

# Comunicaciones por Satélite Curso 2008-09

## Codificación de canal

Codificación convolucional  
Entrelazado

Esquemas de codificación avanzados

**Miguel Calvo Ramón**  
**Ramón Martínez Rodríguez-Osorio**

# Índice

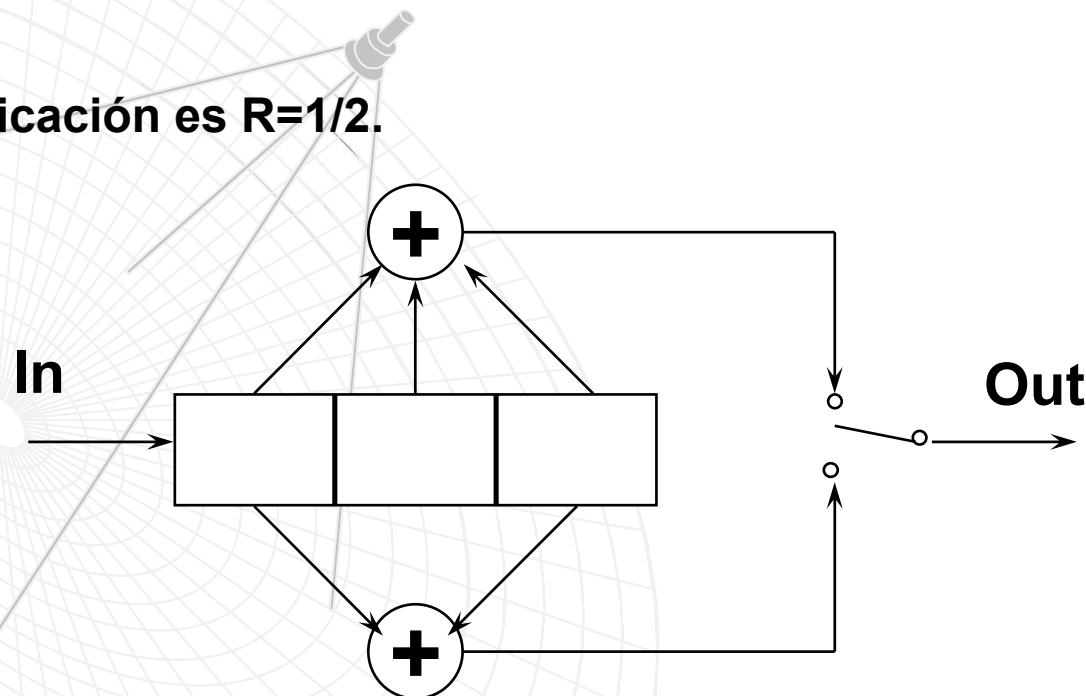
- **Introducción. Concepto y objetivo de la codificación de canal**
- **Sistemas basados en ARQ**
- **Modelos de canal. Decisión dura vs. Decisión blanda**
- **Capacidad del canal. Límite de Shannon**
- **Esquemas FEC**
  - Parámetros característicos de un código
  - Códigos de bloques. Códigos de Hamming, Golay, BCH, Reed-Solomon
  - Códigos convolucionales. Decodificador de Viterbi. Ejemplo
  - Aspectos de implementación
- **Entrelazado**
- **Modulación codificada (TCM)**
- **Turbocódigos**
- **Códigos LDPC**

# Codificación Convolutiva

Este tipo de FEC se usa ampliamente por la disponibilidad de algoritmos de decodificación soft de fácil implementación y que proporcionan ganancias de codificación grandes.

Considérese un codificador convolutivo como el de la figura que acepta un bit a la entrada y genera dos bits a su salida. Los dos bits de salida se multiplexan en una única secuencia con una velocidad doble que la de la entrada.

La tasa de codificación es  $R=1/2$ .



# Estados

Es claro que los bits de salida  $v_{j,1}$  y  $v_{j,2}$  se pueden expresar en función del bit  $u_j$  de entrada y de los dos anteriores,  $u_{j-1}$  y  $u_{j-2}$ , como:

$$v_{j,1} = u_j \oplus u_{j-1} \oplus u_{j-2}$$

$$v_{j,2} = u_j \oplus u_{j-2}$$

donde  $\oplus$  es la suma módulo 2.

El **estado del codificador** se define por el contenido de su registro de desplazamiento excluyendo el último elemento que se pierde al entrar el siguiente bit a codificar.

Sea  $N_m$  la memoria total del codificador, es decir la suma de todas las longitudes de los registros. Hay entonces  $2^{N_m}$  posibles estados.

Para el codificador anterior  $N_m = M = 2$ ; por tanto hay 4 posibles estados 00, 01, 10 y 11.

Se necesita el estado de los dos primeros bits del registro y el bit de entrada, es decir, un total de  $K=3$  bits, para determinar los dos bits de salida.

$K$  se denomina la **longitud de restricción** del código convolucional y es igual a la suma de los bits de estado más los bits de entrada.

# Árbol del Código

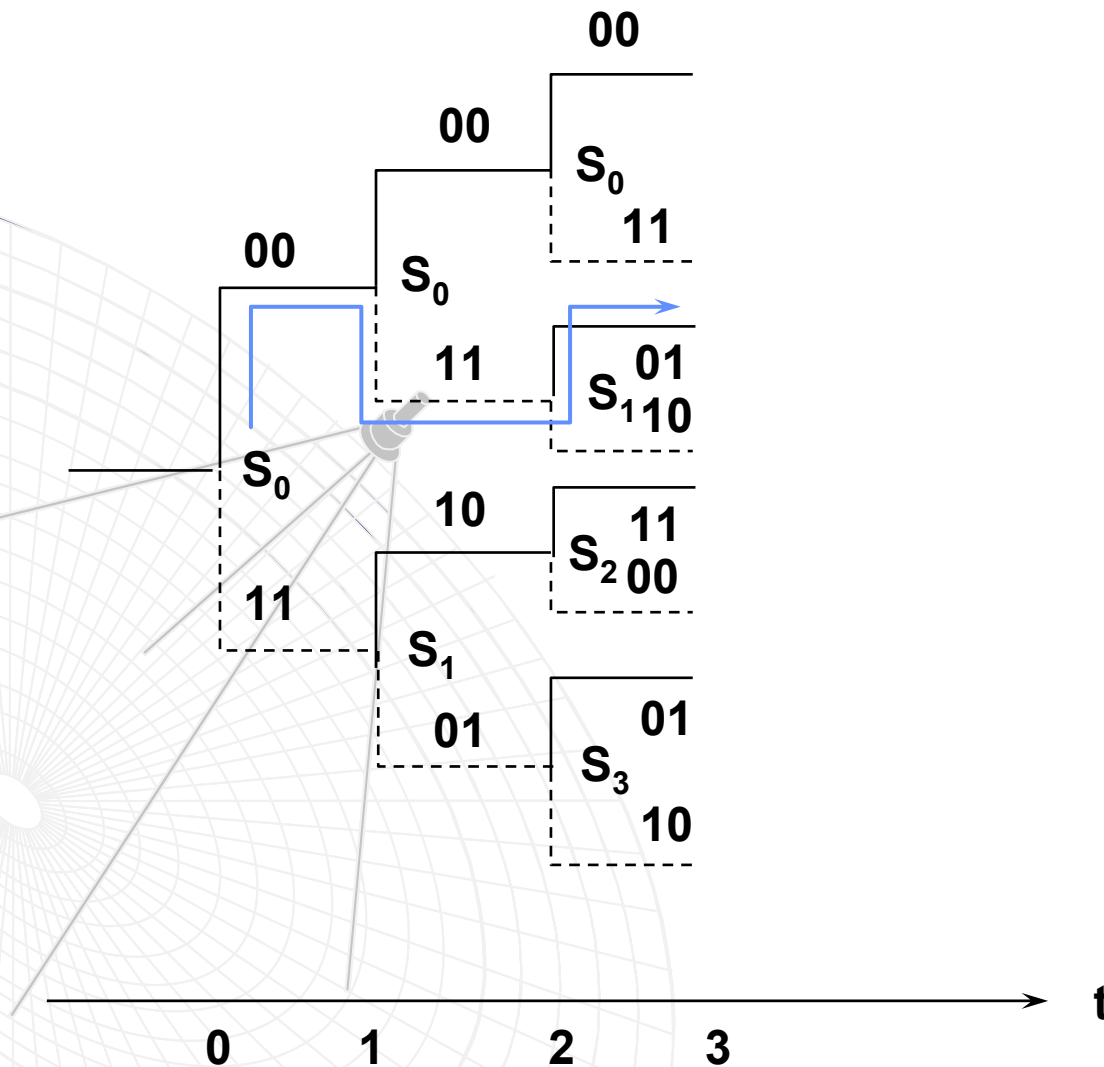
El tipo de relación entre las secuencias de entrada y salida de un codificador convolucional se puede representar gráficamente mediante un *árbol de código*.

La figura muestra el árbol de código para el codificador de ejemplo. Las ramas con línea continua representan una entrada 0 al codificador y las ramas con línea discontinua una entrada 1. Los nodos correspondientes a los estados 00, 01, 10 y 11 se denominan  $S_0$ ,  $S_1$ ,  $S_2$  y  $S_3$ , respectivamente.

Dado que la llegada de un nuevo bloque de  $k$  bits produce una transición a un nuevo estado hay  $2^k$  ramas de  $n$  bits dejando cada estado, cada uno correspondiente a cada bloque de entrada diferente.

Para nuestro codificador  $k=1$ , y por tanto hay 2 ramas de dos bits a la salida de cada estado.

----- In: 1





# Árbol de Código

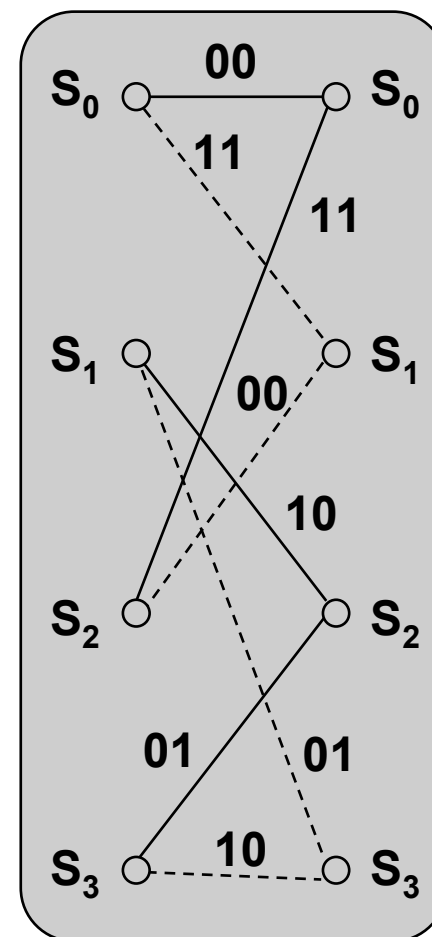
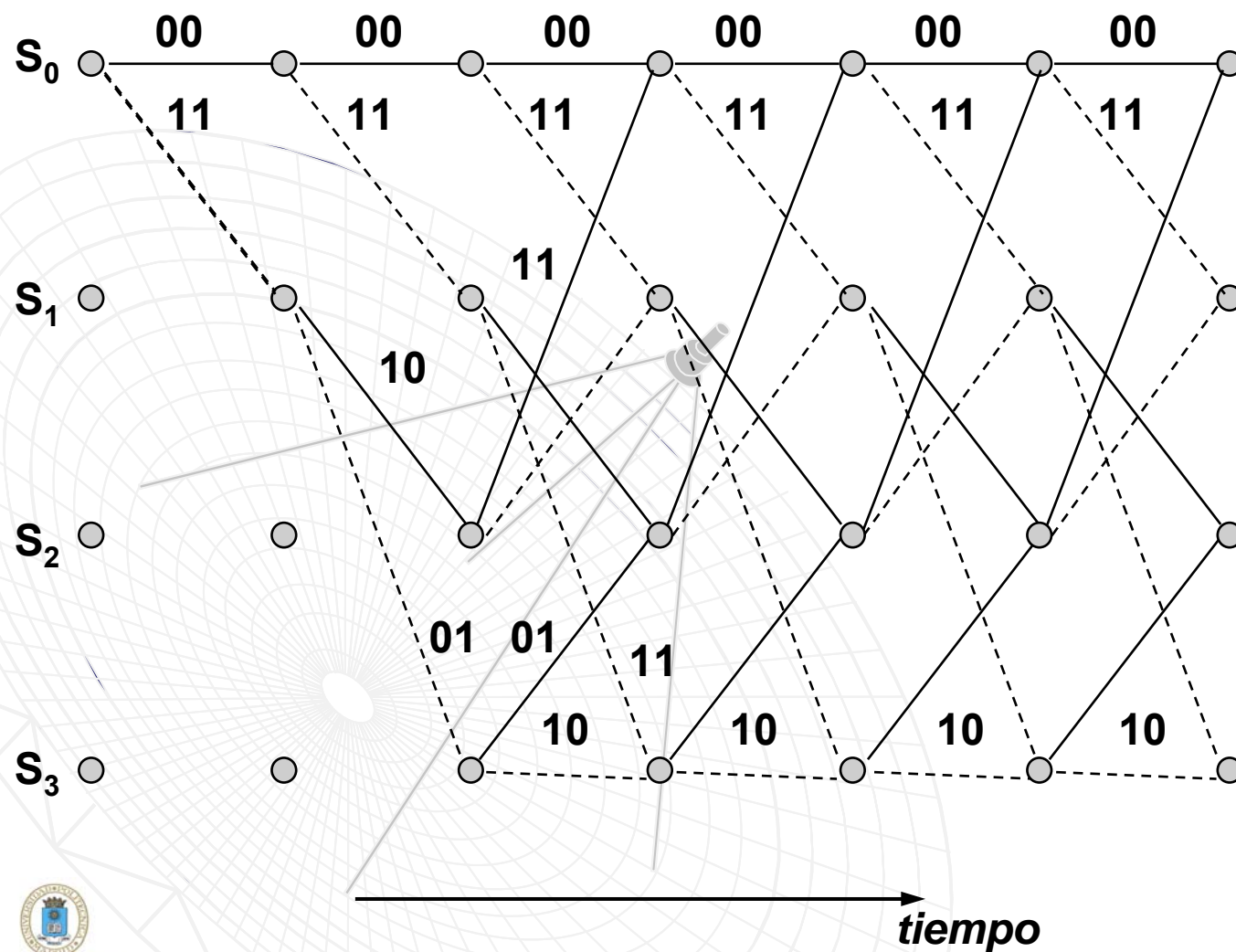
Una secuencia de salida del codificador se determina trazando una trayectoria a través del árbol de código, de izquierda a derecha, en función de cuál haya sido la secuencia de entrada.

La figura anterior muestra la trayectoria correspondiente a una secuencia de entrada 1010, que producirá la secuencia de salida 00 11 10 00.

La estructura de ramas se repite a partir del instante  $j=3$  (ya que  $M+1=3$ ). Por tanto, los conjuntos de trayectos que se originan en nodos que tienen el mismo estado son idénticos.

Esta observación conduce al *diagrama de trellis* en la que los trayectos iguales que se originan en nodos con el mismo estado se juntan en una misma rama.

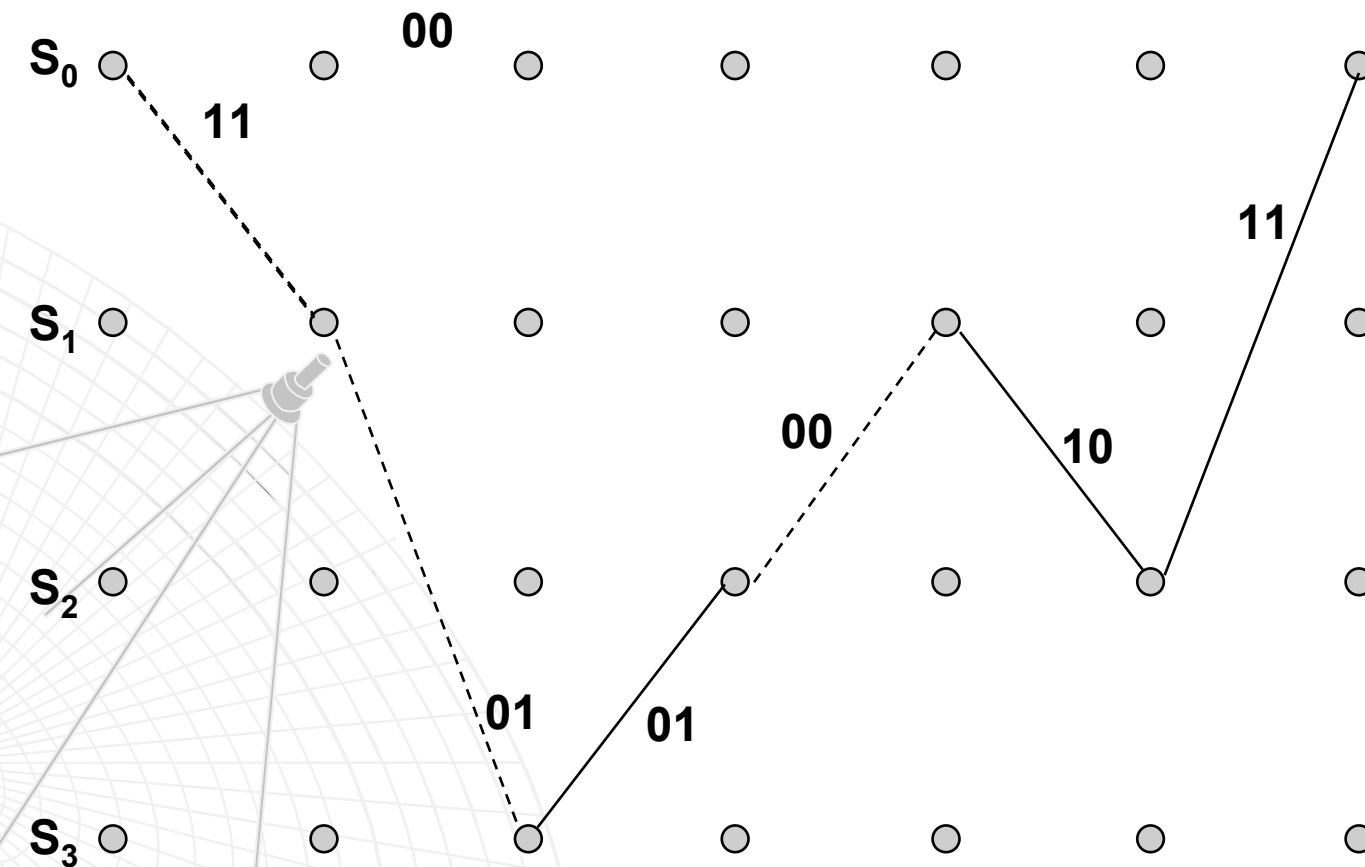
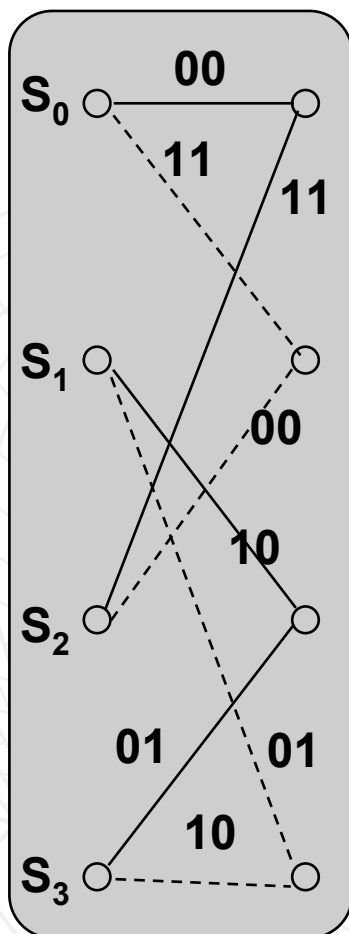
# Diagrama de Trellis





