Digital Communications

Telecommunications Engineering

Chapter 6

Channel coding for error protection

Marcelino Lázaro

Departamento de Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid

Index of contents

Introduction

- Digital communication systems have errors during transmission
- Performance objective of the system
  - BER $<$ Required performance
- Alternatives to decrease probability of error
  - To increment signal energy (power)
    - ★ Limitations: Economical, physical, legal, interferencies, ...
  - Noisy-channel coding theorem (Shannon)
    - ★ Introduction of redundancy bits
    - ★ Coding rate: R (data bits/transmitted bits)
    - ★ Channel capacity: C (bits of information per channel use)
    - ★ BER can be made arbitrarily low as long as

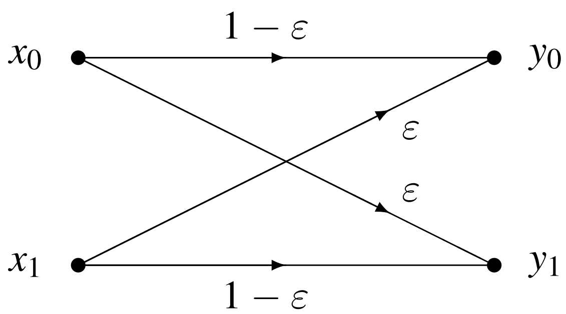
$$R < C$$

Channel capacity - Digital channels

- Discrete memoryless channel (DMC) model
  - Input and output: random variables X and Y
  - Transition (conditional) probabilities $p_{Y|X}(y_j|x_i)$
- Channel capacity

$$C = \max_{p_X(x_i)} I(X, Y) \text{ bits/uso}$$

  - Example: Binary symmetric channel with BER=ε

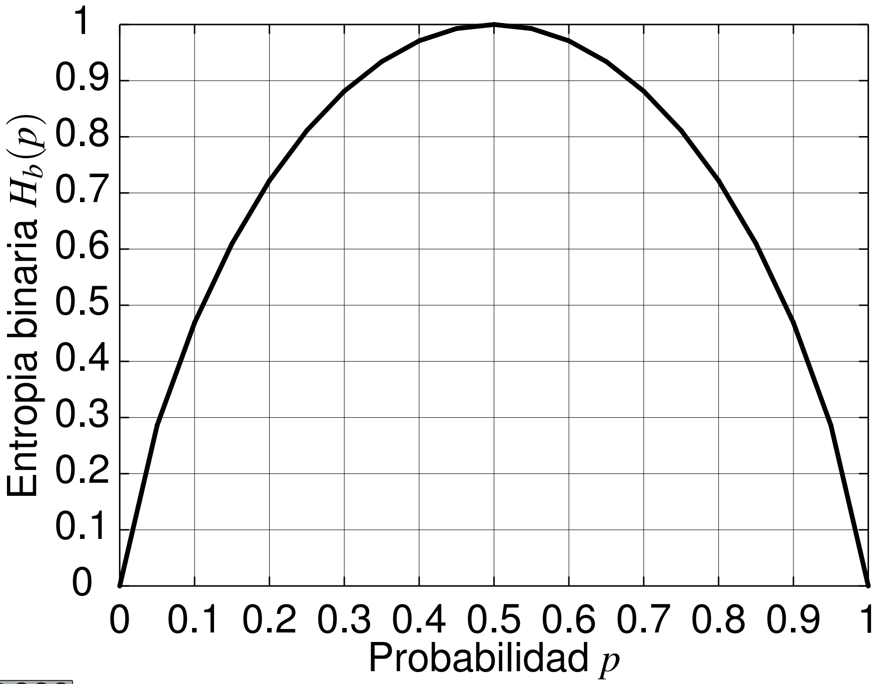


$$C = 1 - H_b(\varepsilon) \text{ bits/use}$$

Binary entropy $H_b(p)$

Entropy of a binary random variable with probabilities $p_X(x_0) = p$ y $p_X(x_1) = 1 - p$

$$H_b(p) = -p \cdot \log_2(p) - (1-p)\log_2(1-p)$$



Channel capacity - Gaussian channel

- The capacity of a Gaussian channels with the following parameters
  - Transmitted signal power: P watt.
  - Noise power: P_N watt.
  - Available channel bandwidth: B Hz

$$C = \frac{1}{n}\log M = \frac{1}{2}\log\left(1 + \frac{P}{P_N}\right)$$

$$P_N = \int_{-B}^{B} \frac{N_o}{2} df = N_o B$$

$$C = \frac{1}{2} \cdot \log\left(1 + \frac{P}{N_o B}\right) \text{ bits/uso}$$

$$C = B \cdot \log\left(1 + SNR\right) \text{ bits/s}$$

Bounds

- Channel capacity as a function of channel bandwidth B

$$\lim_{B\to\infty} C = \frac{P}{N_o}\log_2(e) = 1{,}44 \cdot \frac{P}{N_o}$$

- Practical communication system

$$R_b < B \cdot \log\left(1 + SNR\right) \text{ bits/s}$$

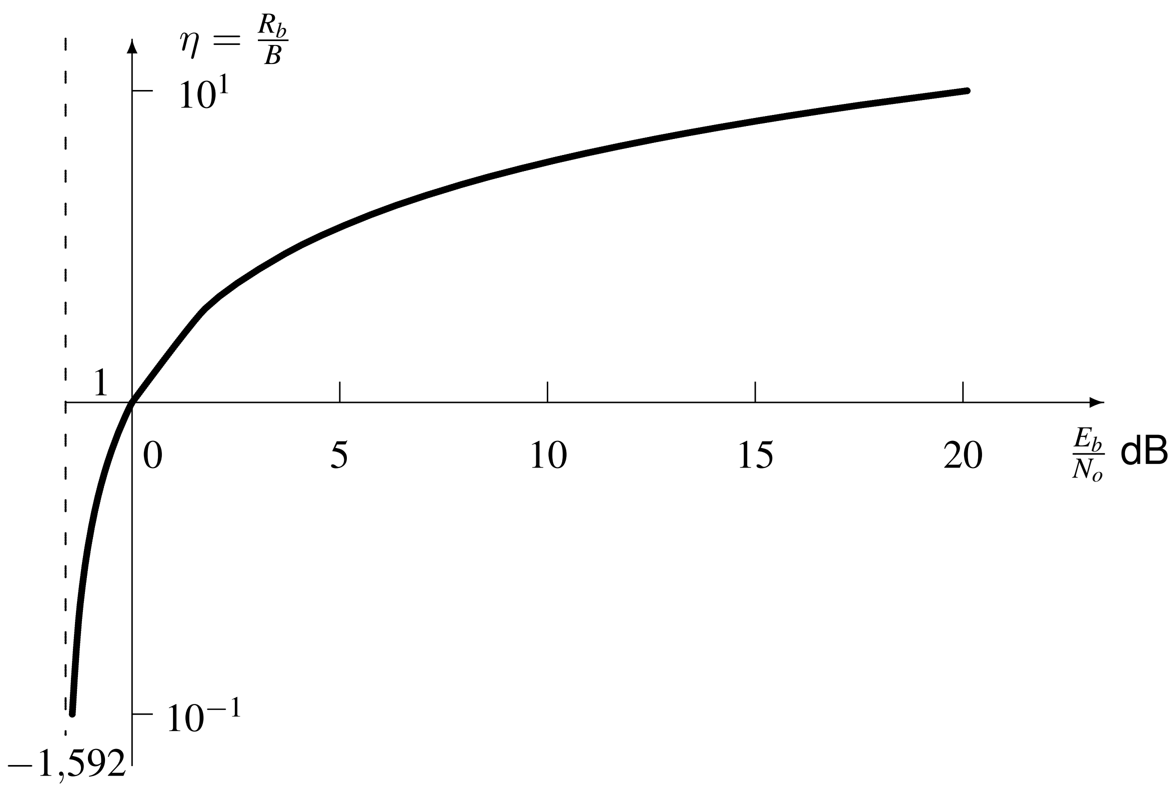
- Spectral binary rate (efficiency): $\eta = \frac{R_b}{B}$ bits/s/Hz
- Mean energy per bit - $E_b = \frac{P}{R_b}$
- Mean energy per bit per N_0 ratio: $E_b/N_0 = \frac{SNR}{\eta}$

