

$\mathbf{r} = \mathbf{H}\mathbf{a} + \mathbf{n}$

con las condiciones anteriores, el criterio óptimo es:

$$\min \|\mathbf{r} - \mathbf{H}\mathbf{a}\|^2$$

$$= \min \|\mathbf{r} - \mathbf{QR}\mathbf{a}\|^2 = \min \|\underbrace{\mathbf{Q}^H}_{\text{unitaria}} \mathbf{r} - \underbrace{\mathbf{R}}_{\mathbf{Q}^H \mathbf{Q} = \mathbf{I}} \mathbf{a}\|^2$$

(mult. x \mathbf{Q}^H)

$$= \min \left\| \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} r_{11} & 0 \\ r_{12} & r_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \right\|^2 = \min \left\| \begin{bmatrix} y_1 - r_{11}a_1 \\ y_2 - r_{12}a_1 - r_{22}a_2 \end{bmatrix} \right\|^2$$

$$= \min (|y_1 - r_{11}a_1|^2 + |y_2 - r_{12}a_1 - r_{22}a_2|^2)$$

$$= \min \left(\sum_{i=1}^2 \left| y_i - \sum_{j=1}^2 r_{ij} a_j \right|^2 \right) //$$

b) La función de coste anterior puede representarse en forma de árbol como el de la figura. Indique cómo utilizaría la estructura en árbol para resolver el problema de optimización planteado:

$d_1(a_1) = y_1 - r_{11}a_1$ ← pero de las ramas del primer nivel

$d_2(a_1, a_2) = y_2 - r_{12}a_1 - r_{22}a_2$ ← ramas del segundo nivel

en cada camino según a_1 y a_2

• a cada nodo se le asigna una distancia acumulada igual a la suma de pesos de cada rama que lo usa al irse a raíz.

• Se van podando las ramas cuya distancia acumulada

Dacumulada
 $= d_1(a_1=1) + d_2(a_1=1, a_2=1)$
 $= y_1 - r_{11} + y_2 - r_{12} - r_{22}$

Cuestión 5 (1,5 puntos)

Cuestión 5 (1,5 puntos)

Sea una matriz \mathbf{X} que define el siguiente código bloque espacio-temporal (STBC):

$$\mathbf{X}(s) = \begin{bmatrix} s_0 & s_1 & -s_2^* & -s_3^* \\ s_1 & s_0 & -s_3^* & -s_2^* \\ s_2 & s_3 & s_0^* & s_1^* \\ s_3 & s_2 & s_1^* & s_0^* \end{bmatrix}$$

a) ¿Es un código ortogonal? **Justifique** su respuesta

Responda **razonadamente** a las siguientes cuestiones:

b) ¿Cuántas antenas transmisoras necesita?

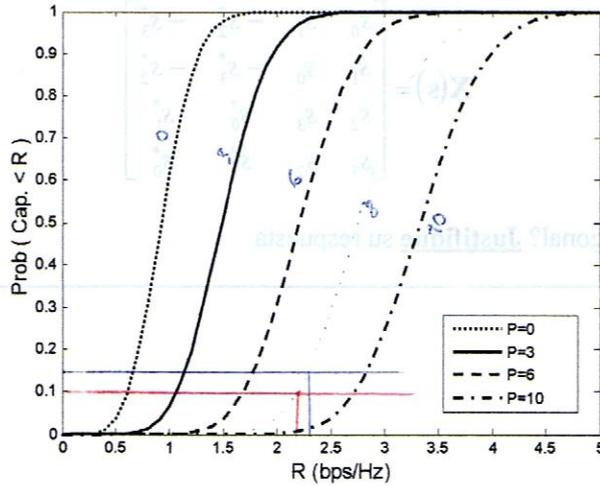
c) ¿Cuál es la tasa del código?

En 4 instantes ha transmitido en total 4 símbolos, por tanto
 $R = 1 \text{ símbolo/s/Hz}$

d) Defina la Ganancia por Codificación y la Ganancia por Diversidad de un código espacio-temporal.

Cuestión 6 (1 punto)

La figura representa la Probabilidad de Outage de un sistema MIMO para distintas potencias de transmisión P (en dBm):



a) Defina la Probabilidad de Outage:

Outage es la probabilidad de que durante el tiempo de transmisión de un símbolo, la capacidad de un canal no ergódico este por debajo de un cierto umbral ($Cap < R$)

Si se tiene una restricción de potencia en el transmisor $P \leq 8$ dBm :

b) ¿Cuál es la mayor tasa que se puede conseguir para una probabilidad de outage igual o menor al 10%? Justifique su respuesta.

$$R < 2.2 \text{ bps/Hz}$$

c) Si la aplicación necesita una tasa de transmisión de 3 bps/Hz y una probabilidad de outage inferior al 15% ¿Es posible conseguirlo con el sistema MIMO aquí considerado? Justifique su respuesta.

No