# Ejemplo de Codificación

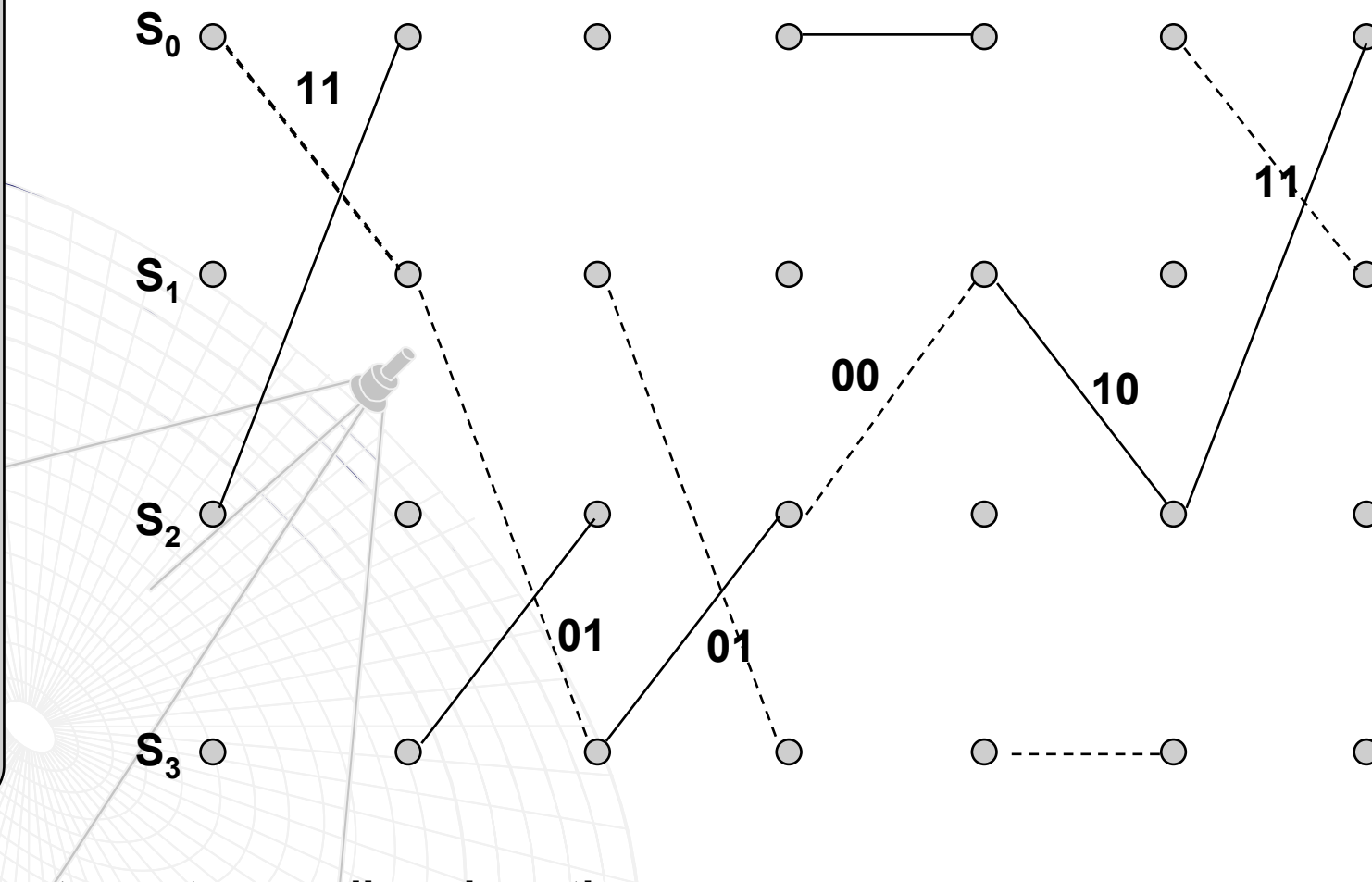
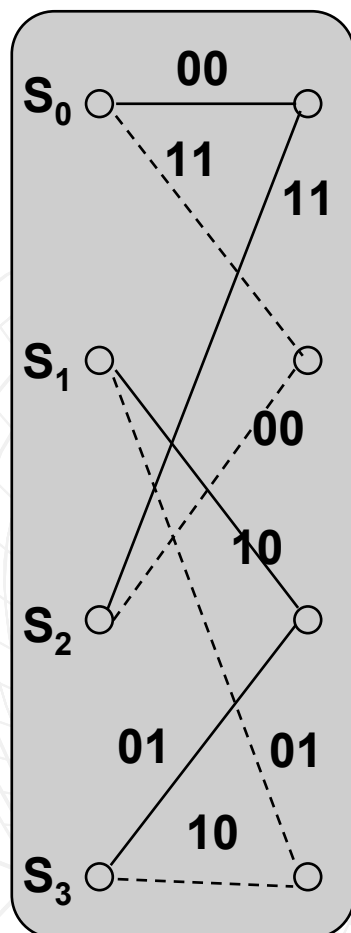
m: 110100



v= 11 01 01 00 10 11

# Decodificación sin Errores

$R = 11\ 01\ 01\ 00\ 10\ 11$



De los posibles trayectos se elige el continuo.

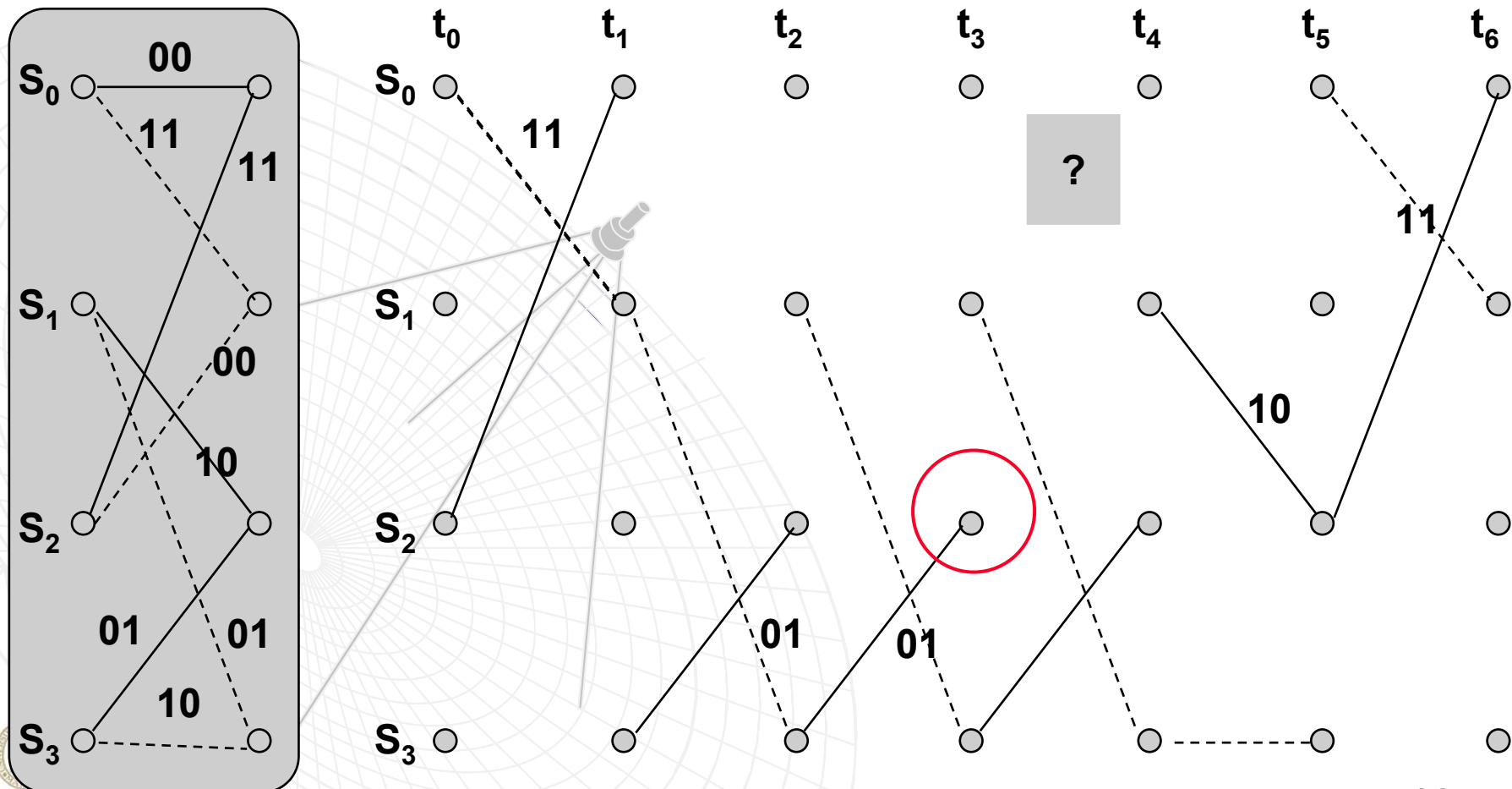
En caso de error se pierde la continuidad.



# Decodificación de mensajes con errores

Supongamos el mensaje transmitido  $V = 11\ 01\ 01\ 00\ 10\ 11$ , que se ha recibido con errores como:  $R = 11\ 01\ 01\ 0\underline{1}\ 10\ 11$ .

Los posibles trayectos en el Trellis son ahora:



# Decodificación de Viterbi

La decodificación de un código convolucional puede hacerse con varios algoritmos como el de Viterbi, la decodificación secuencial y la decodificación con síndrome.

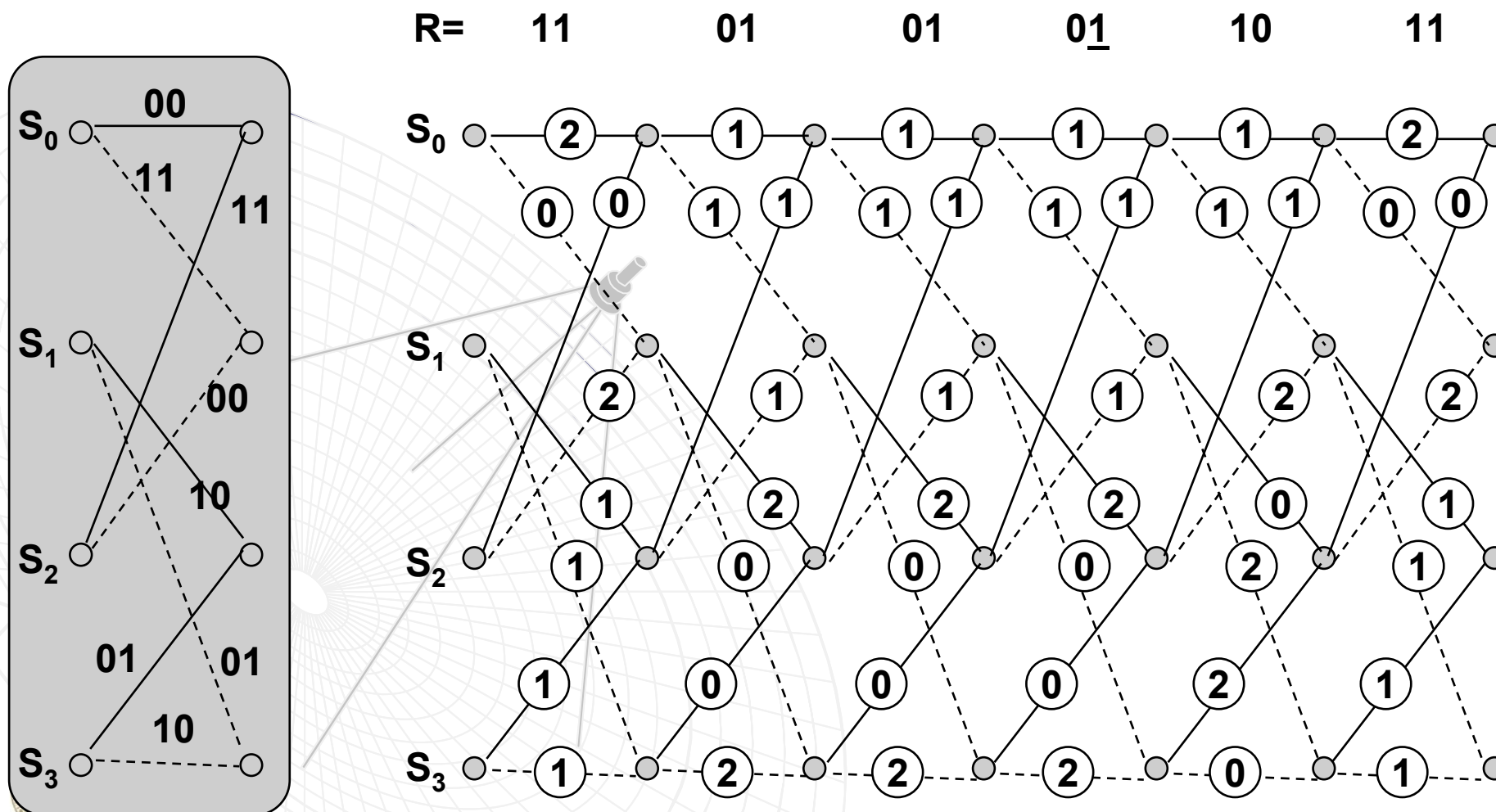
La técnica de decodificación más conocida es el algoritmo de Viterbi que es una aplicación del principio de *máxima verosimilitud* aplicado a decodificación.

El algoritmo de Viterbi selecciona el código que proporciona un mayor valor de la métrica de verosimilitud que es equivalente a un menor valor de la métrica de distancia. Esta es la distancia de Hamming de cada rama respecto a la palabra recibida. Cada rama se etiqueta con su distancia.

El algoritmo procesa la secuencia recibida de forma iterativa comparando los trayectos que entran en cada nodo y reteniendo, junto con su métrica, el trayecto superviviente que es el que tiene un distancia menor.

# Distancia de Hamming

En el trellis del decodificador etiquetamos cada rama, en el intervalo  $t_i$ , con la distancia de Hamming entre la palabra de la rama y la palabra recibida en  $t_i$ .



# Algoritmo de Viterbi

1. Cuando se reciben los símbolos de código cada rama del trellis se etiqueta con la métrica de similitud (distancia de Hamming) entre el símbolo recibido y el símbolo de la rama, en cada intervalo  $t_i$ .
2. El algoritmo de decodificación usa la métrica de la distancia de Hamming para obtener el trayecto más verosímil (menor distancia) a través del trellis.
3. Si dos trayectos a través del trellis se juntan en un estado, el de mayor métrica acumulada puede desecharse al buscar el trayecto óptimo. Ello se debe a que el estado actual resume la historia de todos los estados anteriores en cuanto a su influencia en los estados posteriores.
4. Un código de longitud restringida  $K$  tiene un trellis con  $2^{K-1}$  estados en cada intervalo  $t_i$ . A cada estado entran dos trayectos. La decodificación de Viterbi consiste en calcular las métricas de los dos trayectos que entran en cada estado y eliminar el de mayor métrica. Ello se hace en los  $2^{K-1}$  estados, se pasa al intervalo  $t_{i+1}$  y se repite el proceso.
5. Sobrevive una sola rama en cada intervalo cuando se han decodificado unos  $5K$  intervalos posteriores. Ello representa un retardo en la decodificación.



**R=    11        01        01        01        10        11**



**R=    11        01        01        01        10        11**

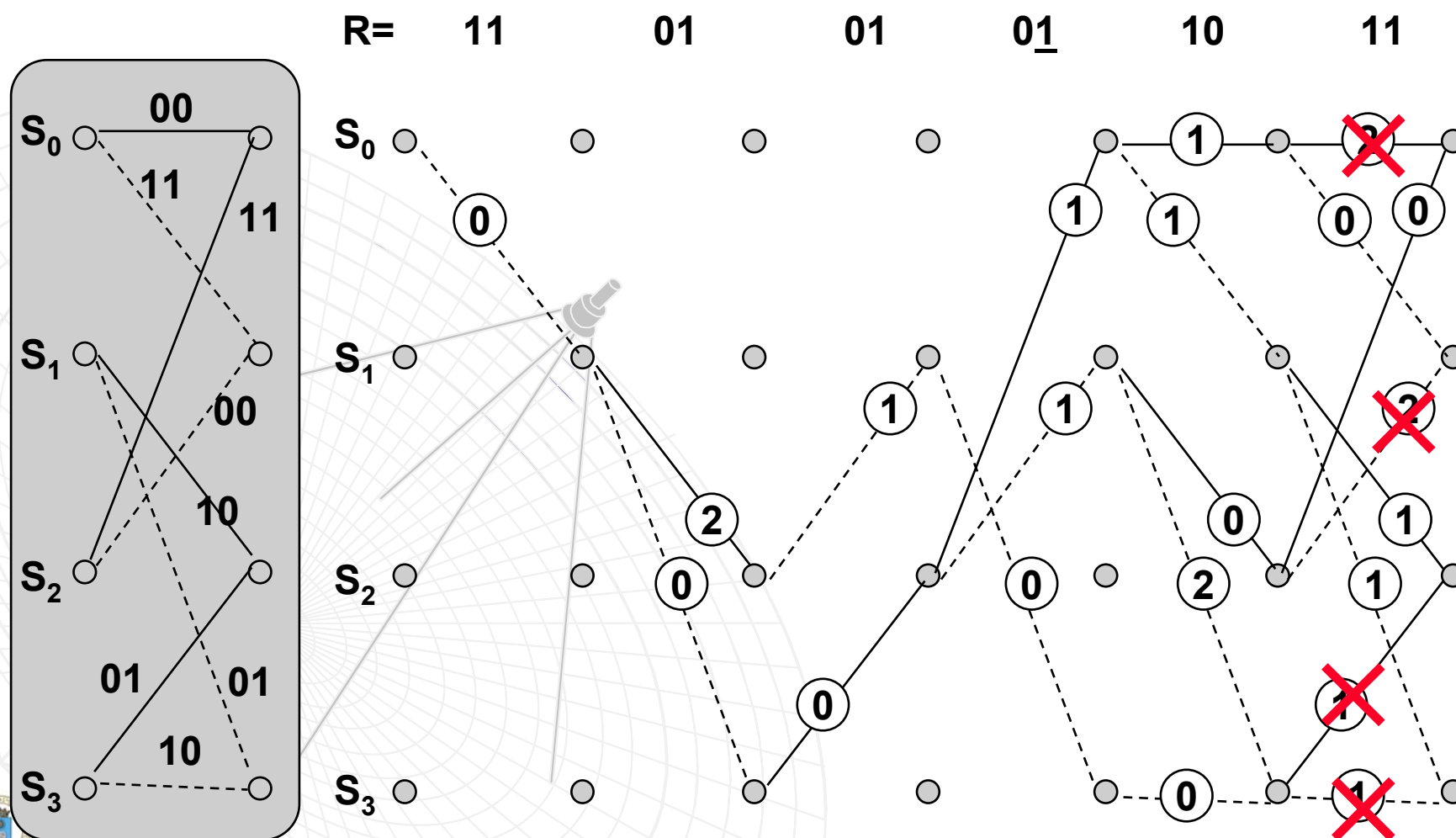


**R=    11        01        01        01        10        11**



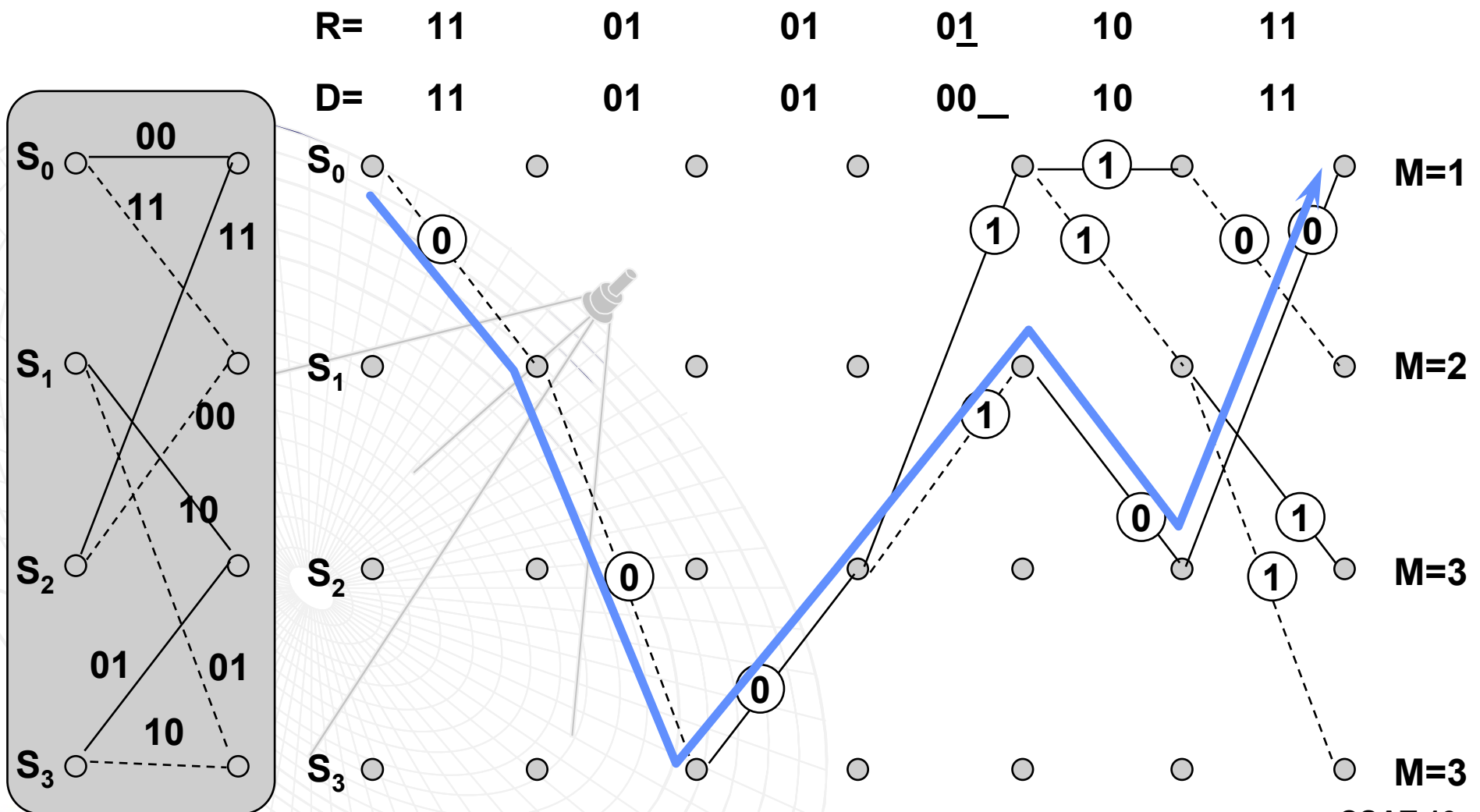
# Ejemplo de Decodificación de Viterbi

Suponiendo conocido que se parte del estado  $S_0$ :