$$\eta < \log\left(1 + SNR\right), \quad \eta < \log\left(1 + \eta \cdot \frac{E_b}{N_o}\right)$$

$$SNR > 2^\eta - 1, \quad \frac{E_b}{N_o} > \frac{2^\eta - 1}{\eta}$$

When $\eta \to 0$ $\quad \frac{E_b}{N_o} = \ln 2 = 0{,}693 \approx -1{,}6$ dB

Spectral binary rate vs E_b/N_0



Normalized signal to noise ratio

- Lower bound for SNR

$$SNR > 2^{\eta} - 1$$

- Definition of normalized SNR

$$SNR_{norm} = \frac{SNR}{2^{\eta} - 1}$$

- Lower bound for SNR_{norm}

$$SNR_{norm} > 1 \text{ (0 dB)}$$

Classification of channel codes

- Capability of code for detection and/or correction of errors
  - Codes with detection capability
  - Codes with correction capability
- Mechanism used to introduce redundancy
  - Block codes
    - Independent coding of data blocks of fixed size (k bits)
    - Key concept: distance between coded words
  - Convolutional codes
    - Continuous coding by using a digital filter bank
- Statistic used for detection
  - Hard output: decoding is performed from detected bits $\hat{B}[\ell]$
  - Soft output: decoding is performed from demodulator output $q[n]$
    - Better performance with higher complexity
  - Bit erasure: bits or symbols with high error probability are labeled instead of decided ("erased")

Coding gain

- Definition: for a code and a specific E_b/N_0 ratio (or equivalently a specific BER), difference in decibels over E_b/N_0 which is necessary to achieve the same BER without channel coding
- Coding gain allow to compare the performance of different channel coding techniques
- Codign gain depends on the objective BER (or on the E_b/N_0 work point)
- Gain can be negative for some rango of E_b/N_0 ratio

Block codes - Definitions

- Codign is performed independently for blocks of k bits
  - Conversion in coded blocks of de n bits $\rightarrow$ Coding rate $R = k/n$
- Notation used to represent blocks of bits
  - Information bits (uncoded):
$$\boldsymbol{b}_i = [b_i[0], b_i[1], \cdots, b_i[k-1]],\ i = 0, 1, \cdots, 2^k - 1$$
  - Coded bits:
$$\boldsymbol{c}_i = [c_i[0], c_i[1], \cdots, c_i[n-1]],\ i = 0, 1, \cdots, 2^k - 1$$
  - Dictionary of the code: message word $\rightarrow$ coded word
$$\boldsymbol{b}_i \rightarrow \boldsymbol{c}_i$$
- Weight of a word $w(\boldsymbol{c}_i)$
  - Number of ones in the word
- Hamming distance for two coded words $d_H(\boldsymbol{c}_i, \boldsymbol{c}_j)$
  - Number of bits wich are different
- Minimum distance of the code: d_{min}
  - Minimum Hamming distance for two different words in the code

Optimum decoding working with hard output

- The available observation conditioned to the transmission of $\boldsymbol{c}_i$ is

$$\boldsymbol{r} = \boldsymbol{c}_i + \boldsymbol{e},\ \boldsymbol{e} = [e[0], e[1], \cdots, e[n-1]]$$

- Probabilistic model for the error pattern if BER = ε

$$p_E(e[j]) = \varepsilon^{e[j]} \cdot (1-\varepsilon)^{1-e[j]} = \begin{cases} \varepsilon, & e[j] = 1 \\ 1-\varepsilon, & e[j] = 0 \end{cases}$$

- Likelihood (coditional probability of observation)

$$p_{\boldsymbol{r}|\boldsymbol{c}}(\boldsymbol{r}|\boldsymbol{c}_i) = \prod_{j=0}^{n-1} \varepsilon^{r[j]-c_i[j]} \cdot (1-\varepsilon)^{1-(r[j]-c_i[j])}$$

- Maximum likelihood (ML) estimation - Minimizing Hamming distance

$$\hat{\boldsymbol{c}}_i = \text{arg} \min_{\boldsymbol{c}_i} d_H(\boldsymbol{r}, \boldsymbol{c}_i)$$

Detection and correction capabilities with hard output

- Performance depends on Hamming distances
  - An error can not be detected if transmission errors transform a coded word in another coded word
  - An error happens when the number of erroneous in the transmission of the coded word makes the observation to be at lower Hamming distance of another coded word
- Performance is limited for minimum distance d_{min}
  - Capability of detecting

$$d = d_{min} - 1 \text{ errors}$$

  - Capability of correcting

$$t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor \text{ errors}$$

Optimum decoding with soft output

- Constellation: M symbols ($m = \log_2(M)$) bits/symbol
- Sequence of symbols for a coded word

$$\boldsymbol{c}_i \rightarrow \boldsymbol{A}_i = [A_i[0], A_i[1], \cdots, A_i[n'-1]],\ n' = \frac{n}{m}$$

- Probabilistic model for observation when $\boldsymbol{c}_i$ is transmitted

$$\boldsymbol{q} = \boldsymbol{A}_i + \boldsymbol{e},\ \boldsymbol{e} = [e[0], e[1], \cdots, e[n'-1]]$$

- Probabilistic model for error pattern

$$f_E(e[j]) = \mathcal{N}(0, \sigma_z^2)$$

- Likelihood (conditional probability of observation)

$$f_{\boldsymbol{q}|\boldsymbol{A}}(\boldsymbol{q}|\boldsymbol{A}_i) = \mathcal{N}(\boldsymbol{A}_i, \sigma_z^2)$$

- Maximum likelihood estimation - Minimizing euclidean distance

$$\hat{\boldsymbol{c}} = \arg \min_i d_e(\boldsymbol{q}, \boldsymbol{A}_i)$$

Linear block codes

- Channel code $C(k, n)$
- Basis for the code: k linearly independent coded words

$$\{\boldsymbol{g}_0, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_{k-1}\}$$
$$\boldsymbol{g}_i = [g_i[0], g_i[1], \cdots, g_i[n-1]]$$