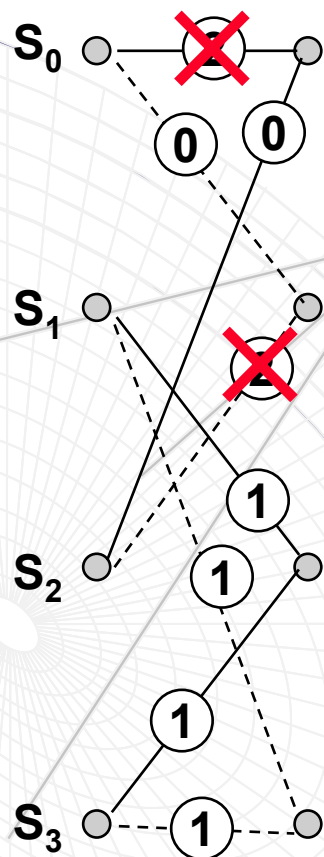
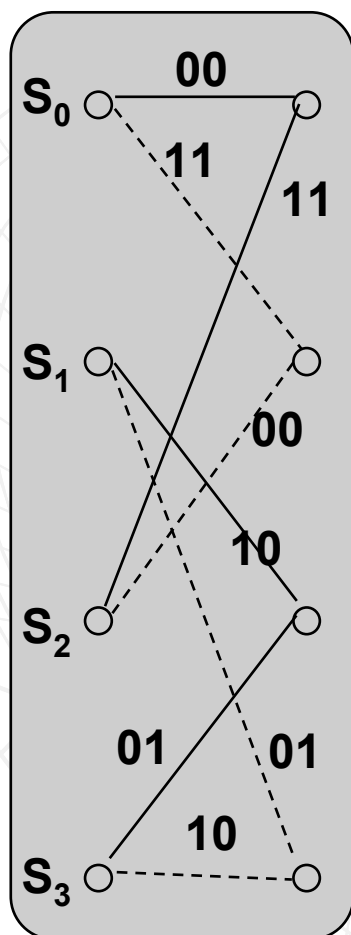
# Ejemplo de Decodificación de Viterbi

Suponiendo conocido que se parte del estado  $S_0$ :



# Ejemplo de Decodificación de Viterbi

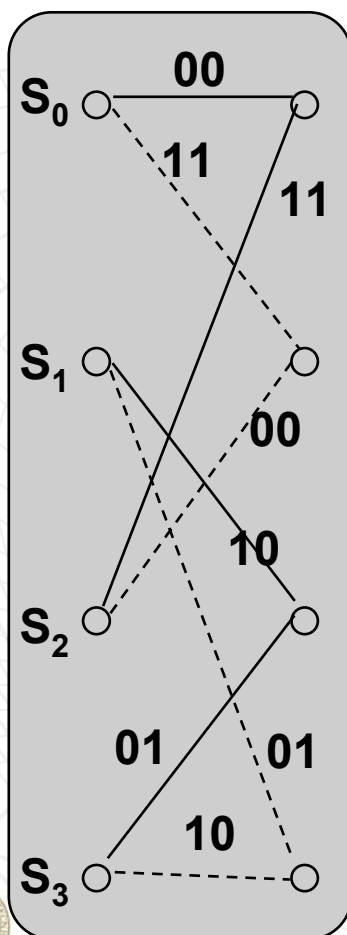
R= 11 01 01 01 10 11



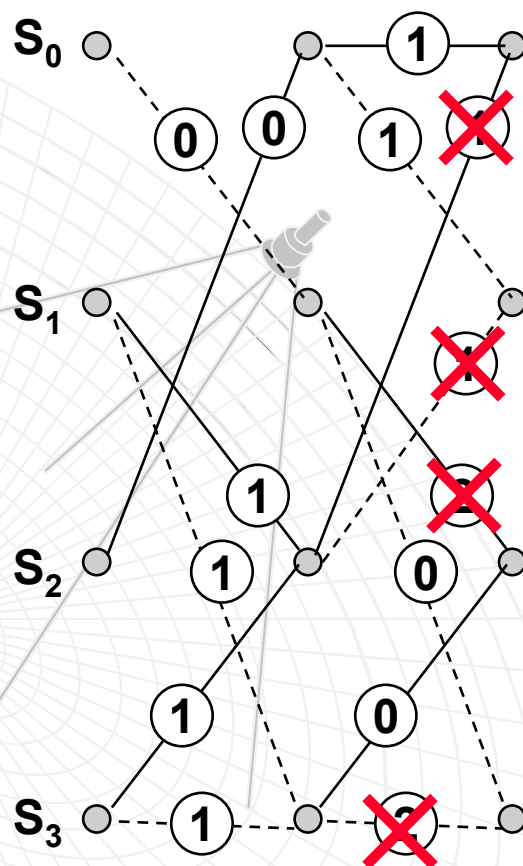




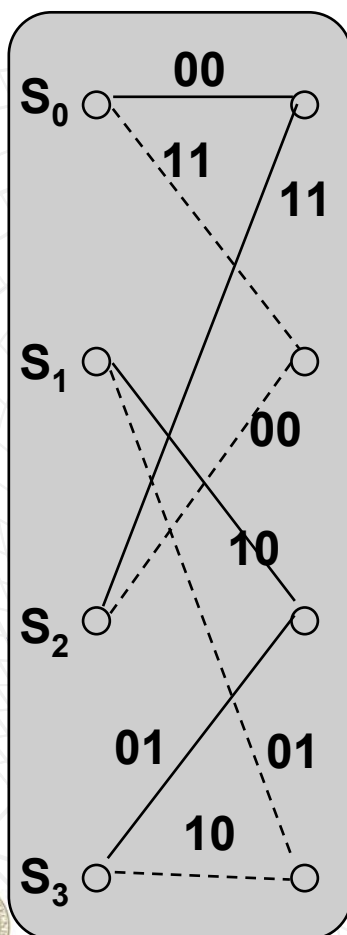
# Ejemplo de Decodificación de Viterbi



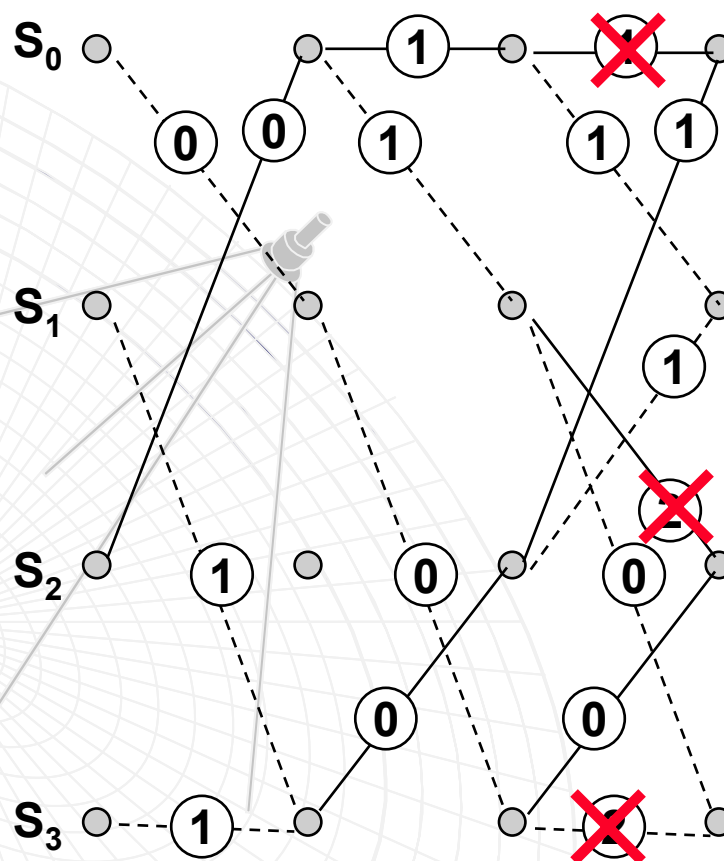
R= 11 01 01 01 10 11



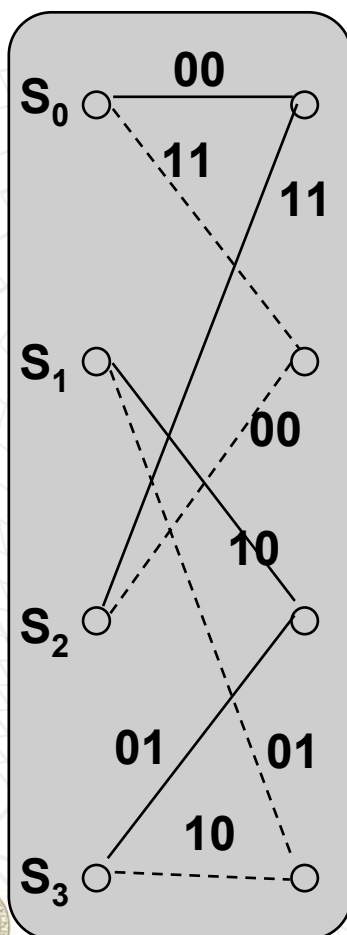
# Ejemplo de Decodificación de Viterbi



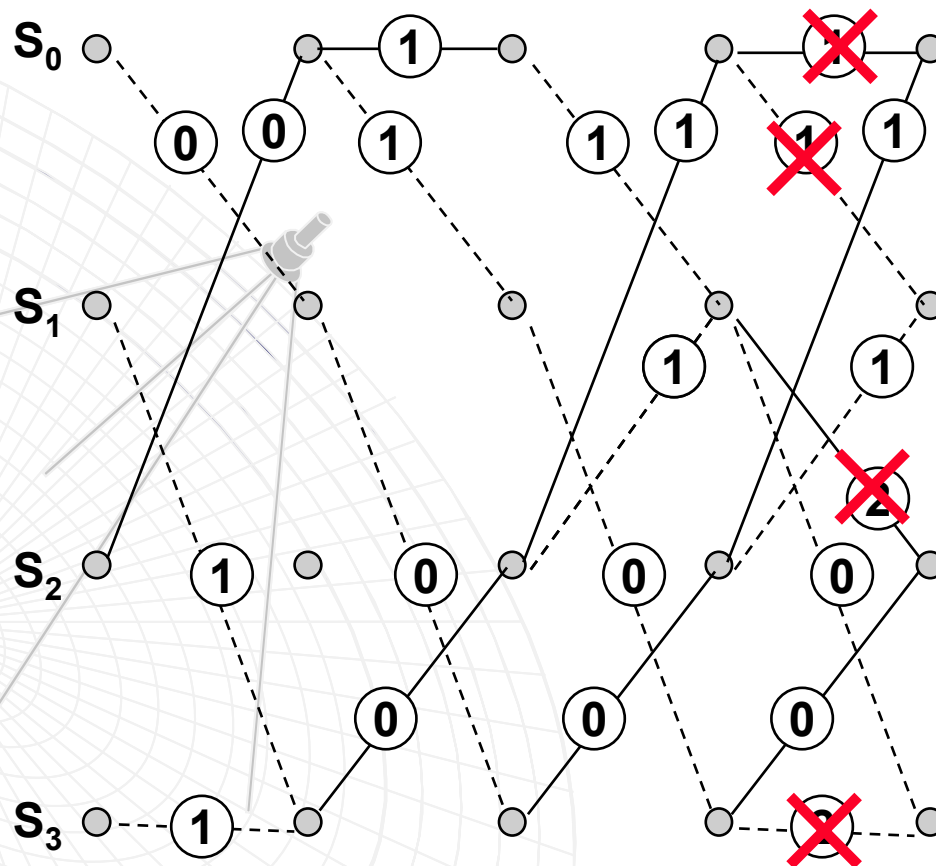
R= 11 01 01 01 10 11



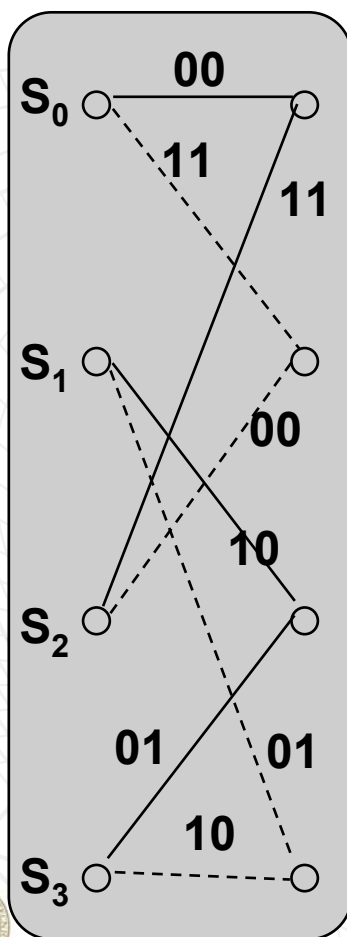
# Ejemplo de Decodificación de Viterbi



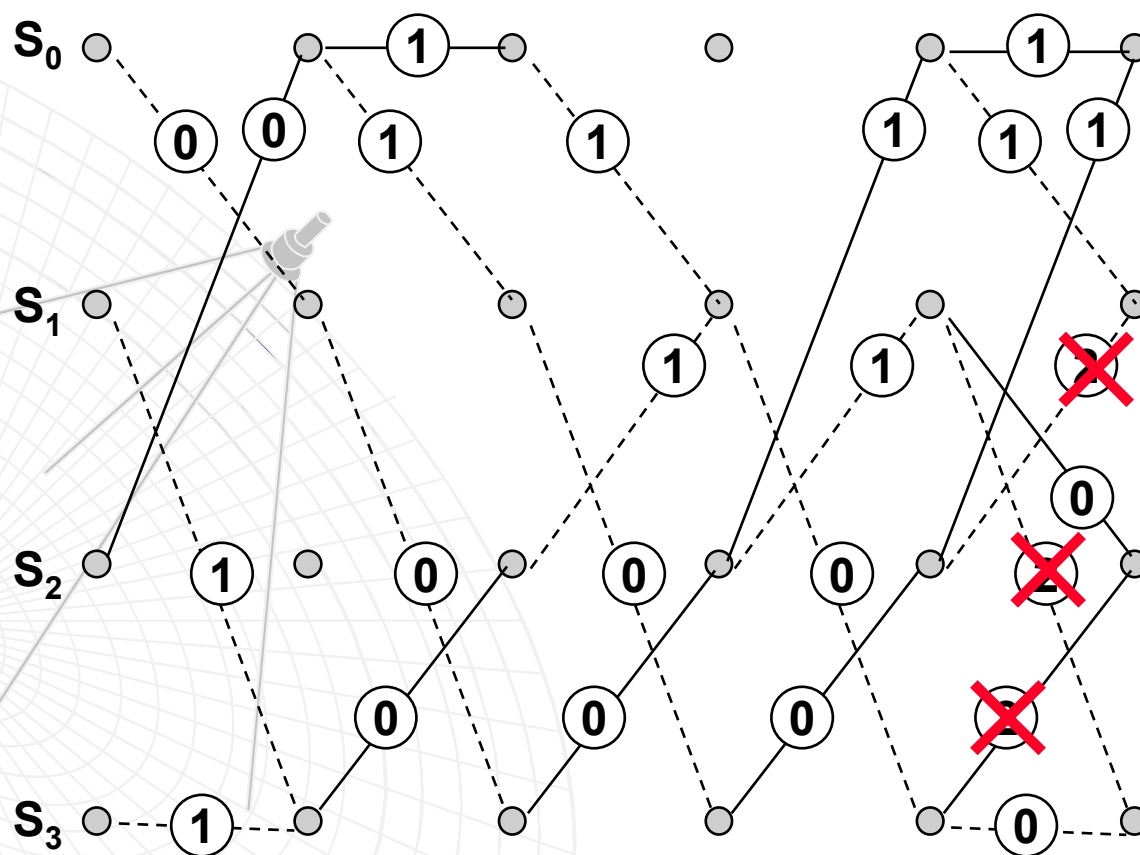
R= 11 01 01 01 10 11



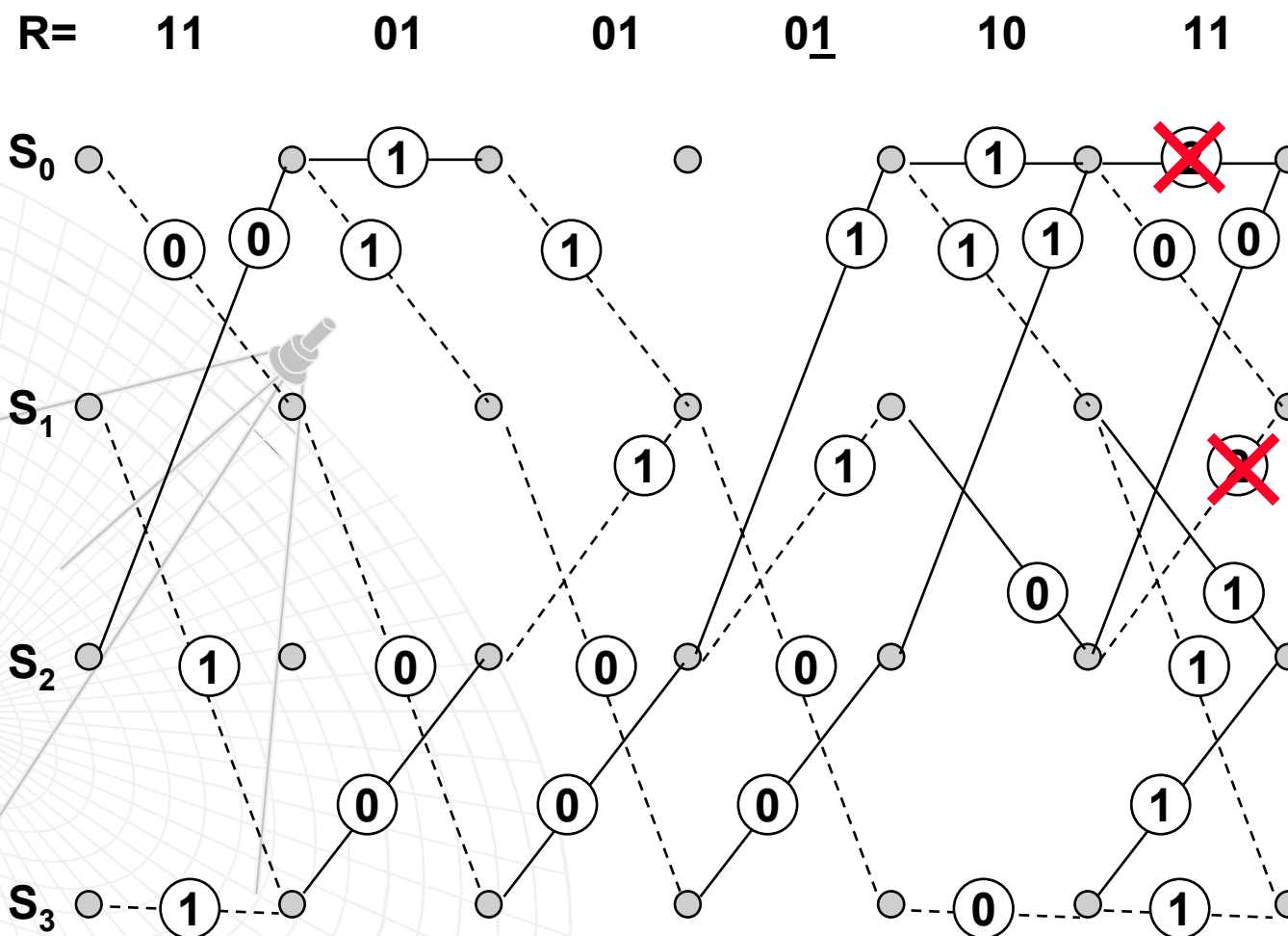
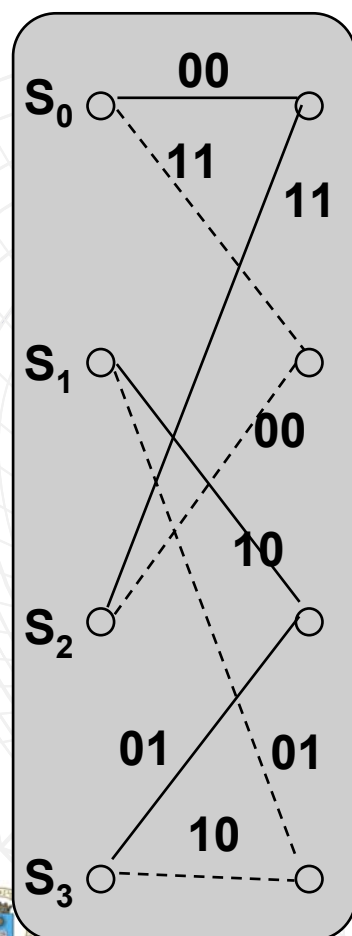
# Ejemplo de Decodificación de Viterbi