- Coding methodology: linear combination of k basis

$$\boldsymbol{c}_i = b_i[0] \cdot \boldsymbol{g}_0 + b_i[1] \cdot \boldsymbol{g}_1 + \cdots + b_i[k-1] \cdot \boldsymbol{g}_{k-1}$$

  Coefficients for the expansion: information bits

- Properties
  - All elements in the basis are coded words (belong to the code)
  - $\boldsymbol{c}_0 = \boldsymbol{0} = [0, 0, \cdots, 0]$ word belongs to the code
  - A linear combination of coded words belongs to the code
  - All coded words have at least another coded word at minimum distance d_{min}
  - Therefore, as $\boldsymbol{c}_0$ has at least a coded word at d_{min}

$$d_{min} = \min_{\boldsymbol{c}_i \neq \boldsymbol{c}_0} w(\boldsymbol{c}_i)$$

Generator matrix

- The basis of the code can be packed in a matrix of size $k \times n$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_0[0] & g_0[1] & \cdots & g_0[n-1] \\ g_1[0] & g_1[1] & \cdots & g_1[n-1] \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1}[0] & g_{k-1}[1] & \cdots & g_{k-1}[n-1] \end{bmatrix}$$

- Coded words can be obtained from this matrix

$$\boldsymbol{c}_i = \boldsymbol{b}_i \cdot \boldsymbol{G}$$

- Systematic: message $\boldsymbol{b}_i$ is part of the coded word $\boldsymbol{c}_i$

$$\boldsymbol{c}_i = [\boldsymbol{b}_i \mid \boldsymbol{p}_i] \ \rightarrow \ \boldsymbol{G} = [\boldsymbol{I}_k \mid \boldsymbol{P}]$$

$$\boldsymbol{c}_i = [\boldsymbol{p}_i \mid \boldsymbol{b}_i] \ \rightarrow \ \boldsymbol{G} = [\boldsymbol{P} \mid \boldsymbol{I}_k]$$

Generator matrix - An example

- Code $C(2,5)$

$$\boldsymbol{G} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

- Words of the code

$\boldsymbol{b}_i$	$\boldsymbol{c}_i$
0 0	0 0 0 0 0
0 1	1 0 1 0 1
1 0	0 1 1 1 0
1 1	1 1 0 1 1

- Systematic codes
- Minimum distance of the code: $d_{min} = 3$
  - Detects 2 errors
  - Corrects 1 error

Parity check matrix

- Matrix of size $(n-k) \times n$: orthogonal complement for $\boldsymbol{G}$

$$\boldsymbol{G} \cdot \boldsymbol{H}^T = \boldsymbol{0} \text{ matriz de } k \times (n-k) \text{ ceros}$$

- Systematic codes

$$\boldsymbol{G} = [\boldsymbol{I}_k \mid \boldsymbol{P}] \;\rightarrow\; \boldsymbol{H} = [\boldsymbol{P}^T \mid \boldsymbol{I}_{n-k}]$$

$$\boldsymbol{G} = [\boldsymbol{P} \mid \boldsymbol{I}_k] \;\rightarrow\; \boldsymbol{H} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{P}^T]$$

- Identification of coded words

$$\boldsymbol{c}_i \cdot \boldsymbol{H}^T = \boldsymbol{b}_i \cdot \boldsymbol{G} \cdot \boldsymbol{H}^T = \boldsymbol{0} \text{ vector of } n-k \text{ zeros}$$

- Syndrome-based decoding
  - Transmission model: $\boldsymbol{r} = \boldsymbol{c}_i + \boldsymbol{e}$ ($\boldsymbol{e}$: error pattern)
  - Syndrome depends only on the error pattern

$$\boldsymbol{s} = \boldsymbol{r} \cdot \boldsymbol{H}^T = (\boldsymbol{c}_i + \boldsymbol{e}) \cdot \boldsymbol{H}^T = \boldsymbol{e} \cdot \boldsymbol{H}^T$$

  - Decodiing can be made from a syndrome table

Syndrome table - An example

$$\boldsymbol{G} = \left[\begin{array}{ccc|cc} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{array}\right] \rightarrow \boldsymbol{H} = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{array}\right]$$

- Syndrome table

$\boldsymbol{e}$	$\boldsymbol{s}$
0 0 0 0 0	0 0 0
1 0 0 0 0	1 0 0
0 1 0 0 0	0 1 0
0 0 1 0 0	0 0 1
0 0 0 1 0	0 1 1
0 0 0 0 1	1 0 1
¿?	1 1 0
¿?	1 1 1

$\boldsymbol{s} = 110 \rightarrow \boldsymbol{e}_1 = 11000,\ \boldsymbol{e}_2 = 00011,\ \boldsymbol{e}_3 = 10110,\ \boldsymbol{e}_4 = 01101$

Best option: to choose one of the patterns with two erroneous bits

Advantage of working with G and H

- Number of coded words can be high: 2^k
  - $k = 2$, $n = 5$: 4 coded words
  - $k = 247$, $n = 255$: $2^{247} \approx 2{,}26 \cdot 10^{74}$ coded words
- Number of symdromes
  - $k = 2$, $n = 5$ $(t = 1)$: 8 syndromes
  - $k = 247$, $n = 255$ $(t = 1)$: 256 syndromes

Gaussian elimination method - Example

- It can be used to transform a non-systematic generator matrix in a systematic-one
- Rows are replaced by linear combinations of rows
  - 1^{st} row: $1^{st} + 2^{nd}$ rows
  - 2^{nd} row: 1^{st} row
- Code $C(2,5)$

$$\boldsymbol{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

- Coded words

$\boldsymbol{b}_i$	$\boldsymbol{c}_i$
0 0	0 0 0 0 0
0 1	0 1 1 1 0
1 0	1 1 0 1 1
1 1	1 0 1 0 1

- Same coded words / different assignation to information bits
  - The same $\boldsymbol{H}$ can be used to generate syndrome table

Hamming bound

- The number of symdromes with a redundancy of $r = n - k$ bits is
$$2^{n-k} = 2^r$$
- Hamming bound: to correct t errors, the minimum necessary redundancy is
$$r \geq \log_2 V(n,t),\ \ V(n,t) = \sum_{j=0}^{t} \binom{n}{j}$$
  - $V(n,t)$: Hamming sphere with radius t
- Interpretation: the number of available symdromes is
$$\binom{0}{0} + \binom{n}{1} + \cdots + \binom{n}{t} \leq 2^r$$
  - Equality: Perfect codes

Perfect codes

- Repetition codes (majority decission)
  - Odd n, $k = 1$, $t = (n-1)/2$
- Hamming codes
  - $n = 2^m - 1$, $k = 2^m - m - 1$, $t = 1$
  - Parity check matrix: in the columns they can be found all the possible binary combinations $(n-k)$ bits, except all zeros
    - ⋆ Example: Hamming code (4,7)