R= 11 01 01 01 10 11



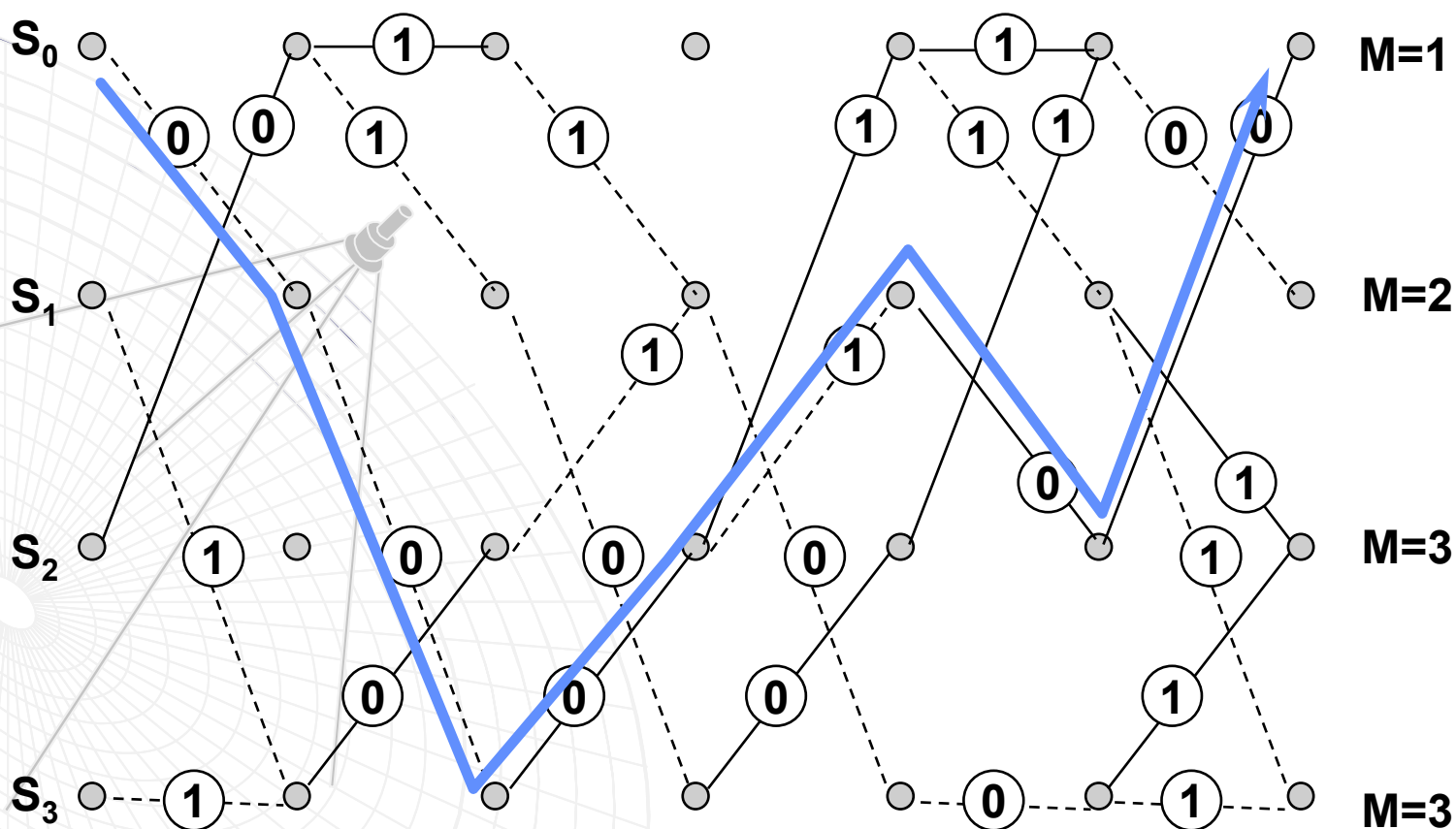
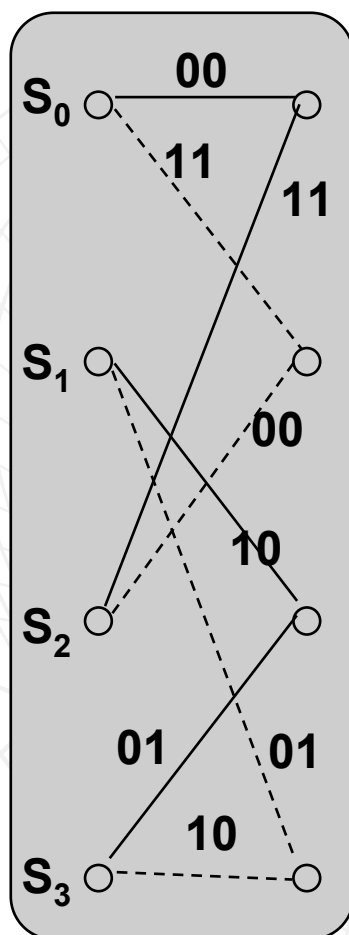
# Ejemplo de Decodificación de Viterbi





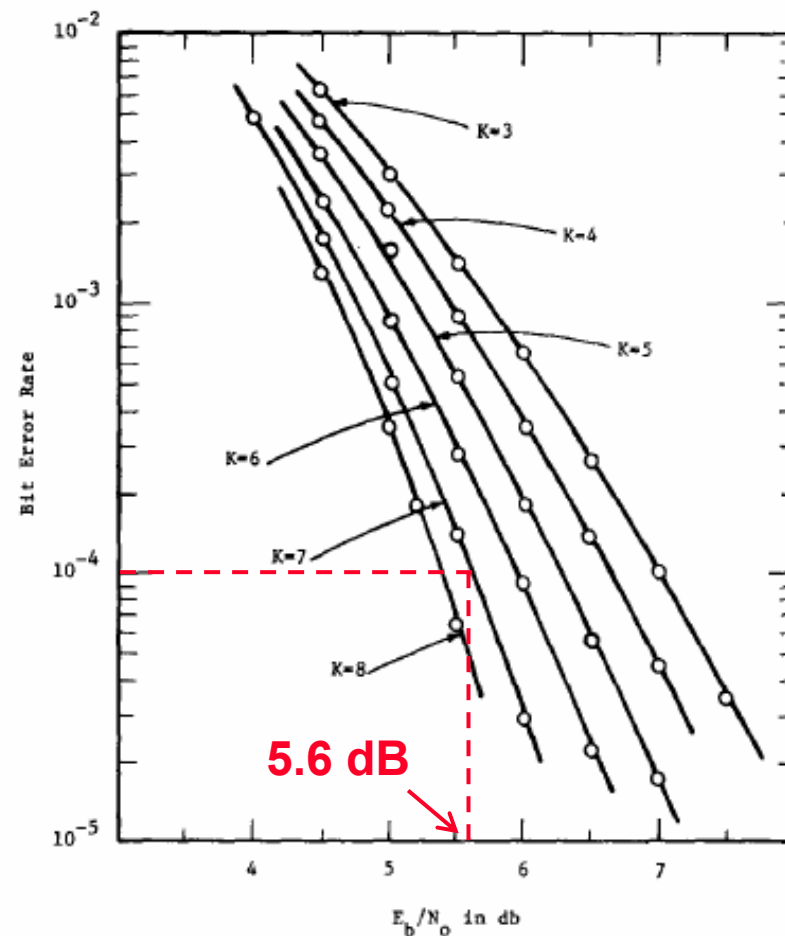
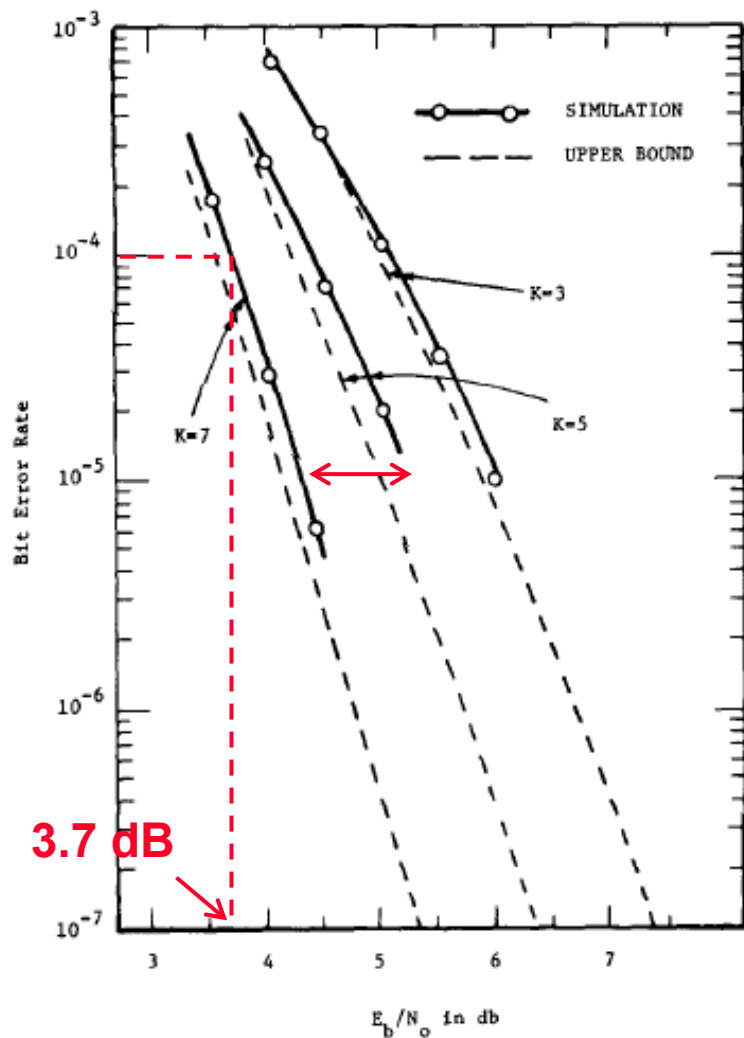
# Ejemplo de Decodificación de Viterbi

R= 11 01 01 01 10 11  
D= 11 01 01 00 10 11



# BER= $f(E_b/N_0)$ para decod. Viterbi

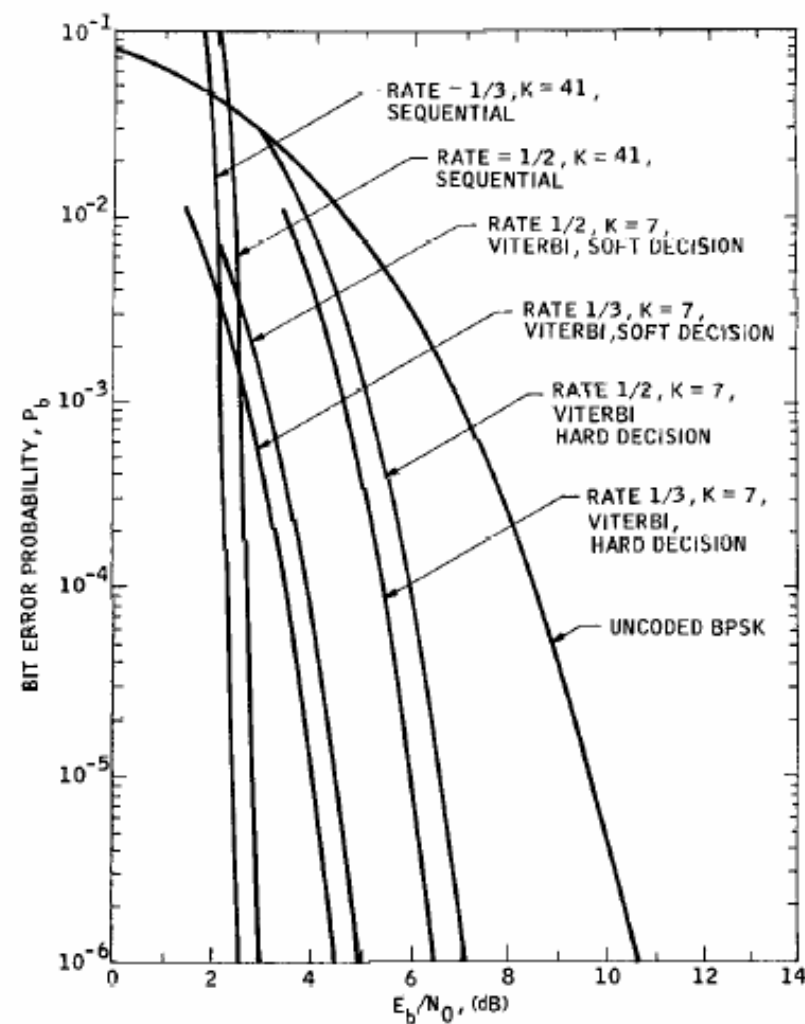
Fuente: [Heller, 71]



# Decodificación secuencial

Fuente: [Heller, 71]

- Hipótesis sobre la secuencia transmitida (*ensayo y error*)
- Avanza por el árbol mientras las métricas indican una secuencia probable
- Si no, vuelve hacia atrás y prueba otras secuencias
- Complejidad independiente de  $K$ , pero dependiente de la SNR
- Problemas de *buffer overflow*



# Decodificación realimentada

- El algoritmo busca en todo el árbol del código
- No toma una decisión hasta que se han recibido  $L$  símbolos en base a la mínima distancia de Hamming acumulada en todas las ramas
- Repite la búsqueda hasta el fin del mensaje
- Cuanto mayor es la ventana temporal  $L$  (*look-ahead length*), se tiene mayor ganancia de codificación, pero mayor es la complejidad

# Propiedades de Distancia

Los códigos convolucionales son lineales y forman un grupo. La **distancia mínima** entre dos secuencias cualesquiera es también la distancia mínima entre una secuencia cualquiera y la secuencia “cero” (todos los bits 0).

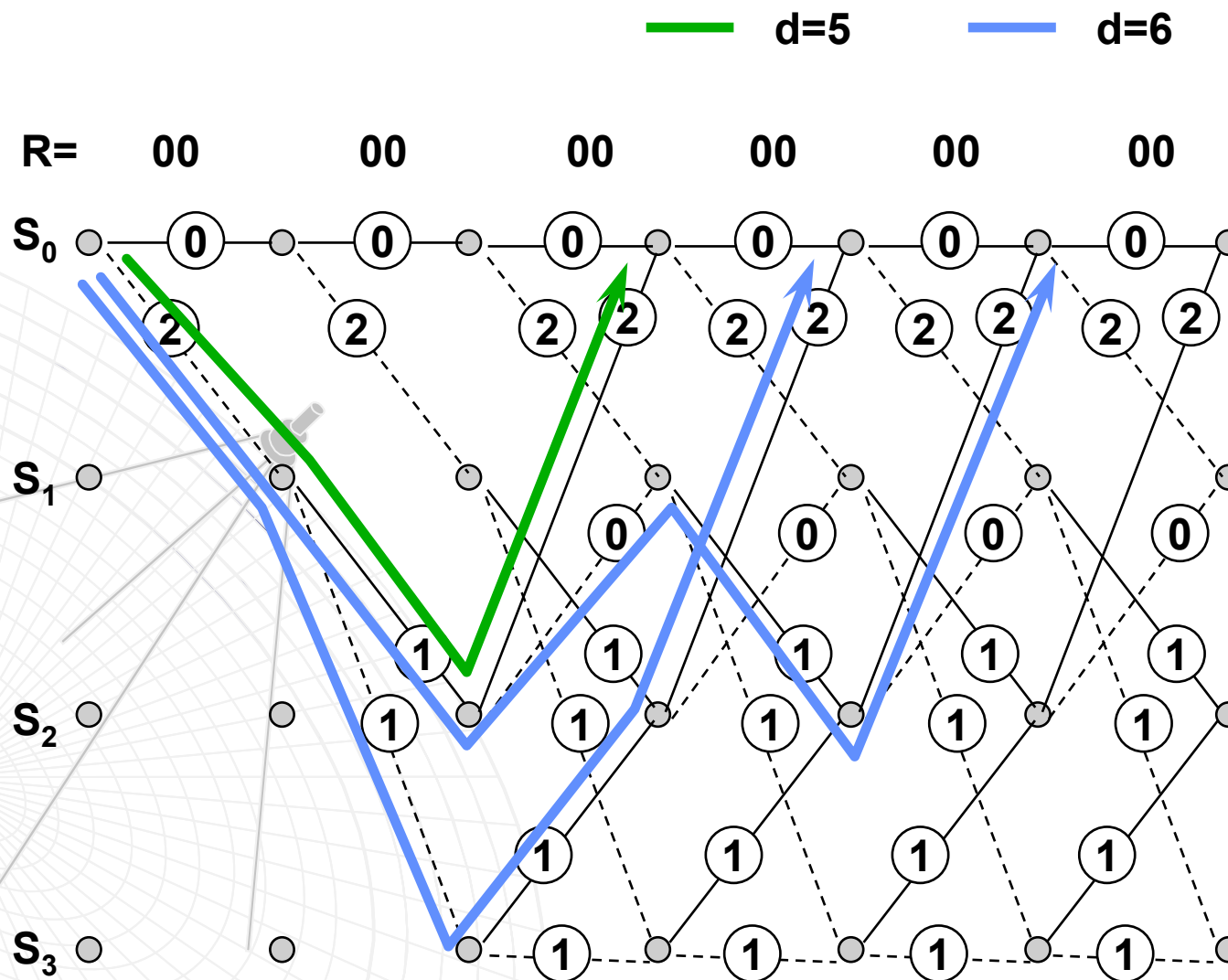
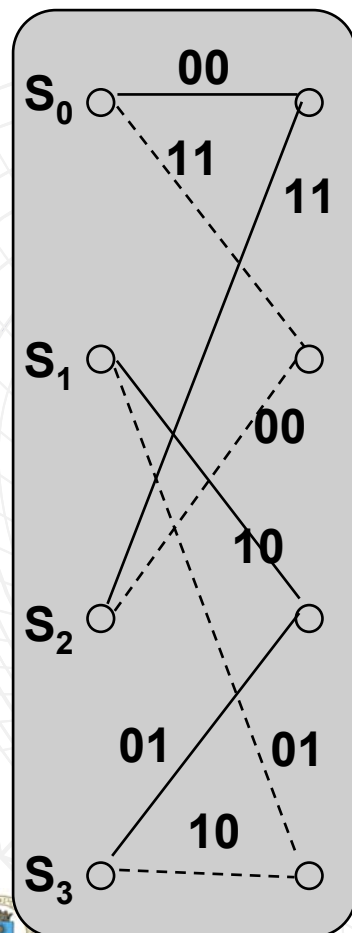
Suponiendo que se transmita la secuencia todo ceros se produce error cuando el trayecto todo ceros no sobrevive en la decodificación. La distancia mínima de un trayecto que produzca error se obtiene observando todos los trayectos que parten del estado 00 y vuelven a él un número de intervalos posterior.

Se etiqueta el trellis del decodificador con las distancias de Hamming al símbolo 00 en todas las ramas. La distancia mínima es la de menor peso y se denomina **distancia libre del código**.

La ganancia de codificación tiene como límite:

$$GC \leq 10 \log \left( \frac{k}{n} d_f \right)$$

# Distancia Mínima

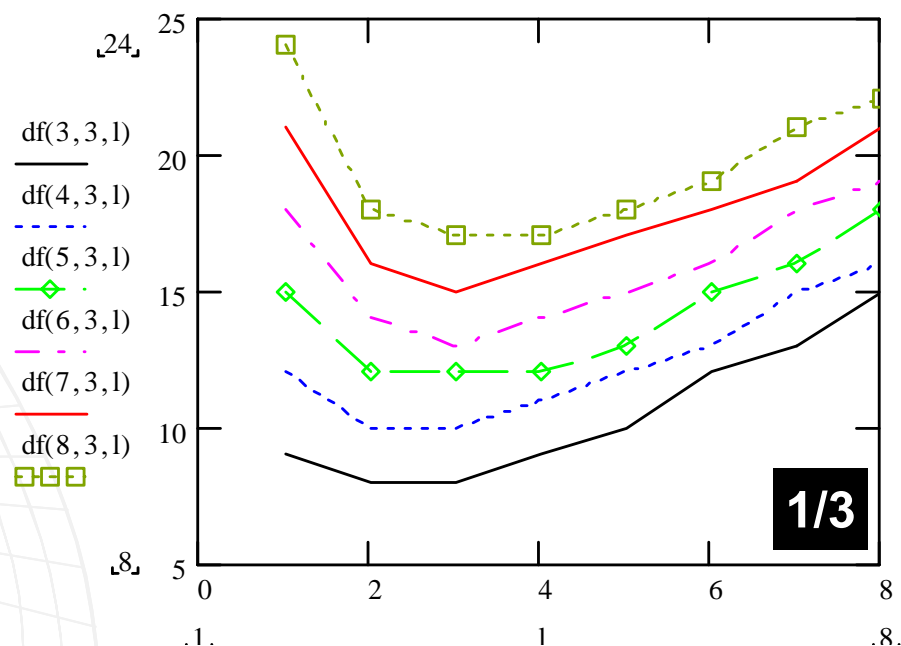
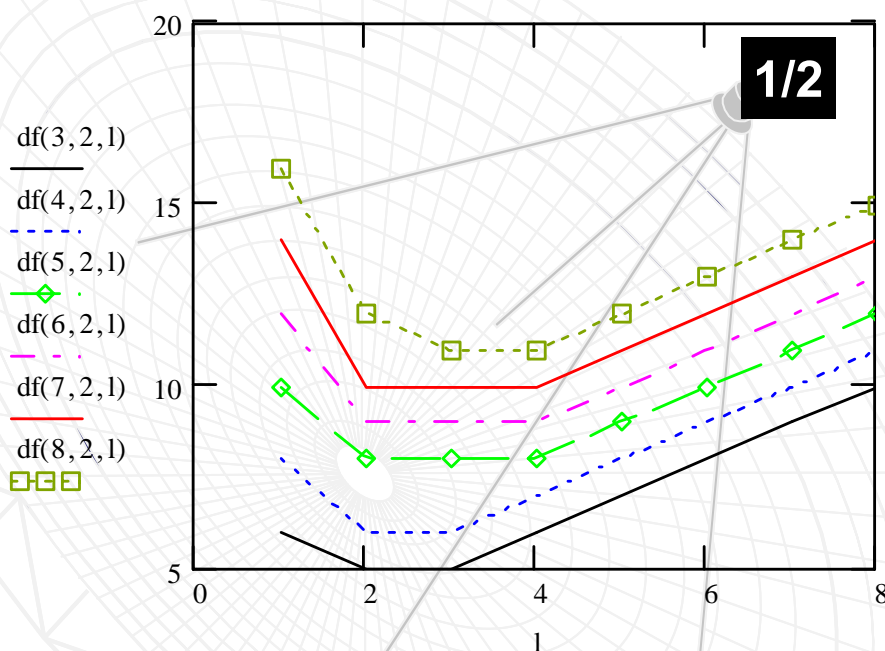




# Ganancia de Codificación

Para códigos 1/n la distancia libre puede obtenerse como:

$$d_f \leq \min_{l \geq 1} \left[ \frac{2^{l-1}}{2^l - 1} (K + l - 1)n \right]$$



# Ganancia de Codificación (HD)

La tabla lista el límite superior de la ganancia de codificación para códigos convolucionales con decisión dura, respecto a un sistema sin codificar con transmisión BPSK coherente.

La ganancia de codificación depende del valor de la BER.

Los valores de la tabla son asintóticos y válidos para valores bajos de la BER.

"1/2"			"1/3"		
K	df	CG(dB)	K	df	CG(dB)
3	5	3,98	3	8	4,26
4	6	4,77	4	10	5,23
5	7	5,44	5	12	6,02
6	8	6,02	6	13	6,37
7	10	6,99	7	15	6,99
8	10	6,99	8	16	7,27
9	12	7,78	9	18	7,78

# Ganancia de Codificación (SD)

- La tabla de lista la ganancia de codificación para códigos convolucionales con decisión blanda respecto a un sistema con transmisión BPSK coherente.

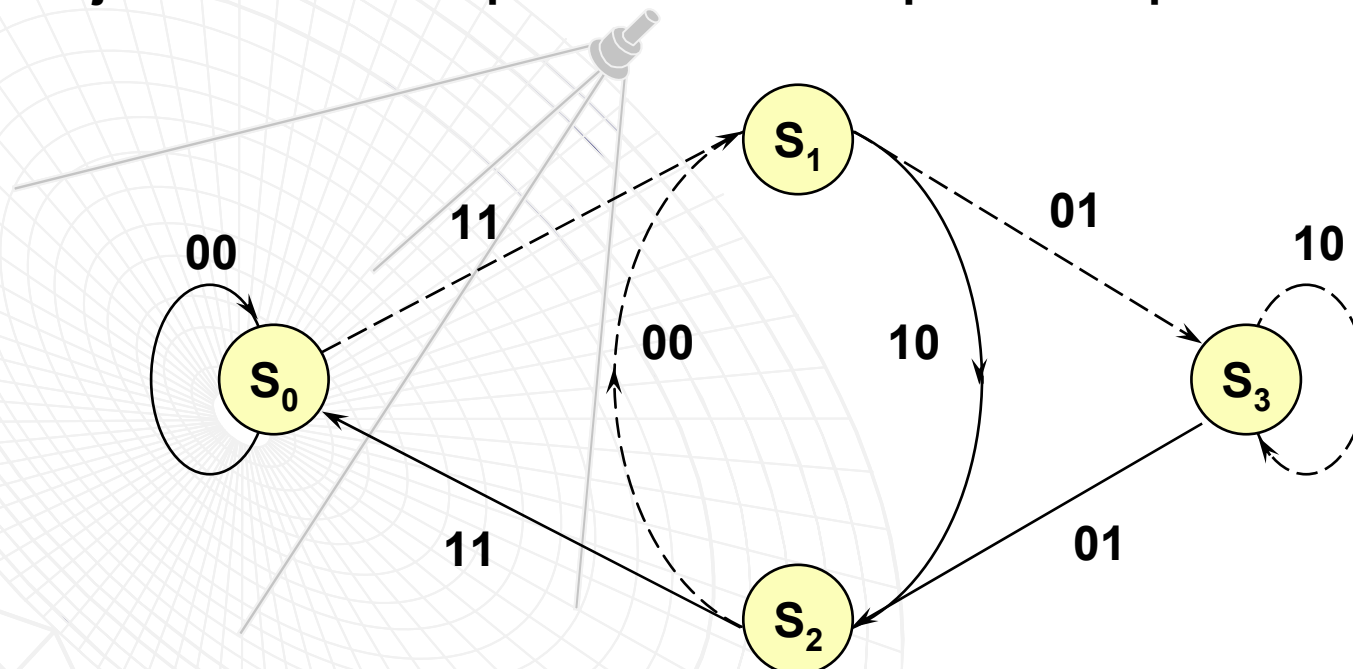
Eb/No (dB) (sin codificar)	Code Rate		"1/3"		"1/2"		
	Pb	K	7	8	5	6	7
6,8	$10^{-3}$		4,2	4,4	3,3	3,5	3,8
9,6	$10^{-5}$		5,7	5,9	4,3	4,6	5,1
11,3	$10^{-7}$		6,2	6,5	4,9	5,3	5,8
Upper Bound			7	7,3	5,4	6	7

# Diagrama de Estados

El codificador convolucional es una máquina de estados finitos y, por tanto, puede representarse mediante un diagrama de estados.

El diagrama de estados del codificador convolucional que venimos estudiando es el mostrado en la figura.

Las líneas de transición continuas representan entradas 0 al codificador y el símbolo junto a la línea representa la salida producida por el codificador.



# Diagrama de Estados Expandido

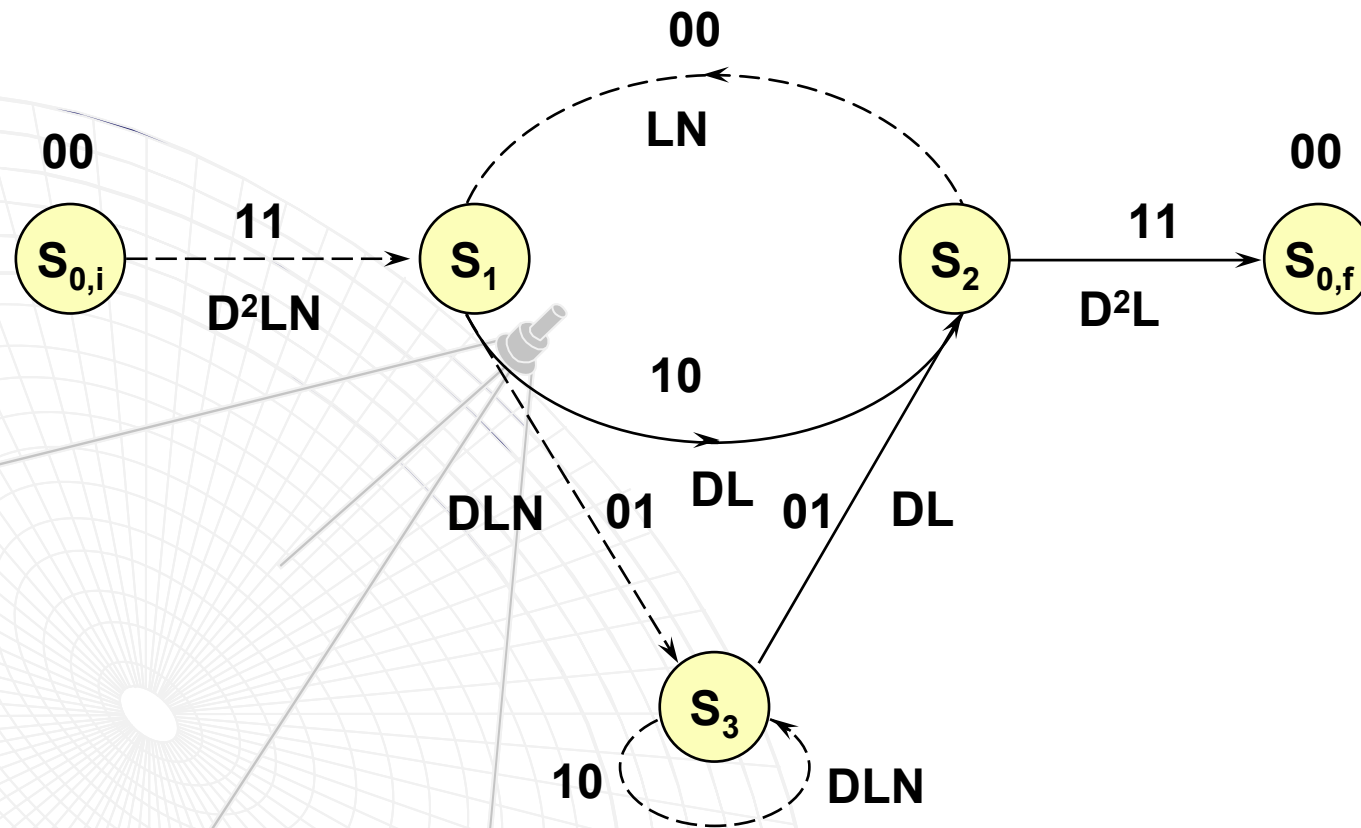
Otra representación que resulta útil para el cálculo de las propiedades de distancia de los codificadores convolucionales, necesarias para conocer sus prestaciones, es el diagrama de estados expandido. Éste se obtiene del diagrama de estados con el estado cero  $S_0$  dividido en un estado inicial  $S_{0,i}$  y un estado final  $S_{0,f}$  y borrando el autolazo de  $S_0$ .

Cada rama se etiqueta con:

- una ganancia de rama  $D^i$ , donde  $i$  es el peso de los  $n$  bits de salida asociados a la transición de la rama.
- un parámetro  $N^j$  donde  $j$  es el peso de los  $k$  bits de entrada.
- un parámetro  $L$  asociado a la longitud de la secuencia de entrada (permite determinar el número de ramas de un trayecto entre  $S_{0,i}$  y  $S_{0,f}$ )

El diagrama de estados expandido correspondiente al codificador bajo estudio es el de la figura de la siguiente transparencia.

# Diagrama de Estados Expandido



# Función de Transferencia

Llamamos  $X_{0,i}$ ,  $X_{0,f}$ ,  $X_1$ ,  $X_2$  y  $X_3$  a las variables de estado correspondientes a los estados del diagrama expandido. Pueden obtenerse las siguientes ecuaciones de estado:

$$X_1 = D^2 L N X_{0,i} + L N X_2$$

$$X_2 = D L X_1 + D L X_3$$

$$X_3 = D L N X_1 + D L N X_3$$

$$X_{0,f} = D^2 L X_3$$

La función de transferencia, definida como  $T(D,L,N) = X_{0,f} / X_{0,i}$ , resulta:

$$\begin{aligned} T(D,L,N) &= \frac{D^5 L^3 N}{1 - D L (1 + L) N} \\ &= D^5 L^3 N + D^6 L^4 (1 + L) N^2 + \dots \end{aligned}$$

Vemos que hay un trayecto de distancia 5, longitud 3 y que difiere en un bit de la secuencia todos cero. Hay dos trayectos de distancia 6, longitudes 4 y 5 y que difieren en dos bits de la secuencia todos cero.



# Características Límite

Para un canal BSC, con probabilidad de transición  $p$ , la probabilidad de bit erróneo está acotada por:

$$P_B \leq \frac{dT(D, L = 1, N)}{dN} \Big|_{N=1, D=2\sqrt{p(1-p)}}$$

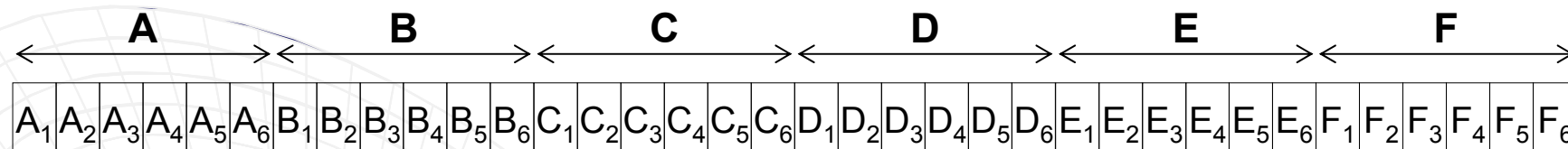
Para el codec de tasa 1/2 con distancia libre 5, con decisión dura y transmisión BPSK coherente, se puede demostrar que:

$$P_B \leq \frac{Q(\sqrt{5E_b/N_0})}{[1 - 2\exp(-E_b/2N_0)]^2}$$

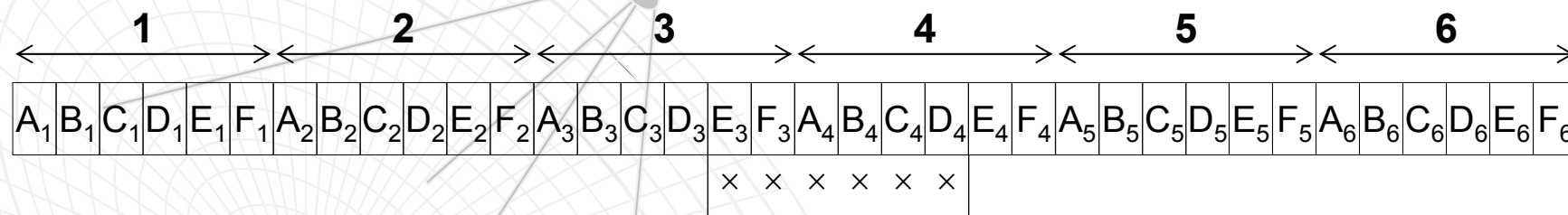
# Entrelazado (*Interleaving*)

**Objetivo:** Convertir los errores a ráfagas producidos por el canal de propagación en errores independientes y aislados

- Convertir el canal con memoria en un canal sin memoria
- Errores a ráfagas: ruido impulsivo, desvanecimientos, etc.

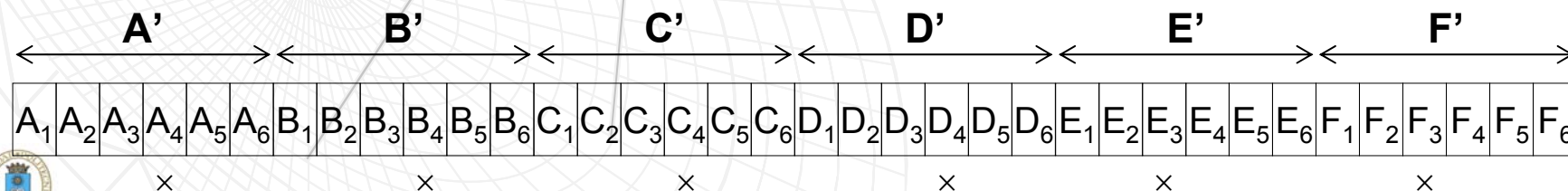


(a) Entrada al entrelazador



Ráfaga de errores

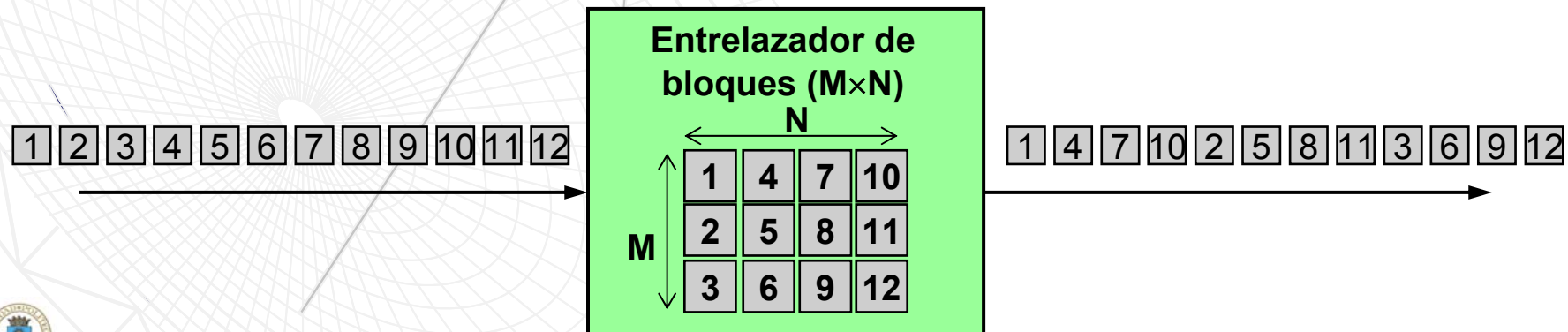
(b) Señal recibida con una ráfaga de errores (secuencia entrelazada)



(c) Entrada al decodificador (secuencia desentrelazada)

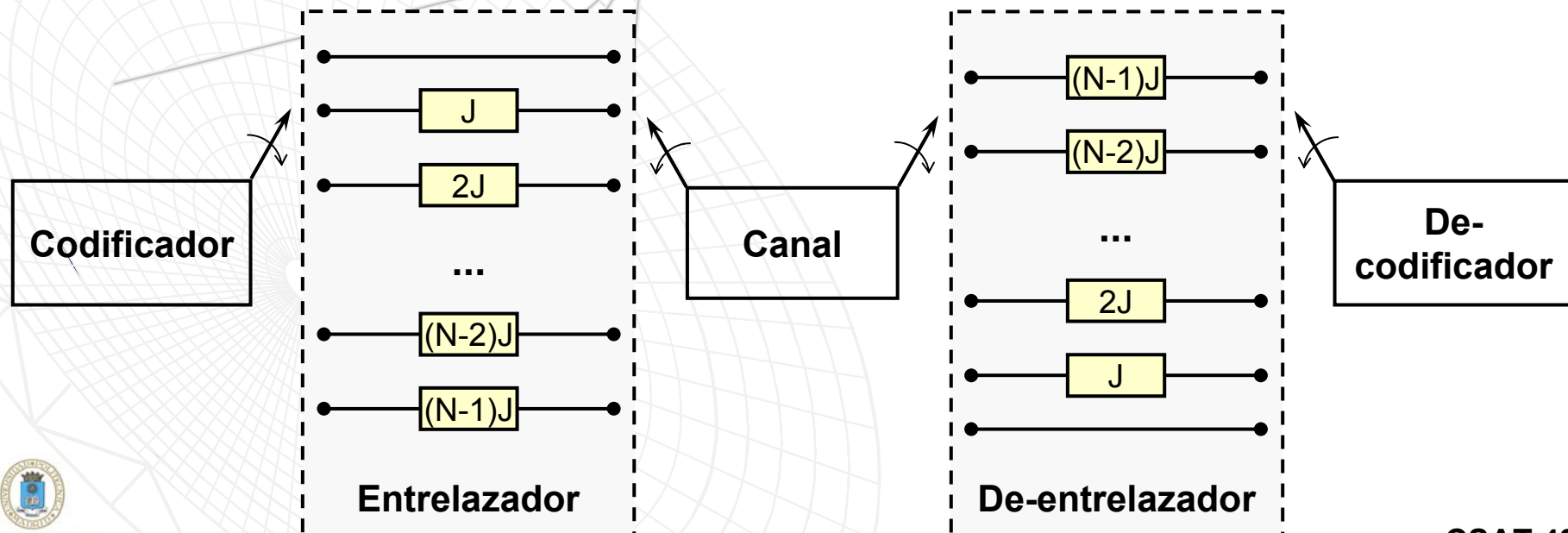
# Entrelazado por bloques

- Los símbolos van rellenando las columnas de una matriz  $M \times N$  con la secuencia codificada
- Los símbolos se envían al modulador por filas
- La profundidad del entrelazador ( $N$ ) debe ser mayor que duración de la ráfaga de errores
- El valor de  $M$  se selecciona en función el esquema de codificación:
  - Códigos de bloques:  $M > n$  (como máximo, un error por palabra código)
  - Códigos convolucionales:  $M > K$  (como máximo, un error cada  $K$  símbolos)
- Retardo:  $2MN$  símbolos; Memoria:  $MN$  símbolos