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

- Golay code
  - $n = 23$, $k = 11$, $t = 3$

Performance - Hard decoding

- Assumption: bit error rate (BER) in transmission: ε
- Errors happen whe the correction capability is exceeded
- Codes with correction capability of t errors in the block of n bits

$$P_e = \sum_{j=t+1}^{n} \binom{n}{j} \cdot \varepsilon^j \cdot (1-\varepsilon)^{n-j}$$

- Codes all patterns of t errors and a patterns of $t+1$ errors

$$P_e = \left[\binom{n}{t+1} - a\right] \cdot \varepsilon^{t+1} \cdot (1-\varepsilon)^{n-t-1} + \sum_{j=t+2}^{n} \binom{n}{j} \cdot \varepsilon^j \cdot (1-\varepsilon)^{n-j}$$

- With Gray or pseudo-Gray bit assignement and high SNR

$$BER \approx \frac{1}{k} P_e$$

Performance - Soft decoding

- Probability of error

$$P_e \approx c \cdot Q\left(\frac{d^e_{min}}{2\sqrt{N_0/2}}\right)$$

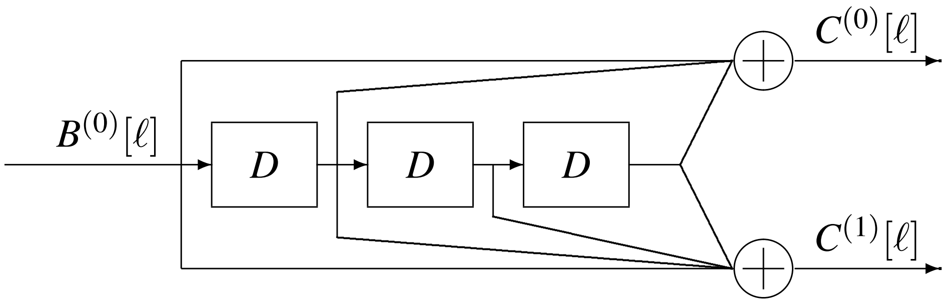
- d^e_{min}: minimum euclidean distance of symbol sequences corresponding to two different coded words
- c: maximum number of coded words at minimum distance of another coded word
  - For binary modulations

$$d^e(\boldsymbol{c}_i, \boldsymbol{c}_j) = d(\boldsymbol{a}_0, \boldsymbol{a}_1)\sqrt{d_H(\boldsymbol{c}_i, \boldsymbol{c}_j)}$$

$$P_e \approx c \cdot Q\left(\frac{d(\boldsymbol{a}_0, \boldsymbol{a}_1)}{2\sqrt{N_0/2}}\sqrt{d_{min}}\right)$$

Convolutional channel codes

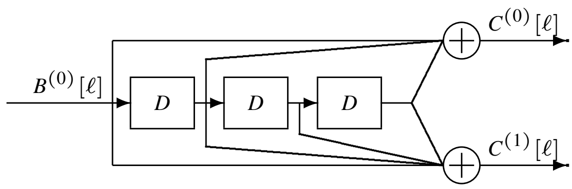
- Redundancy is introduced by digital filtering
  - Introduction of memory in the transmission
- Rate $R = k/n$: filter bank with
  - k inputs
  - n outputs



- Notation:
  - Sequences for inputs: $B^{(i)}[\ell]$, con $i = 0, 1, \cdots, k-1$
  - Sequences for outpus: $C^{(j)}[\ell]$, con $j = 0, 1, \cdots, n-1$

Representation of convolutional codes

- Schematic representation of the filter bank



- Analytic relationship of all outputs with respect to inputs

$$C^{(0)}[\ell] = B^{(0)}[\ell] + B^{(0)}[\ell-1] + B^{(0)}[\ell-3]$$

$$C^{(1)}[\ell] = B^{(0)}[\ell] + B^{(0)}[\ell-1] + B^{(0)}[\ell-2] + B^{(0)}[\ell-3]$$

- D polinomial representation of sequences

$$B^{(i)}(D) = \sum_{\ell} B^{(i)}[\ell] \cdot D^{\ell}$$

  - Property of D representation with respecto to delays

$$B^{(i)}[\ell - d] \leftrightarrow B^{(i)}(D) \cdot D^{d}$$

Representations of convolutional codes (II)

- Notation using D polinomial representation

$$C^{(0)}(D) = B^{(0)}(D)\left\{1 + D + D^3\right\}$$

$$C^{(1)}(D) = B^{(0)}(D)\left\{1 + D + D^2 + D^3\right\}$$

- Matrix notation (polinomial):

$$\boldsymbol{C}(D)_{1\times n} = \boldsymbol{B}(D)_{1\times k} \cdot \boldsymbol{G}(D)_{k\times n}$$

- Generator matrix of size $k \times n$
  - Element in row i column j: contribution into j-th output coming from i-th input
  - Examples
    - Previous example (A): $k = 1$, $n = 2$

$$\boldsymbol{G}(D) = \left[1 + D + D^3,\ 1 + D + D^2 + D^3\right]_{1\times 2}$$

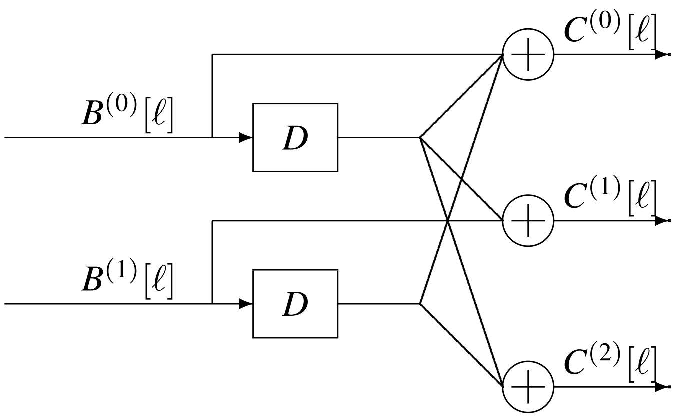
    - Another example (B): $k = 2$,

$$\boldsymbol{G}(D) = \begin{bmatrix} 1+D & D & D \\ D & 1 & D \end{bmatrix}_{2\times 3}$$

Translation to squematic representation - Example B

$$\boldsymbol{G}(D) = \begin{bmatrix} 1+D & D & D \\ D & 1 & D \end{bmatrix}_{2\times 3}$$

- Number of inputs of the bank: $k = 2$
- Number of outputs of the bank: $n = 3$
- Number of memories in each input: maximum degree of all polinomials in this row



Parameters of interest

- Total memory of the code: M_t
  - Total number of delay units (memories)

$$M_t = \sum_{i=0}^{k-1} M^{(i)}$$

  - Memory of the i-th input:

$$M^{(i)} = \max_j \text{degree}(g_{i,j}(D))$$

- Constraint length: K
  - Length of the inpulse response of the encoder (maximum number of discrete time instants where an input bit contributes to any system output)

$$K = 1 + \max_{i,j} \text{degree}(g_{i,j}(D))$$

  - In general, performance improves with higher K values

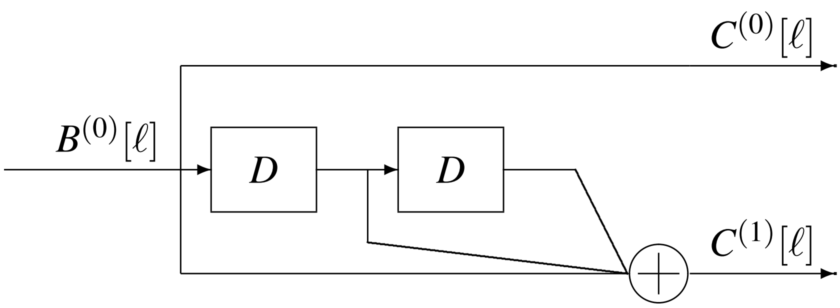
Systematic codes

- Generator matrix

$$\boldsymbol{G}(D) = [\boldsymbol{I}_k \mid \boldsymbol{P}(D)]$$

  - Inputs are "replicated" in a subset of outputs
  - Example (C)

$$\boldsymbol{G}(D) = \left[1,\ 1 + D + D^2\right]$$



Trellis diagram

- Definition for the system state
  - Contents of the memories

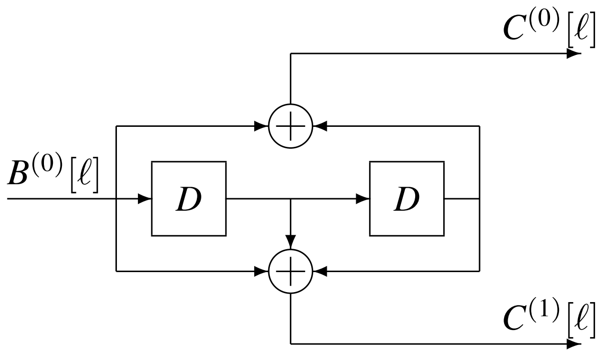
$\psi[\ell] = [B^{(0)}[\ell-1], \cdots, B^{(0)}[\ell - M^{(0)}], \cdots, B^{(k-1)}[\ell-1], \cdots, B^{(k-1)}[l - M^{(k-1)}]]$

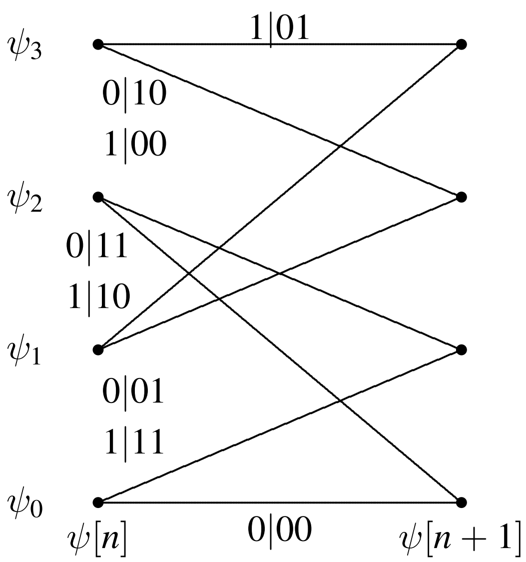
- Labelling of the trellis
  - Bits at the input (uncoded)
  - Bits at the output (encoded)

$$B^{(0)}[\ell], B^{(1)}[\ell], \cdots B^{(k-1)}[\ell] \; \Big| \; C^{(0)}[\ell], C^{(1)}[\ell], \cdots C^{(n-1)}[\ell]$$

Example - Convolutional D

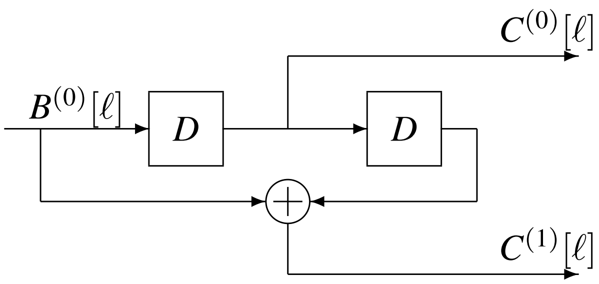
- State: $\psi[\ell] = \left[B^{(0)}[\ell-1], B^{(0)}[\ell-2]\right]$
- Values: $\psi_0 = [0,0]$, $\psi_1 = [1,0]$, $\psi_2 = [0,1]$, $\psi_3 = [1,1]$

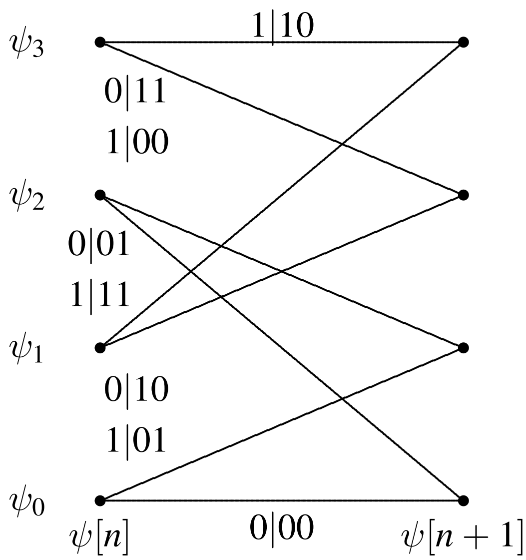




Example - Convolutional E

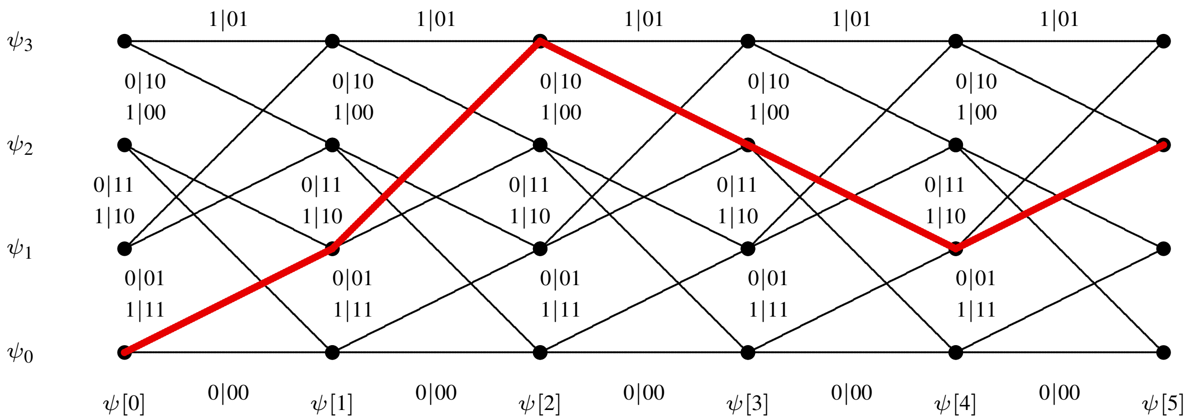
- State: $\psi[\ell] = \left[B^{(0)}[\ell-1], B^{(0)}[\ell-2]\right]$
- Values: $\psi_0 = [0,0]$, $\psi_1 = [1,0]$, $\psi_2 = [0,1]$, $\psi_3 = [1,1]$