# Entrelazado convolucional

- La estructura se basa en un conjunto de  $N$  registros de desplazamiento
- Cada registro almacena  $J$  símbolos más que el anterior
- Los símbolos rellenan secuencialmente los registros
- El conmutador va recorriendo los  $N$  registros, extrayendo un símbolo cada vez
- Retardo:  $NJ(N-1)$  símbolos; Memoria:  $NJ(N-1)/2$  símbolos
  - Mitad de memoria y retardo que entrelazado por bloques

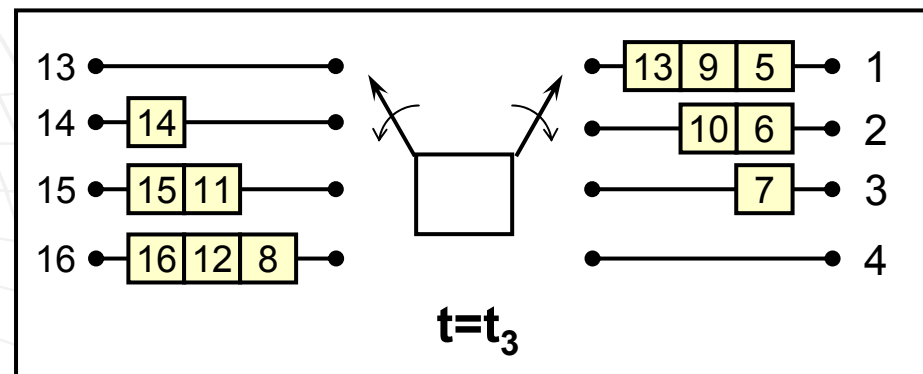
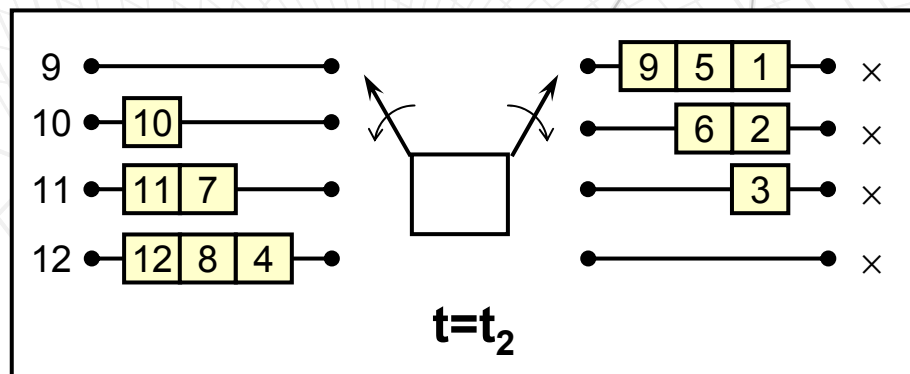
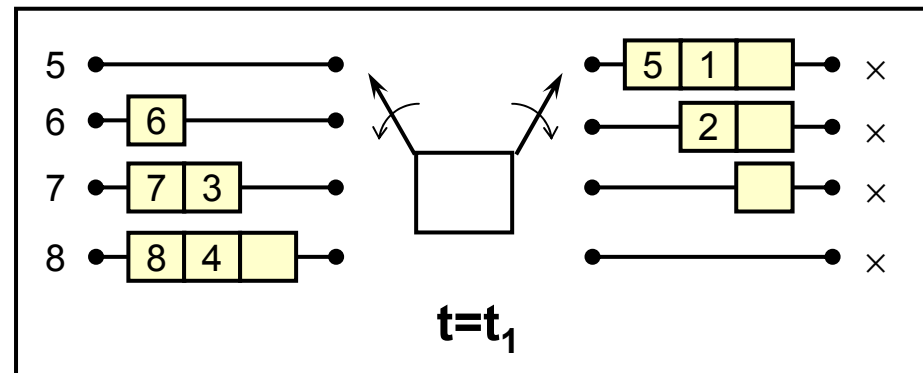
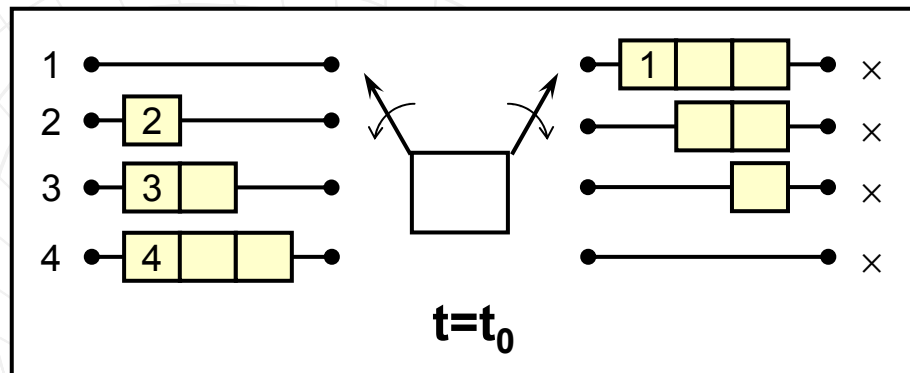


# Entrelazado convolucional

## • Ejemplo:

–Secuencia de entrada: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...

– $J=1$ ,  $N=4$



# Modulación Codificada

- La corrección adelantada de errores FEC con codificación de bloques o convolucional intercambia eficiencia en ancho de banda por eficiencia en potencia.
- La expansión del ancho de banda resultante es igual al recíproco de la tasa de codificación  $R$ , mientras que la reducción en potencia es igual a la ganancia de codificación.
- En los sistemas por satélite que estén limitados en banda como serán los sistemas móviles MSS, el uso de FEC convencional no resultará satisfactorio.
- El MPSK es un sistema de modulación que intercambia menor eficiencia en potencia por una mayor eficiencia en ancho de banda. Por ello, si se combina un código adecuadamente elegido con un sistema MPSK eficiente en ancho de banda, el incremento del ancho de banda producido por el código se puede compensar por la mejora obtenida con la modulación.
- El sistema codificado permite una reducción en la  $E_b/N_0$  requerida en una cantidad igual a la ganancia de codificación, lo que puede compensar el incremento de potencia requerido por la modulación MPSK y proporcionar así una mejora neta de prestaciones.



# Modulación Codificada Trellis (TCM)

- Considérese una señal QPSK sin codificar con cuatro símbolos (2 bits por símbolo).
- Si se utiliza una tasa de codificación de  $2/3$  se requiere una expansión del ancho de banda de  $3/2$ . Si ahora se utiliza una modulación 8PSK con el código, la reducción en ancho de banda es de  $3/2$ , con lo que el sistema codificado 8PSK utiliza el mismo ancho de banda que el QPSK sin codificar.
- Sin embargo, una señal 8PSK requiere aproximadamente 4 dB más de potencia que el QPSK y el código deberá tener una ganancia de codificación superior para que el sistema sea efectivo.
- Un código integrado con un sistema de modulación se denomina sistema de modulación codificado trellis ó TCM. Pueden usarse codificadores de bloques o convolucionales. Se prefieren estos últimos por permitir de forma más simple la implementación de decisión soft con el algoritmo de Viterbi.
- Consiste en un codificador convolucional seguido de un modulador con un mapping de señal que proyecta la redundancia introducida por el codificador sobre un conjunto expandido de símbolos de modulación.

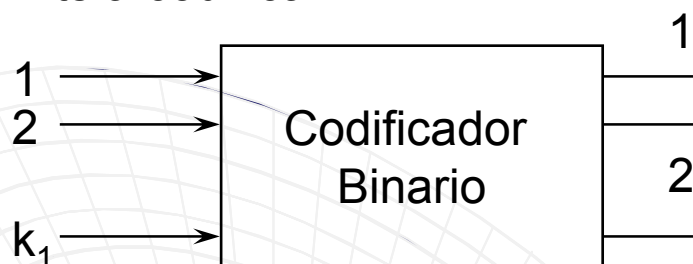


## OBJETIVO:

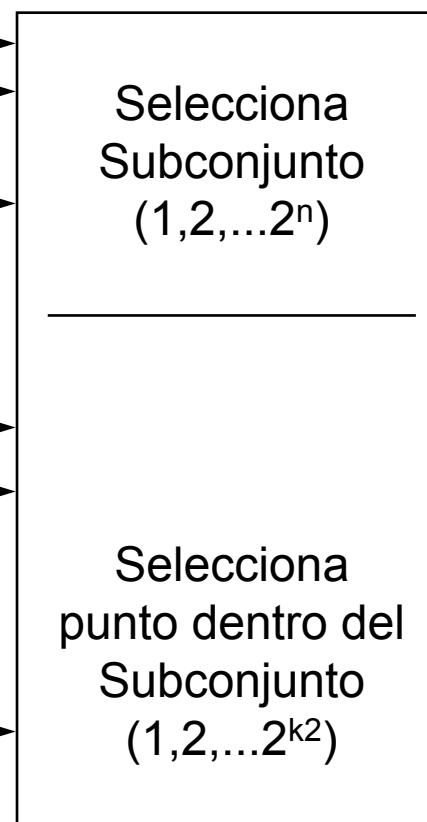
**Aumentar la distancia euclídea entre aquellas señales que es más probable confundir (por ejemplo, 000 y 001).**

# Codificador-Modulador

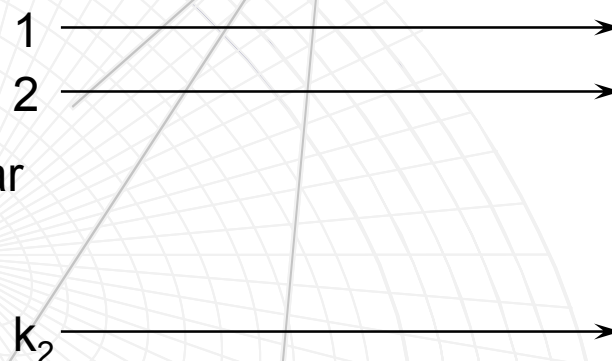
Bits a codificar



Mapping



Bits sin codificar

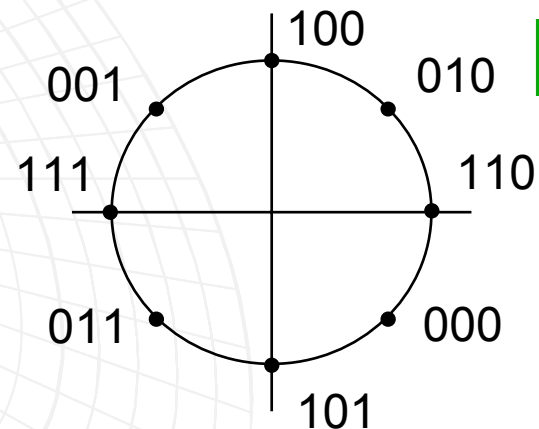
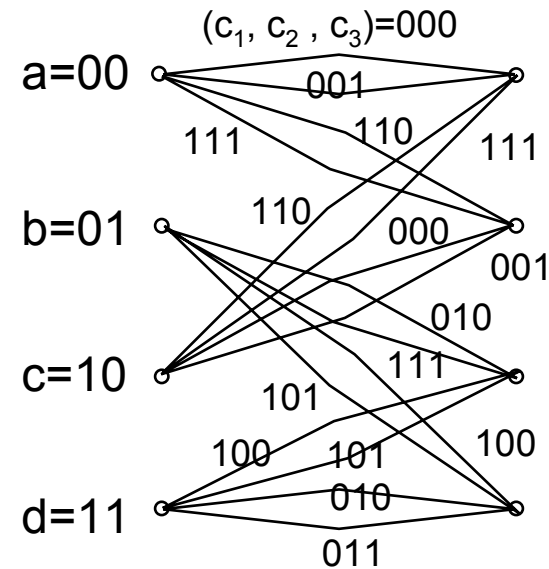
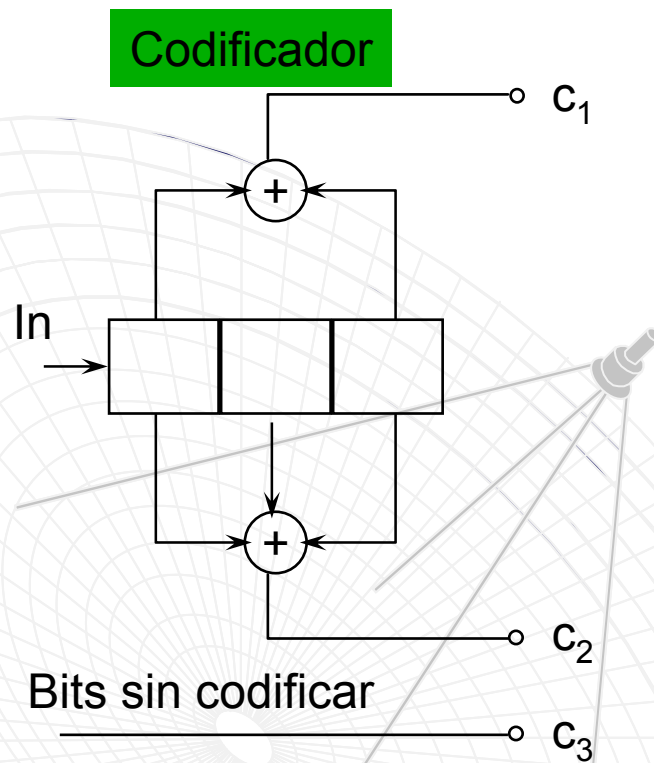


Punto de la  
constelación  
de señal

# Trellis Coded Modulation (TCM)

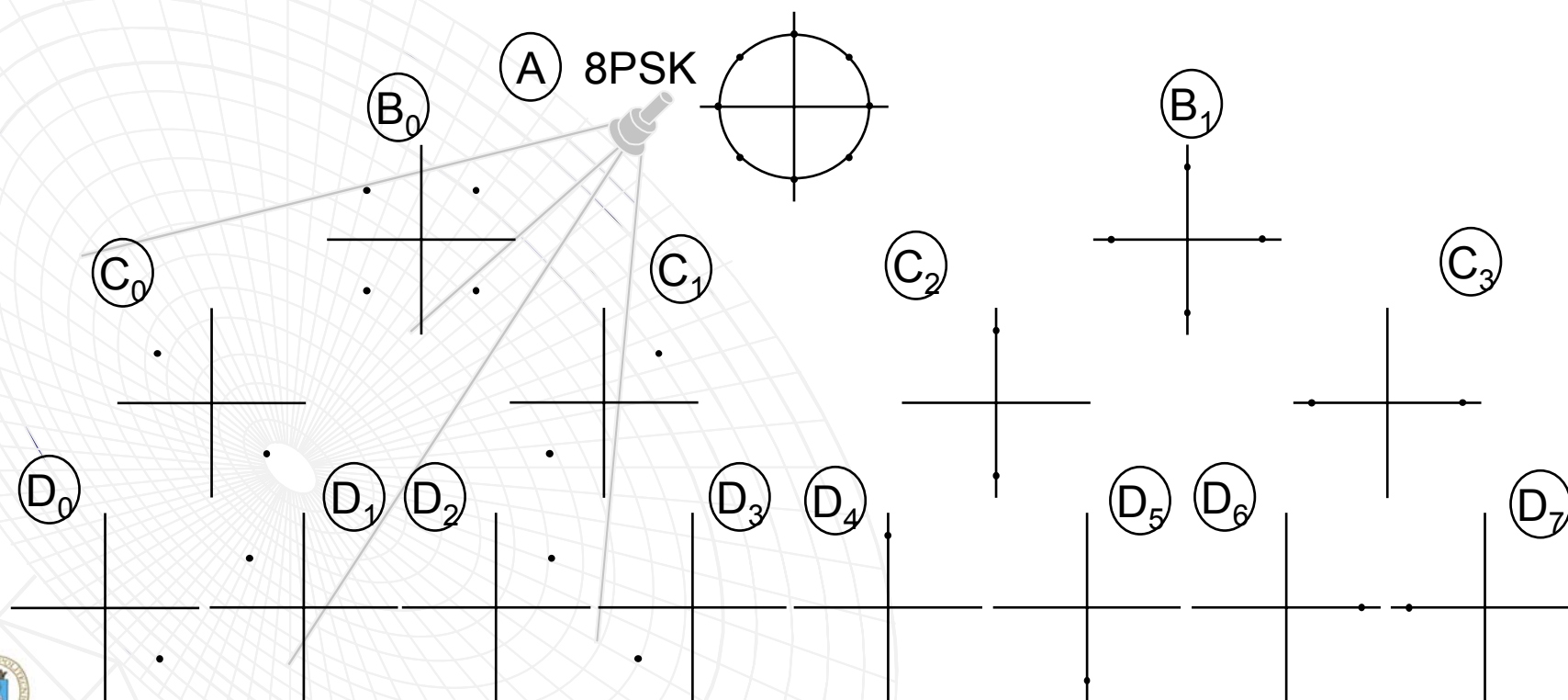
- En general, dado un ancho de banda, se selecciona primero el tamaño  $2^k$  del conjunto de símbolos que se requieren para transmitir sin codificación una determinada tasa de bits.
- Así, un sistema sin codificar requerirá un esquema de modulación  $2^k$ .
- Después se dobla el tamaño del conjunto de símbolos a  $2^{k+1}$  y se le incorpora un codificador convolucional de tasa  $R=k/(k+1)$  que producirá un bit redundante extra.
- De esta forma no hay cambio en la tasa de símbolos, el sistema codificado y sin codificar usan el mismo ancho de banda y transmiten el mismo número de bits de información por símbolo, o sea  $k$  bits por símbolo.
- El problema que resta es proyectar los bits codificados por el codificador sobre la constelación de la señal o conjunto de símbolos.

# Ejemplo TCM-2/3 8PSK



# Particionamiento

- Un esquema denominado proyección por particionamiento del conjunto fue desarrollado por Ungerboeck. La constelación se divide primero en dos subconjuntos que maximizan la distancia euclídea entre los símbolos. Los subconjuntos se subdividen a su vez con el mismo criterio, y así sucesivamente.



# Reglas de Partición

En el conjunto de señales de 8 fases los puntos de señal se sitúan en el círculo de radio  $\sqrt{E}$ . Tienen por tanto una distancia Euclídea mínima de:

$$d_0 = 2\sqrt{E} \sin \frac{\pi}{8} = \sqrt{(2 - \sqrt{2})E} = 0.765\sqrt{E}$$

En la primera partición los ocho puntos se subdividen en dos subconjuntos de cuatro puntos cada uno. La distancia Euclídea mínima se incrementa a:  $d_1 = \sqrt{2}E$ .

En el segundo nivel de partición cada subconjunto se subdivide en dos de dos puntos con una distancia Euclídea mínima de:  $d_2 = 2\sqrt{E}$ .

El diagrama de trellis contiene cuatro estados. Cada transición contiene dos ramas paralelas. Vemos que se ha construido el código siguiendo las reglas:

- a) Las transiciones paralelas se asignan a puntos de la constelación separados por la máxima distancia Euclídea  $d_2$ .
- b) Las transiciones que se originan en un estado, o que convergen a un estado, se les asignan subconjuntos con distancia  $d_1$
- c) Todos los puntos de señal son equiprobables.