Sequence of bits - Path through the trellis

- Sequence of data: $B^{(0)}[\ell] = [11010]$
- Initial: ψ_0 (can be fixef with a bit header)



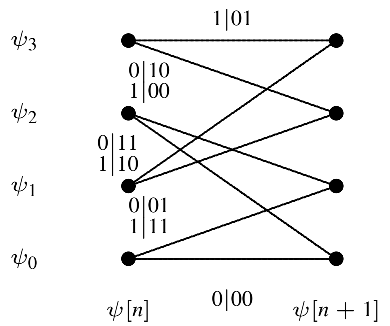
- Encoded sequence: $C[k] = [11\ 10\ 10\ 00\ 01]$

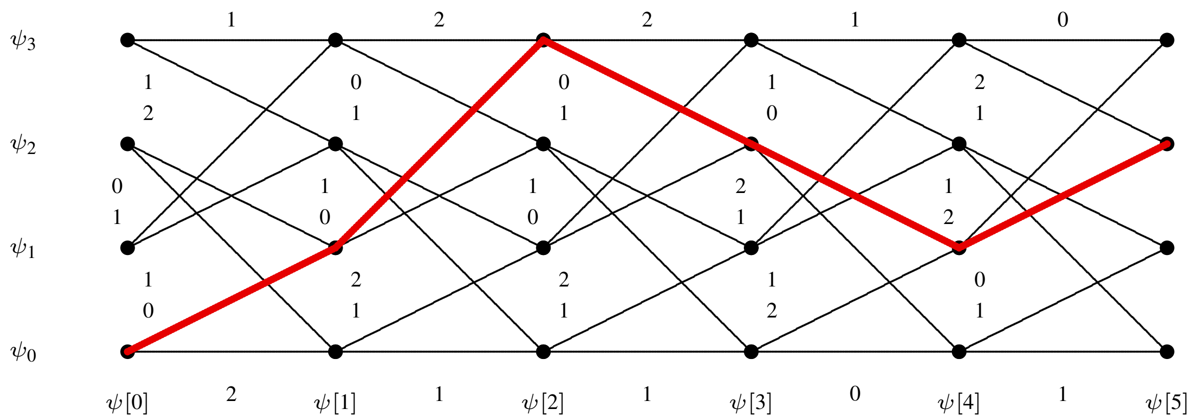
Decoding - Viterbi algorithm

- Recovers the maximum likelihood (ML) data sequence
- Estados inicial/final
  - Reference header (usually zeros “*bit flushing*”)
- Hard output: observation of detected bits
  - Sequence with the lowest number of encoded bits being different than the observed bits
  - Branch metric: Hamming distance with respect to the observation
- Soft output: observation of demodulator output $q[n]$
  - Sequence whos associated transmitted symbols are at minimum euclidean distance of observation
  - Branch metric: $|q[n] - A[n]|^2$
    - ⋆ The constellation used to transmit information has to be taken into account to translate trellis labels to symbols of the constellation ($A[n]$)
  - Better performance than working with hard output

Branch metric - Hard output - Example

- Received sequence: $R[k] = [11\ 10\ 10\ 00\ 01]$





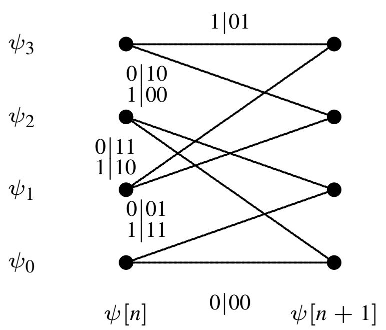
Accumulated metrics for each state (Viterbi)

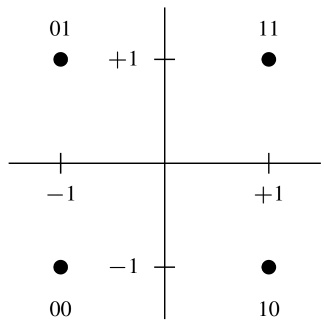
	$\psi[1]$	$\psi[2]$	$\psi[3]$	$\psi[4]$	$\psi[5]$
ψ_3	–	0	**2**/3	**3**/4	3/**2**
ψ_2	–	2	**0**/5	**3**/4	0/**0**
ψ_1	0	3	**3**/4	**0**/5	4/**3**
ψ_0	2	3	3/**4**	**2**/3	4/**3**

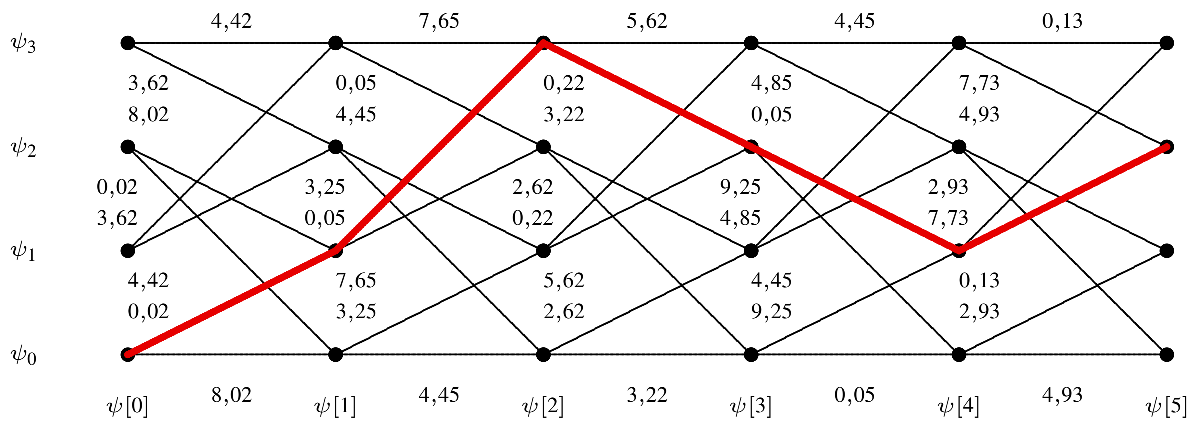
- Transmission without errors: one survival path has null accumulated metric

Branch metric - Soft output - Example

- Received sequence: $q[n] = \begin{bmatrix} 1{,}1 \\ 0{,}9 \end{bmatrix} \begin{bmatrix} 1{,}1 \\ -0{,}8 \end{bmatrix} \begin{bmatrix} 0{,}75 \\ -0{,}6 \end{bmatrix} \begin{bmatrix} -1{,}2 \\ -1{,}1 \end{bmatrix} \begin{bmatrix} -0{,}7 \\ 1{,}2 \end{bmatrix}$







Accumulated metric for each state (Viterbi)

	$\psi[1]$	$\psi[2]$	$\psi[3]$	$\psi[4]$	$\psi[5]$
ψ_3	–	0,07	**5,69**/11,49	**10,14**/15,74	10,27/**8,07**
ψ_2	–	7,67	**0,29**/16,89	**10,54**/15,34	17,87/**0,47**
ψ_1	0,02	11,27	**10,89**/15,09	**0,34**/19,54	15,47/**12,47**
ψ_0	8,02	12,47	**10,29**/15,69	**9,54**/10,34	**13,47**/14,47

Performance

- Hard output

$$Pe \approx c \cdot \sum_{i=t}^{n\cdot z} \binom{n\cdot z}{i} \cdot \varepsilon^i \cdot (1-\varepsilon)^{n\cdot z-i}$$

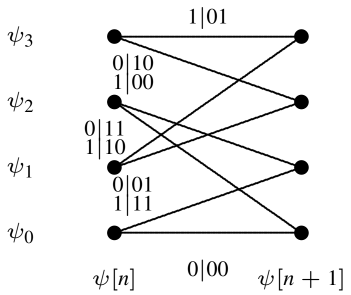
  - D_{min}^H: minimum Hamming distance for different coded sequences
  - z: length of erroneous event of minimum distance D_{min}^H
  - $t = \left\lfloor \frac{D_{min}^H - 1}{2} \right\rfloor$ (correction capability over $n \times z$ bits)
  - ε: bit error rate during transmission (BER)
- Soft output

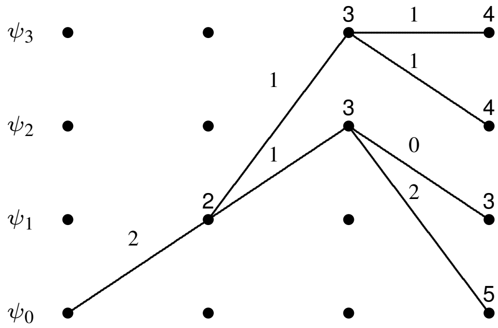
$$P_e \approx c \cdot Q\left(\frac{D_{min}^e}{2\sqrt{N_0/2}}\right)$$

  - D_{min}^e: minimum euclidean distance between sequences of symbols transmitted for two different data sequences

Evaluation for D_{min}^H

- Comparison with the encoded sequence of all zeros
- Viterbi algorithm can be used to evaluate distances over erroneous events





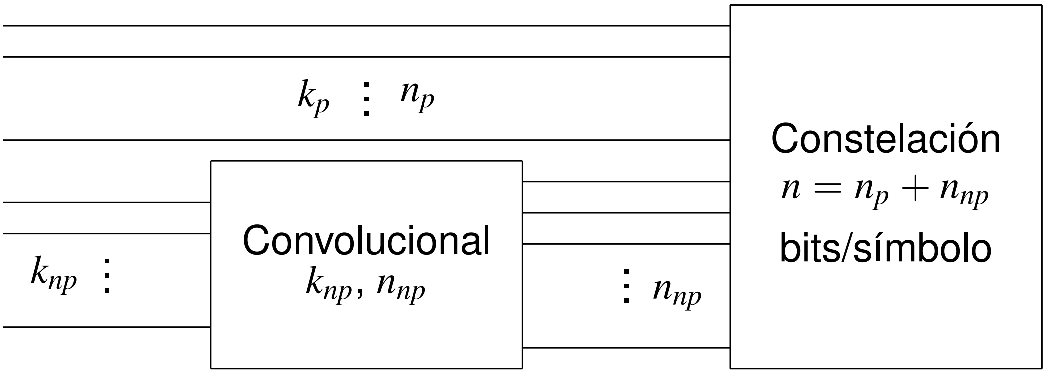
$$D_{min}^H = 5$$

$$z = 3$$

Trellis codes

- Also known as TCM codes
  - *Trellis Coded Modulation*
- Joint design of convolutional encoders and transmitter encoder (bit assignement in the system constellation)
- Efficient use of available channel bandwidth
  - Redundancy is introduced by augmenting the order of the constellation
    - Constellation order is matched to the size of the encoder (number of outputs)
- Possibility of relatively high rates by using the “parallel transitions”

Trellis codes - Definitions



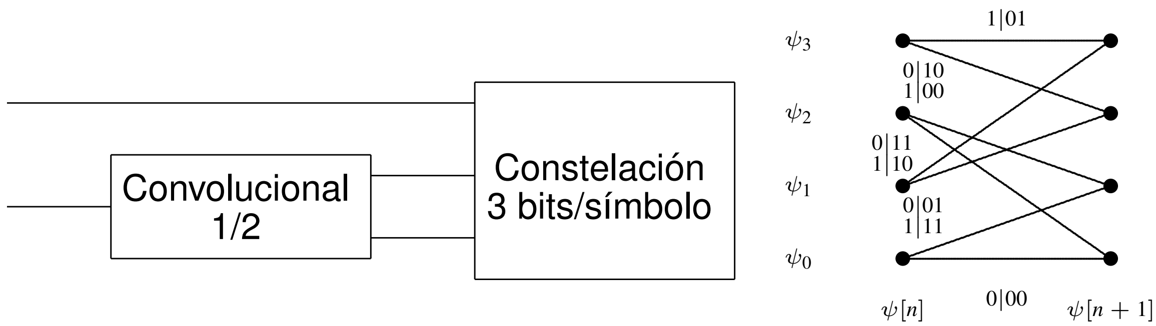
- k inputs
  - k_p "parallel transitions" (unprotected)
  - k_{np} "non-paralel transitions" (protected)
- n output
  - $n_p = k_p$ "parallel transitions" (unprotected)
  - n_{np} "non-paralel transitions" (protected)
- Constellation: 2^n symbols ($n = n_p + n_{np}$ bits/symbol)

Trellis codes - Bit assignement

- Design rules: Ungerboeck rules
  - Bit assignement to minimize the probability of error
  - Constellation division + bit assignement
- Constellation is divided in sub-constellations
  - Sequential division in two halves augmenting minimum distance at each step
  - Final number of sub-constellations: $2^{n_{np}}$
  - Final number of symbols per su-constellation: 2^{n_p}
- Bit assignement
  - Parallel transitions: select a symbol in a sub-constellation
    - ⋆ "Physical" protection over constellation (Gray assignement)
  - Non-parallel transitions: selection of a sub-constellation
    - ⋆ Protection by means of the convolutional encoder
    - ⋆ Maximum distance: branches going out of or comming to the same state
    - ⋆ Equiprobability in symbol (sub-constellation) assignement