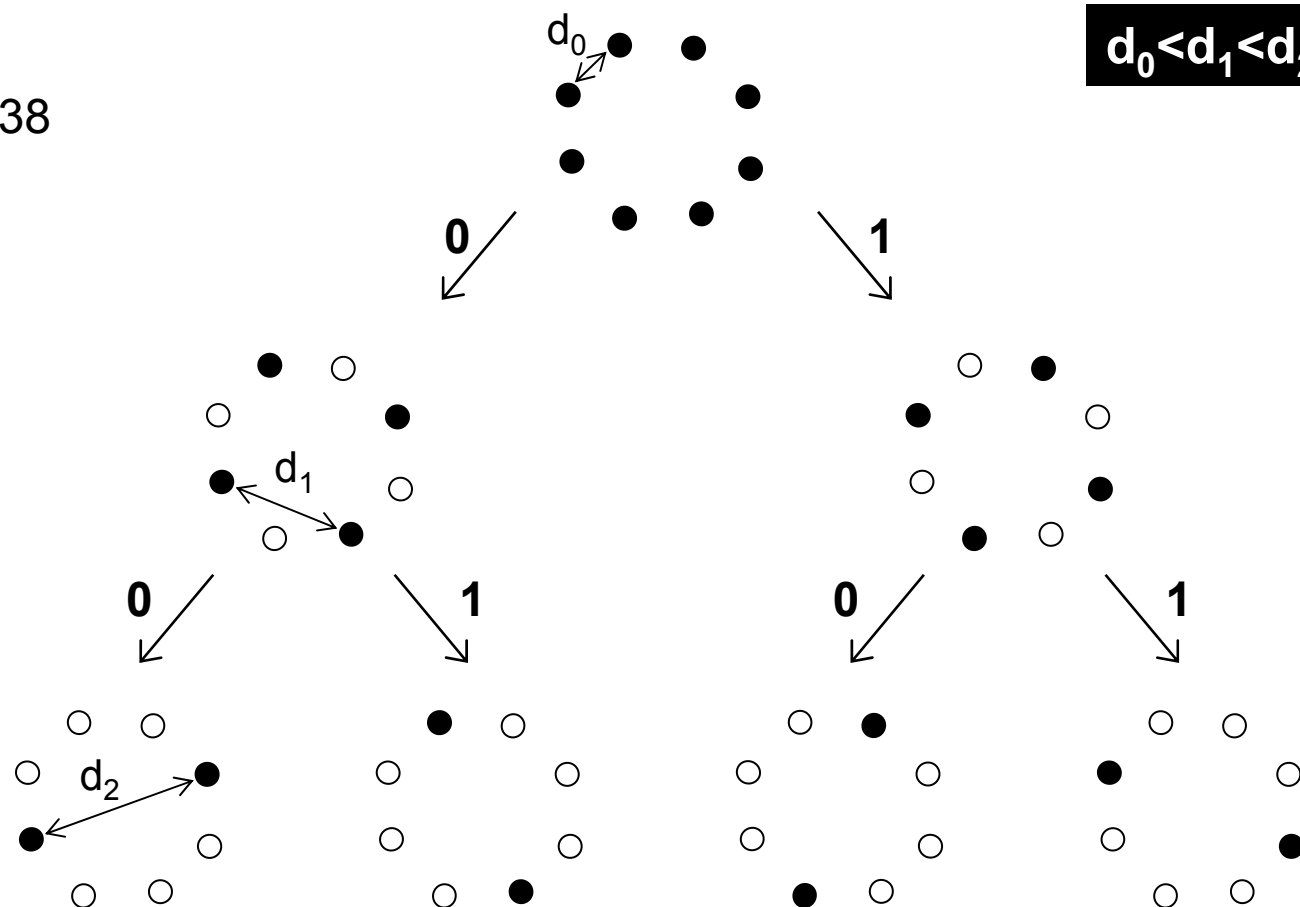
# Particionamiento: 8PSK

$$d_0 = 2 \cdot \sin(\pi/8) \sim 0.38$$

$$d_1 = \sqrt{2} \sim 1.41$$

$$d_2 = 2$$

$$d_0 < d_1 < d_2$$



Número  
de señal

00  
0

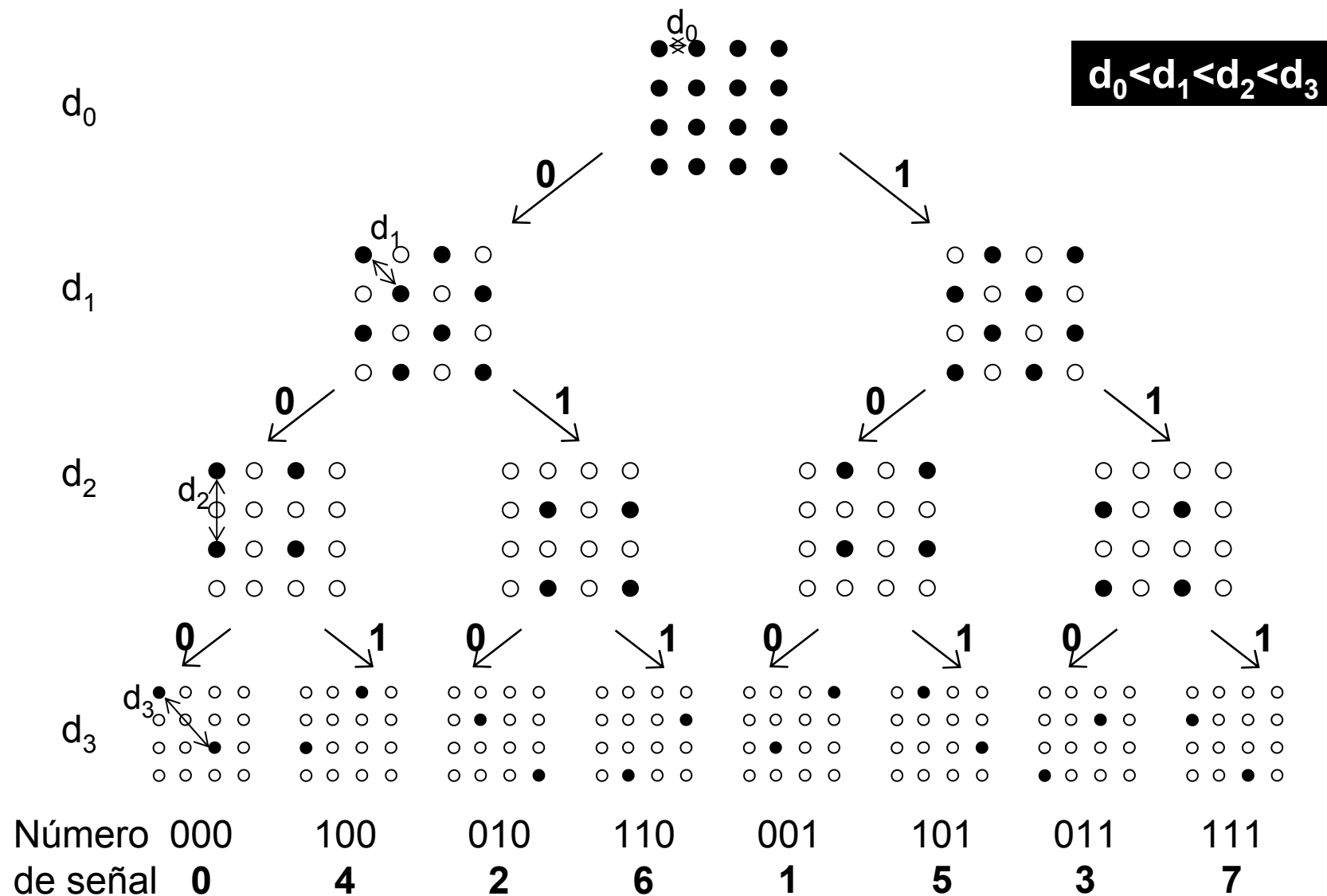
10  
2

01  
1

11  
4

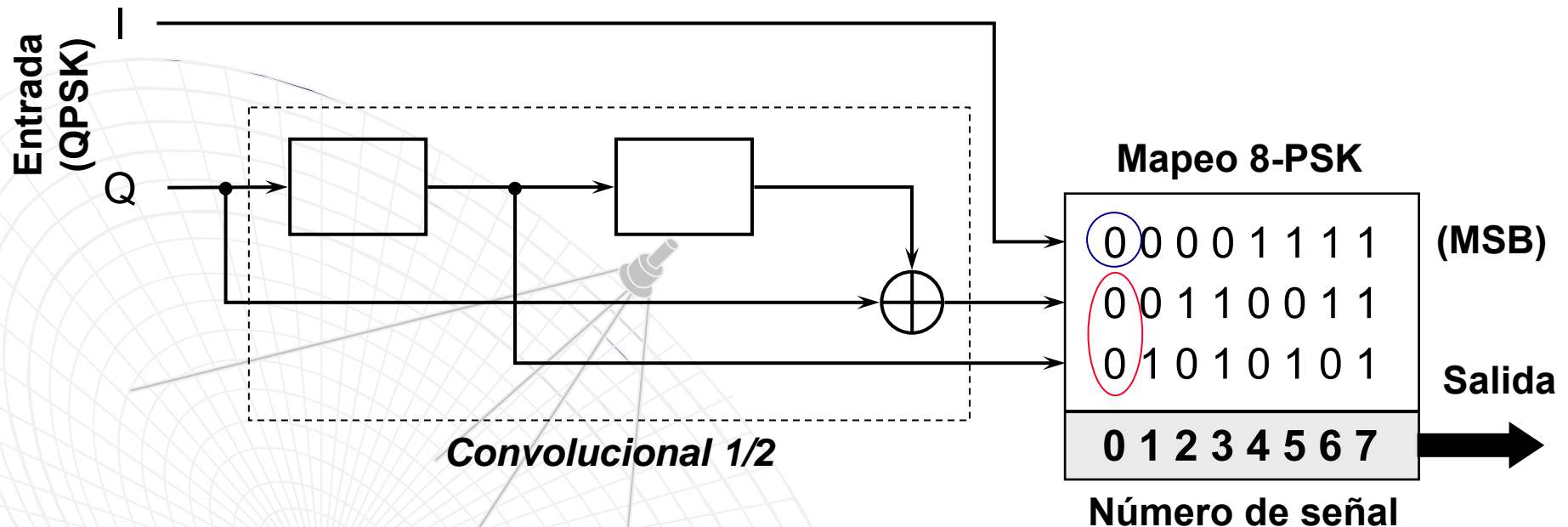


# Particionamiento: 16QAM



# Ejemplo: TCM-1/2 8PSK (4 estados)

Se emplea un codificador convolucional de tasa  $\frac{1}{2}$  (dos estados).

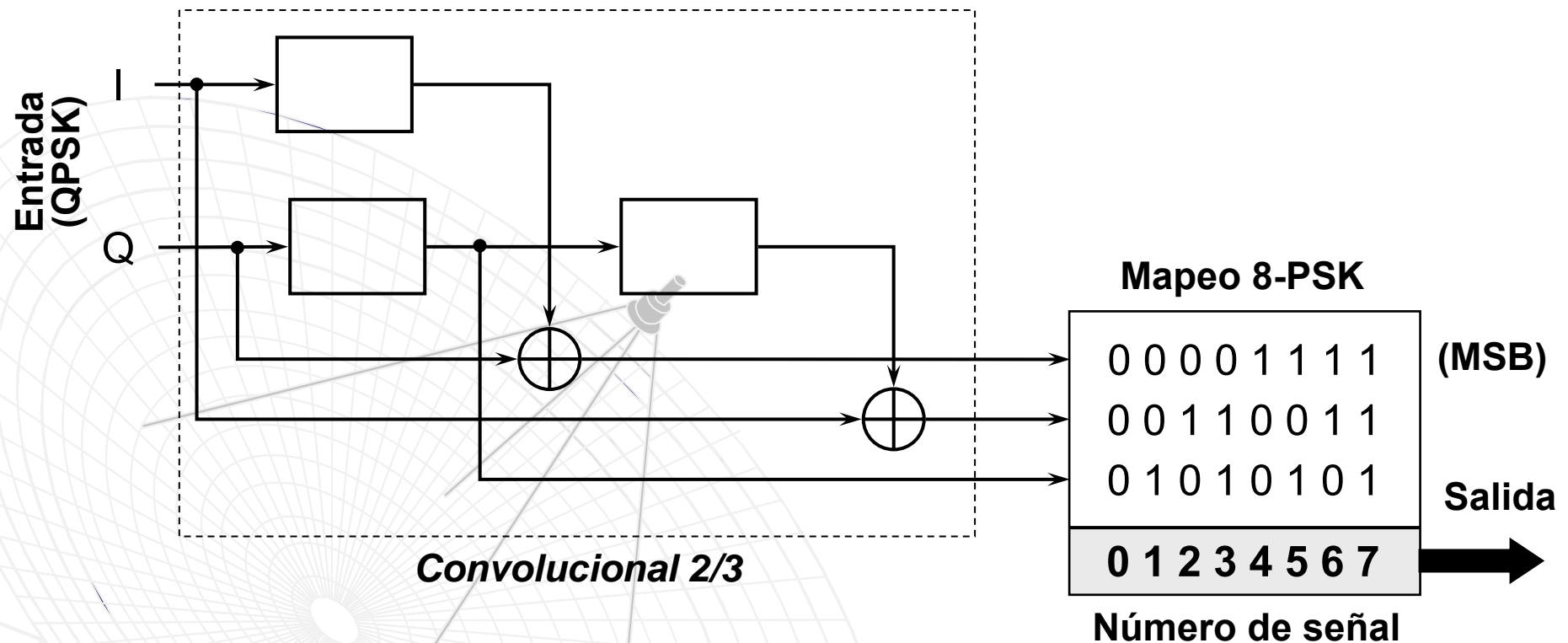


Este bit (uncoded) determina el punto dentro de la sub-constelación

Estos dos bits (coded) determinan la sub-constelación

# Ejemplo: TCM-2/3 8PSK (8 estados)

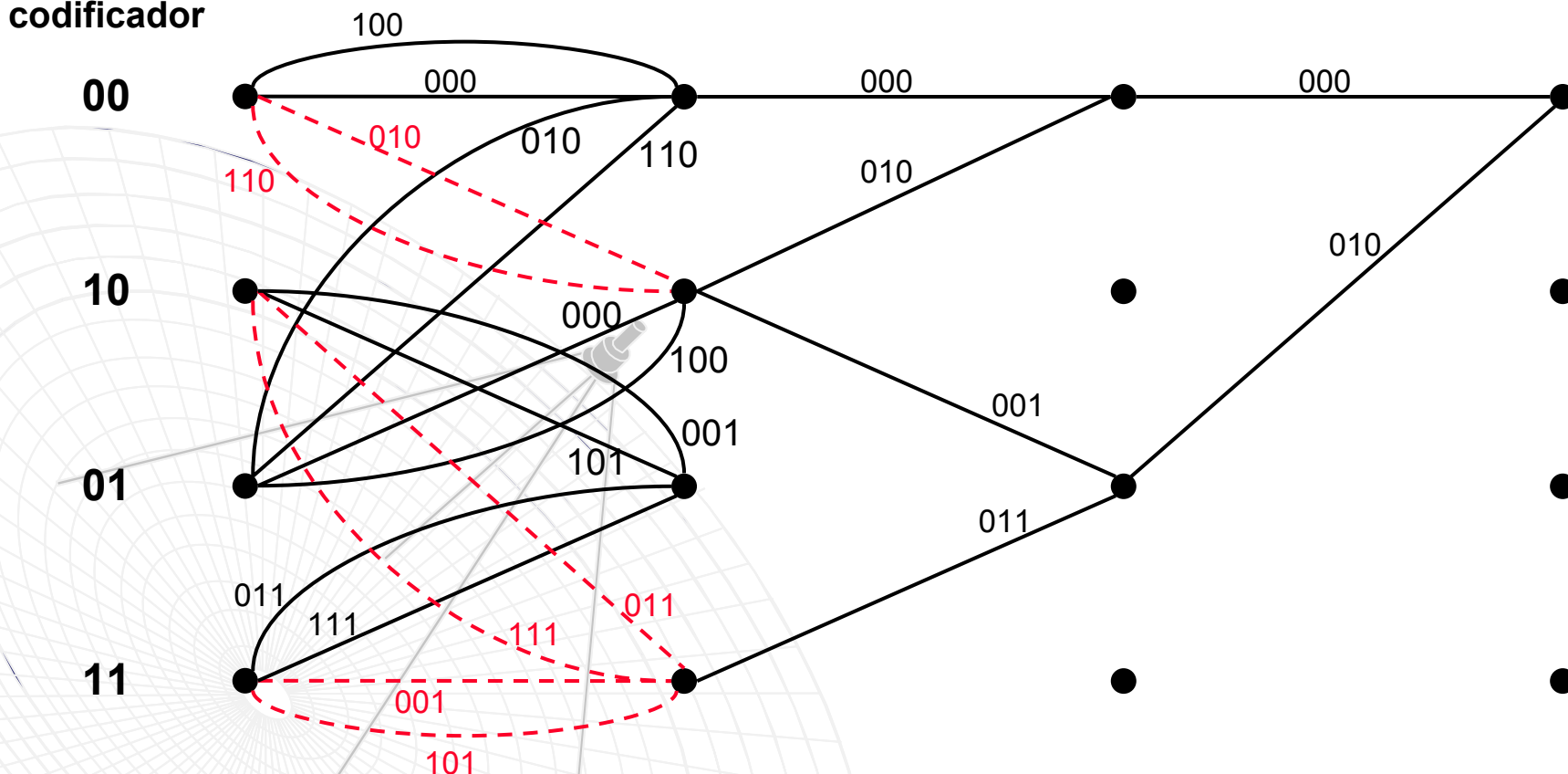
Se emplea un codificador convolucional de tasa 2/3 (tres estados).



Este codificador de tres estados tiene mayor complejidad que el de dos.  
Se codifican todos los bits.

# Ejemplo: trellis del TCM-1/2 8PSK

Estado del  
codificador



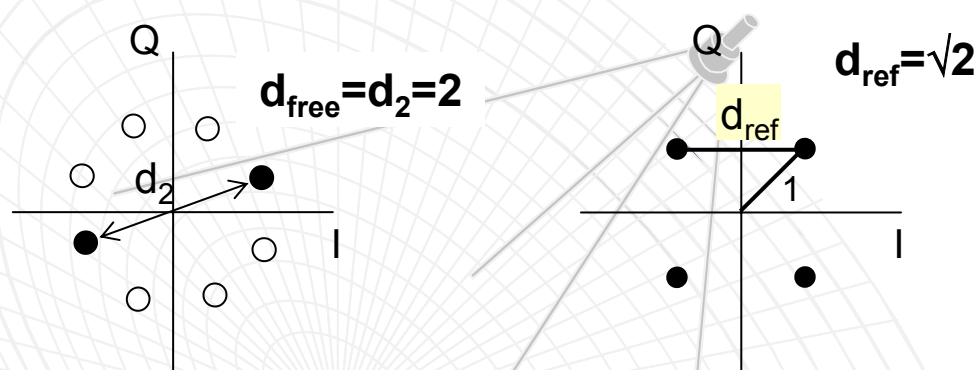
Las transiciones paralelas del Trellis corresponden a puntos separados la máxima distancia euclídea (nivel máximo de particionamiento).

# Prestaciones de un TCM

Las prestaciones de un sistema TCM se cuantifican con la *ganancia de codificación asintótica*  $G_a$ :

$$G_a = 10 \log_{10} \left( \frac{d_{\text{free}}^2}{d_{\text{ref}}^2} \right)$$

Para el código 8-PSK con convolucional  $1/2$ , respecto de la QPSK con la misma  $E_b/N_0$ , se tiene:



$$G_a = 10 \log_{10} \left( \frac{2^2}{\sqrt{2}^2} \right) = 3 \text{ dB}$$

A mayor número de estados, mayor ganancia de codificación:

Número de estados	4	8	16	32	64	128	256	512
Ga (dB)	3	3,6	4,1	4,6	4,8	5	5,4	5,7

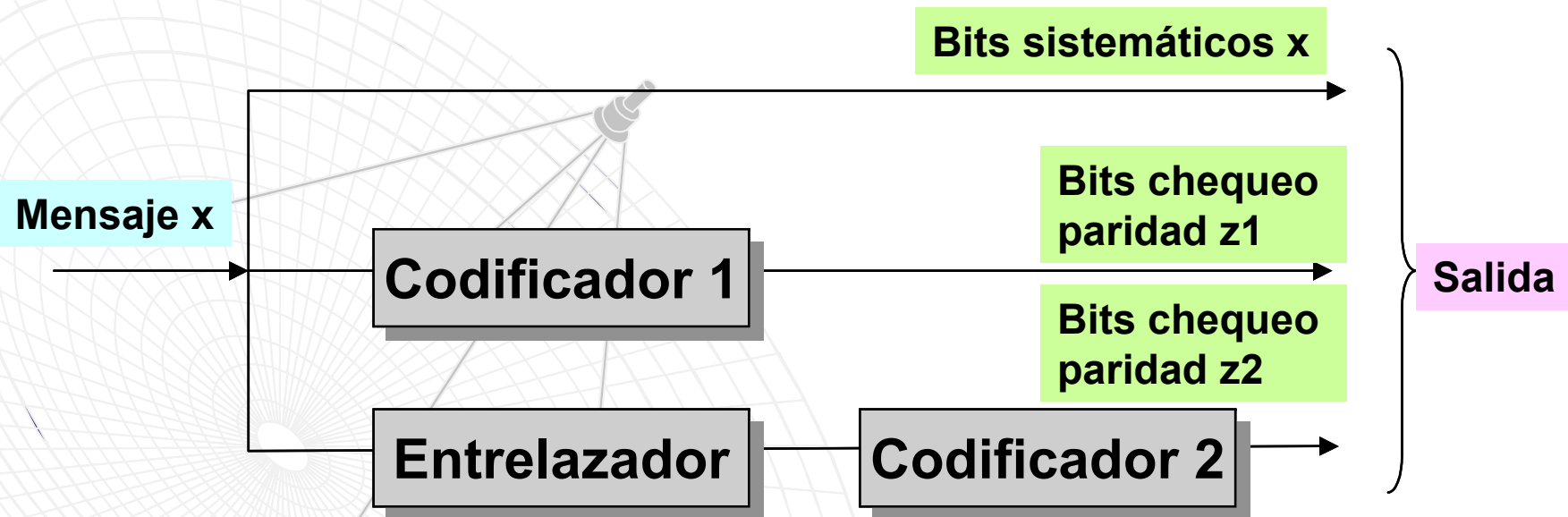
# Turbocódigos

- En los códigos tradicionales hay que aumentar la longitud de la palabra, en los códigos de bloques, o la longitud de restricción en los convolucionales para acercarnos al límite de capacidad de Shannon.
- La complejidad tiende a crecer exponencialmente.
- Una alternativa es construir códigos con longitud de bloque “equivalente” grande pero estructurada de forma que la decodificación pueda hacerse por trozos en un determinado número de pasos manejable.
- Los intentos que han tenido mayor éxito han sido los turbocódigos y los códigos de chequeo de paridad de baja densidad LDPC.