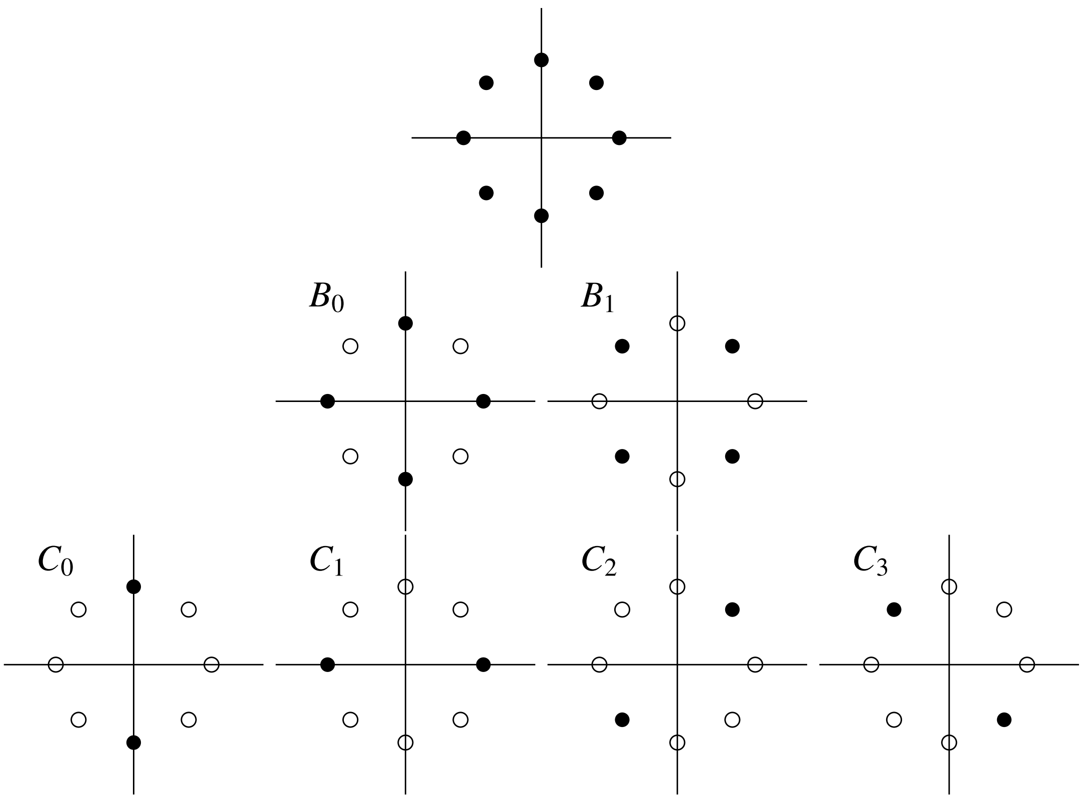
Example: $k_p = 1$, $k_{np} = 1$, $n_{np} = 2$, 8-PSK

- Coding rate 2/3
  - One parallel transition
  - Convolutional encoder with coding rate 1/2 (Convolutional of example D)

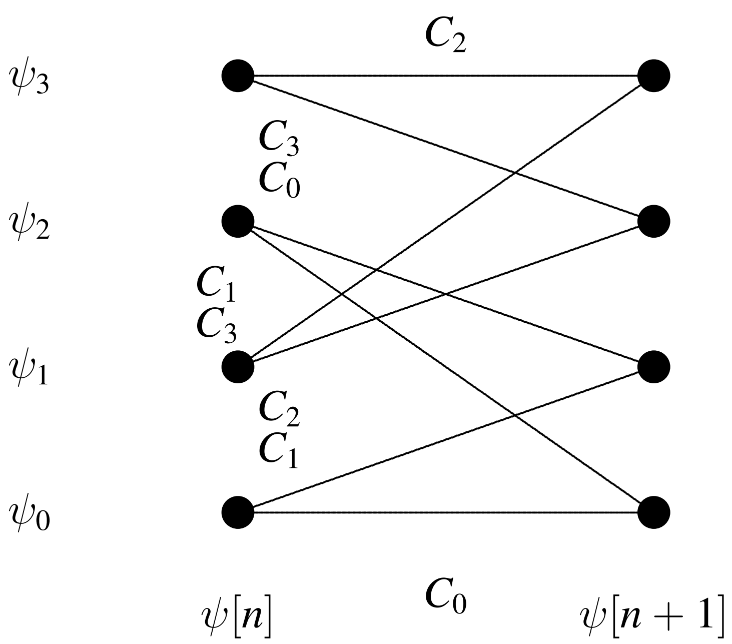


- Constellation with 3 bits per symbol: 8 symbols
  - 8-PSK

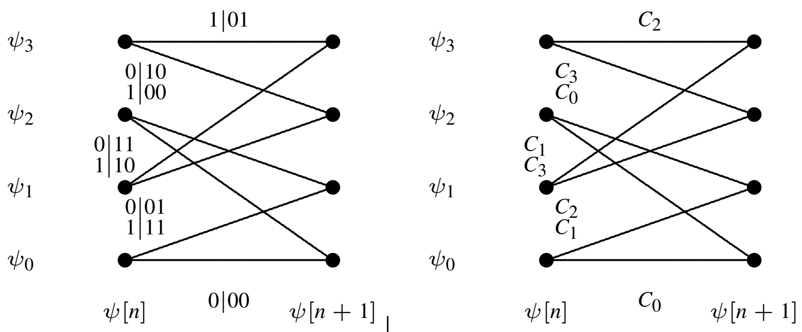
Division of the constellation - Example

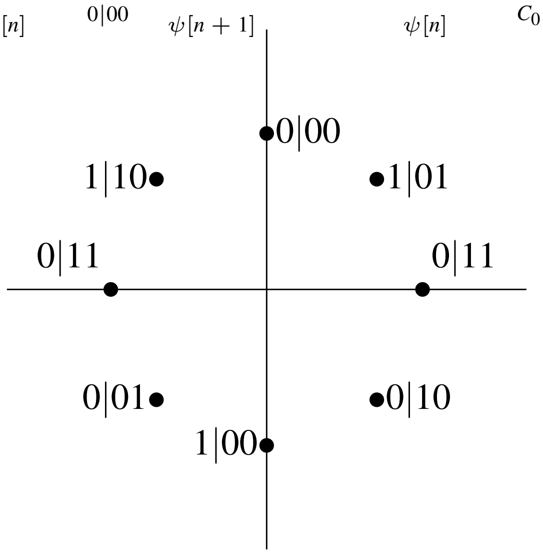


Assignement in the non-parallel transmitions - Example



Final assignement - Example (convolutional D)





Performance

- Minimum distance of the code - The minimum of:
  - Minimum parallel distance
  - Example (for normalized constellation): $d_p = 2$
    - ⋆ Minimum distance between points in sub-constellations
  - Minimum non-parallel distance
    - ⋆ Minimum distance of the convolutional encoder measured over assigned sub-constellations
    - ⋆ Example

$$d_{np} = \sqrt{d^2(C_0, C_1) + d^2(C_0, C_2) + d^2(C_0, C_1)} = 2{,}14$$

- Coding gain
  - Comparison with the equivalent constellation without channel coding (with the same effective transmission rate)
  - Example: reference constellation 4-PSK

$$G = \frac{E_b^{SC}}{E_b^{TCM}} \cdot \frac{\left(d_{min}^{TCM}\right)^2}{\left(d_{min}^{SC}\right)^2} = 2 \text{ (3 dB)}$$