# Turbocodificadores

En su forma más básica el codificador consiste en dos codificadores sistemáticos (*codificadores componentes*) unidos por medio de un entrelazador.



# Entrelazador y Codificador RSC

**Las razones para usar un entrelazador en un turbo-codificador son:**

- Agrupar errores que pueden producirse fácilmente, en la mitad del turbo codificador, con errores muy ocasionales, que se producen en la otra mitad. A esto se debe en gran parte el mejor comportamiento frente a los codificadores tradicionales.**
- Proporcionar robustez frente a decodificación desadaptada que se produce cuando las estadísticas del canal no son conocidas o se han especificado incorrectamente.**

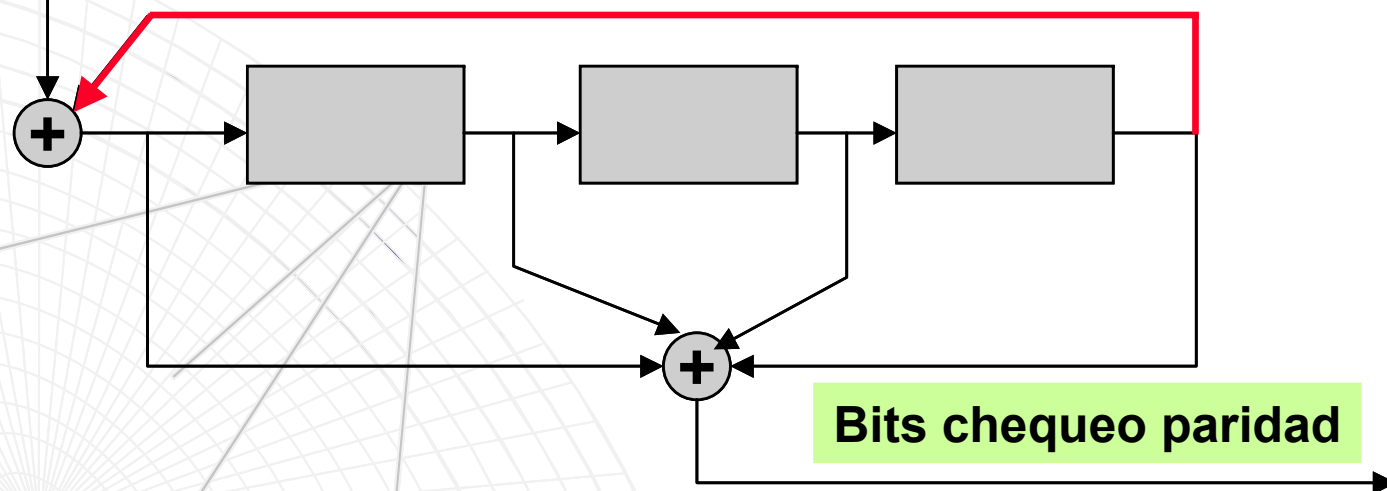
**Típicamente (pero no necesariamente) se utiliza el mismo codificador para los dos codificadores que forman el turbo codificador. Los codificadores recomendados son codificadores convolucionales sistemáticos recursivos de corta longitud de restricción (RSC).**

**La razón para hacer recursivo el codificador convolucional (es decir realimentar una o más de las salidas del registro de desplazamiento a la entrada) es hacer depender el estado del registro de salidas pasadas. Esto afecta al comportamiento de los patrones de error (un error en los bits sistemáticos produce un número  $\infty$  de errores de paridad) y ello redundará en mejores prestaciones.**

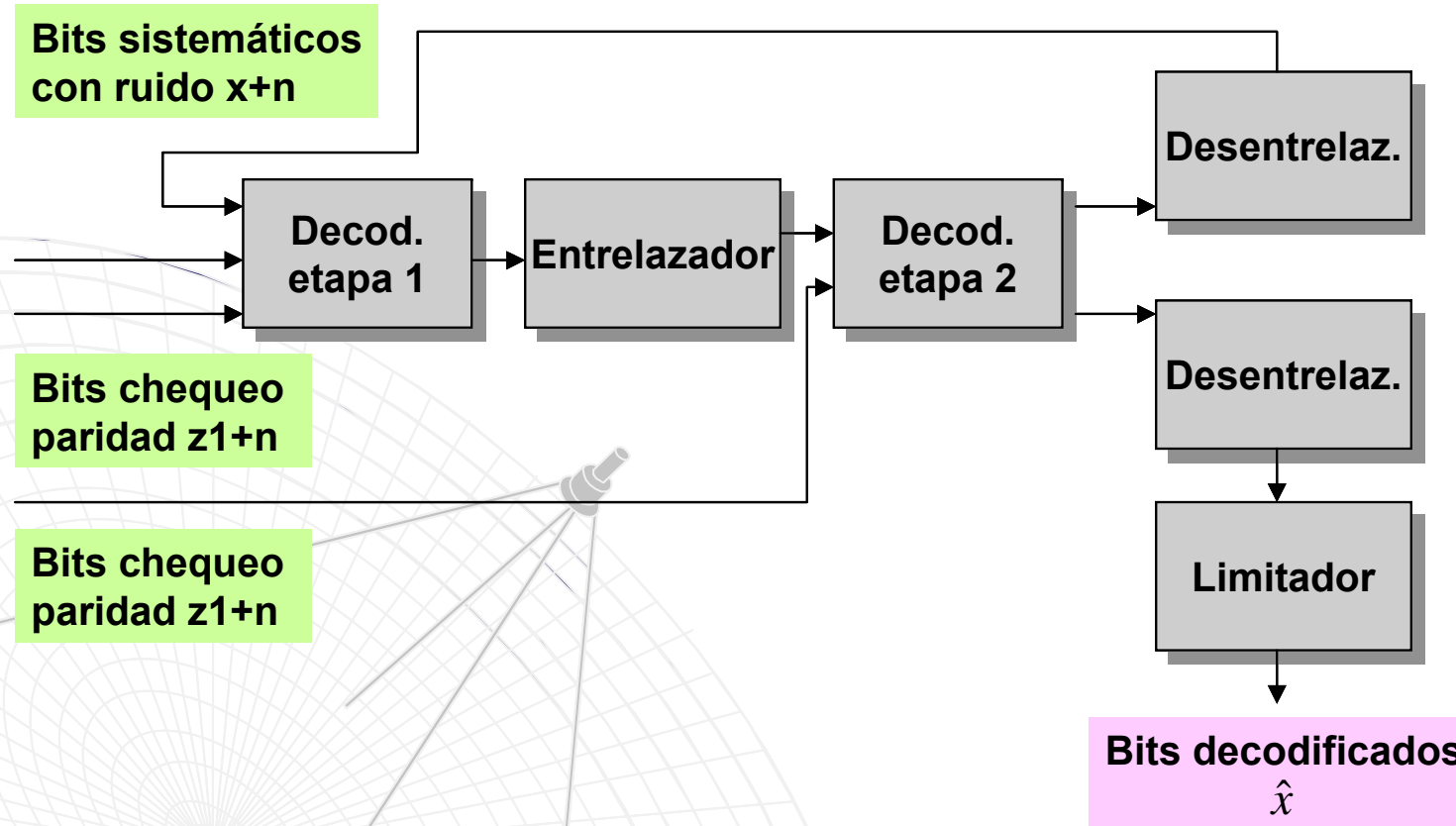
# Codificador RSC de 8 estados

Mensaje x

Bits sistemáticos x



# Turbo decodificador



Cada uno de los decodificadores usa un algoritmo BCJR (Bahl, Cocke, Jelinek, Ravi) para proporcionar una detección MAP (máximo a posteriori).

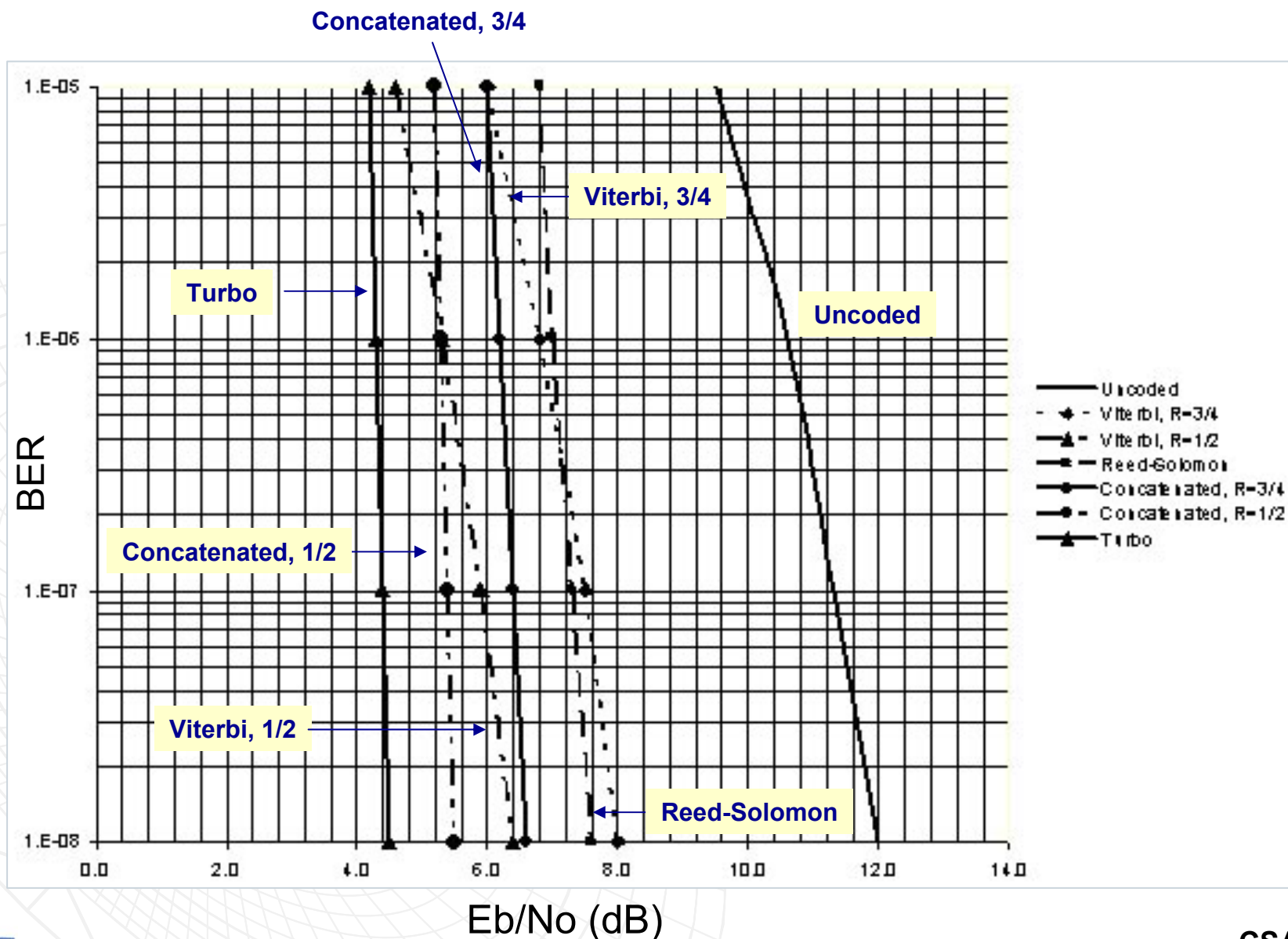
Este algoritmo difiere del de Viterbi.

# Algoritmo BCJR vs Viterbi

BCJR	Viterbi
<ul style="list-style-type: none"> <li>- Entrada y salida <u>blandas</u> con dos recursiones hacia <u>delante</u> y hacia <u>atrás</u>, ambas con decisiones <u>blandas</u></li> <li>- Mayor <u>complejidad</u> computacional por recursión</li> <li>- Minimiza los errores de bit hacia atrás</li> </ul>	<ul style="list-style-type: none"> <li>- Entrada blanda y salida dura con una sola recursión hacia adelante con decisiones blandas</li> <li>- La recursión se para con una decisión <u>dura</u> y sólo <u>sobrevive</u> un trayecto.</li> </ul>
<ul style="list-style-type: none"> <li>- Estimando las probabilidades a posteriori de los bits individuales en la palabra codificada (criterio MAP o de máxima verosimilitud a posteriori)</li> <li>- Para reconstruir la secuencia de datos se toman limitaciones duras de las salidas blandas del</li> </ul>	<ul style="list-style-type: none"> <li>- Estimador de secuencias de máxima verosimilitud (criterio ML)</li> <li>- Maximiza la función de verosimilitud para toda la secuencia y <u>no</u> para cada bit.</li> </ul>

algoritmo BJCR.

# Comparación de BER

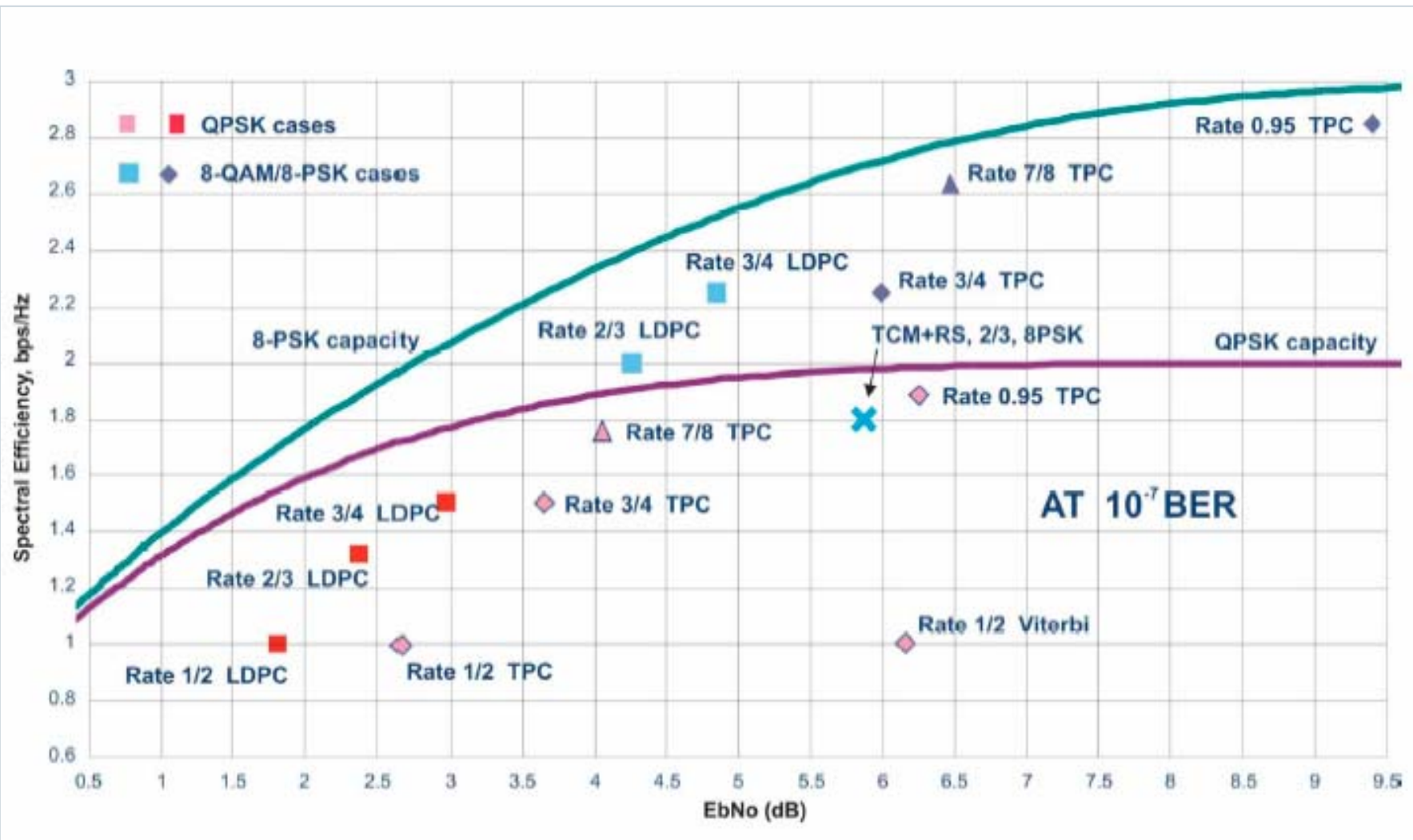




# Codificación LDPC

- Robert G. Gallager (1963)
- Códigos con matriz de chequeo de paridad de baja densidad
  - El número de elementos distintos de cero es reducido
  - La distancia mínima crece linealmente con la longitud de los bloques de código
- Lineales e iterativos
- Mejores prestaciones que los turbocódigos para la misma longitud
  - P.e., para tasa  $\frac{1}{2}$  y longitud 10.000.000 bits:  $\sim 0.0045$  dB por encima del límite de Shannon
- **Codificación**: a partir de la matriz de chequeo de paridad,
- **Decodificación**: de manera iterativa con algoritmos de tipo *belief propagation* (BP)

# Eficiencia espectral



# Sistemas comerciales: CDM-600

## Modems

Model	Description	Frequency	Data Rate	FEC	Modulation
<a href="#">SLM-5650</a>	<a href="#">Satellite Modem, MIL-188-165A</a>	50 – 90 MHz 100 – 180 MHz 950 – 2000 MHz	64 kbps – 52 Mbps (EIA-613) 64 kbps – >200 Mbps	Uncoded TPC Viterbi Reed-Solomon Trellis LDPC	BPSK QPSK OQPSK 8-PSK 16-QAM (8- and 64-QAM upgrade)
<a href="#">CDM-Qx</a>	<a href="#">Multi-Channel Satellite Modem</a>	50 – 90 MHz 100 – 180 MHz	10 kbps – 20 Mbps	TPC Viterbi	BPSK QPSK OQPSK 8-PSK 16-QAM

CDM-600 & CDM-600L Open Network Satellite Modems



# CDM-600



## Opciones de configuración:

TX-IF: CARRIER = ON (ON,OFF,RTI)  
TX FREQ=176.0000MHz SPECTRUM

OUTPUT POWER: MODE = MANUAL (MANUAL,AUPC)  
OUTPUT POWER LEVEL = -20.0 dBm

ENCODER = TPC (NONE,VIT,SEQ,TCM,TPC,LDPC)  
REED-SOLOMON= OFF (ON,OFF)

REED-SOLOMON ENCODING = ON(200/180)

MODULATION= QPSK (B,Q OQ,8PSK,16Q,8QAM)  
FEC RATE = 1/2 (1/2,3/4,7/8)

RECEIVE EbNo ALARM POINT = 02.0 dB

## Retardo de los esquemas FEC:

FEC Mode (64 kbps data rate)	End-to-end delay, ms
Viterbi, Rate 1/2	12
Sequential, Rate 1/2	74
Viterbi Rate 1/2 + Reed Solomon	266
Sequential Rate 1/2 + Reed Solomon	522
Turbo Product Coding, Rate 3/4, O/QPSK	47
Turbo Product Coding, Rate 21/44, BPSK	64
Turbo Product Coding, Rate 5/16, BPSK	48
Turbo Product Coding, Rate 7/8, O/QPSK	245 *
Turbo Product Coding, Rate 0.95, O/QPSK	69
LDPC Coding, Rate 1/2	248
LDPC Coding, Rate 2/3, O/QPSK	296
LDPC Coding, Rate 2/3, 8-PSK, 8-QAM	350
LDPC Coding, Rate 3/4, O/QPSK	321
LDPC Coding, Rate 3/4, 8-PSK, 8-QAM, 16-QAM	395