

Instituto Superior de Engenharia de Lisboa
Engenharia Informática e de Computadores
Secção de Análise de Sinais
Compressão e Codificação de Dados

Codificação de canal - códigos lineares de bloco

Artur Ferreira {arturj@cc.isel.ipl.pt}

3 Junho 2004

Versão 1.0

Índice

1	Introdução	1
2	Codificação de canal	1
3	Caracterização de canal	2
3.1	Canal físico	2
3.2	Canal	3
3.3	Teorema da codificação de canal e capacidade de canal	5
3.3.1	Analogia com o teorema da codificação de fonte	5
3.3.2	Capacidade de transferência de informação	6
3.3.3	Capacidade do BSC	6
3.3.4	Cálculo alternativo da capacidade do BSC	8
4	Códigos de codificação de canal	9
4.1	Caracterização do codificador e decodificador	9
4.1.1	Codificador	9
4.1.2	Decodificador	10
4.2	Características dos códigos de bloco	10
4.3	Código de repetição (3,1)	11
4.4	Código bit de paridade par (3,2)	12
4.5	Códigos lineares	13
4.5.1	Características	14
4.5.2	Códigos de Hamming	15
4.5.3	Sub-espço vectorial	16
4.6	Descodificação	16
4.6.1	Tabela de síndromas	17
4.6.2	Exemplos de descodificação	18
4.6.3	Cálculo da distância mínima	18
4.7	Código de Hamming (7,4) não sistemático	19
4.8	Códigos lineares modificados	19
4.8.1	Extensão	20
4.8.2	Redução	21
4.8.3	Código dual	21
4.9	Análise comparativa de códigos	22
4.10	Aplicações	23

5	Utilização do MATLAB - Communications Toolbox	24
5.1	Codificação e decodificação	24
5.2	Verificação do teorema da codificação de canal	25
6	Apêndice: Teoria da Informação	26

1 Introdução

Este documento apresenta o conceito de codificação de canal. Caracterizam-se canais de comunicação através de modelos assentes em probabilidades e variáveis aleatórias (v.a.). Introduzem-se os códigos detectores e correctores de erros e as motivações para a sua existência. Descrevem-se os códigos, bem como os processos de codificação e decodificação e apresentam-se exemplos. Tratam-se códigos lineares de bloco sistemáticos e não sistemáticos, referindo aplicações dos mesmos.

Na secção 2 introduz-se o conceito de codificação de canal, enquadrando-o com a codificação de fonte. Na secção 3, apresenta-se o teorema da codificação de canal e a capacidade de canal. A secção 4 trata a construção de códigos, as características elementares dos códigos, os códigos lineares de bloco binários, com especial ênfase nos códigos de Hamming, concluindo com exemplos de aplicações dos códigos. A secção 5 conclui o documento com simulações em MATLAB, incluindo a verificação experimental do teorema da codificação de canal.

2 Codificação de canal

A figura 1 apresenta o cenário de utilização da codificação de fonte e codificação de canal. Os símbolos produzidos por determinada fonte são codificados numa representação com pouca redundância. A codificação de fonte (*source coding*) realiza a descrição de acontecimentos produzidos por determinada fonte, procurando minimizar o comprimento médio das palavras do código estabelecido [2, 8, 9], de tal forma que esse comprimento médio tenda para a entropia da fonte.

A codificação de canal (*channel coding*) adiciona redundância à mensagem para fazer face aos erros causados pela passagem no canal (de transmissão/armazenamento) ruidoso. Desta forma, espera-se conseguir transmissão isenta de erros, através de canal ruidoso. A figura 2

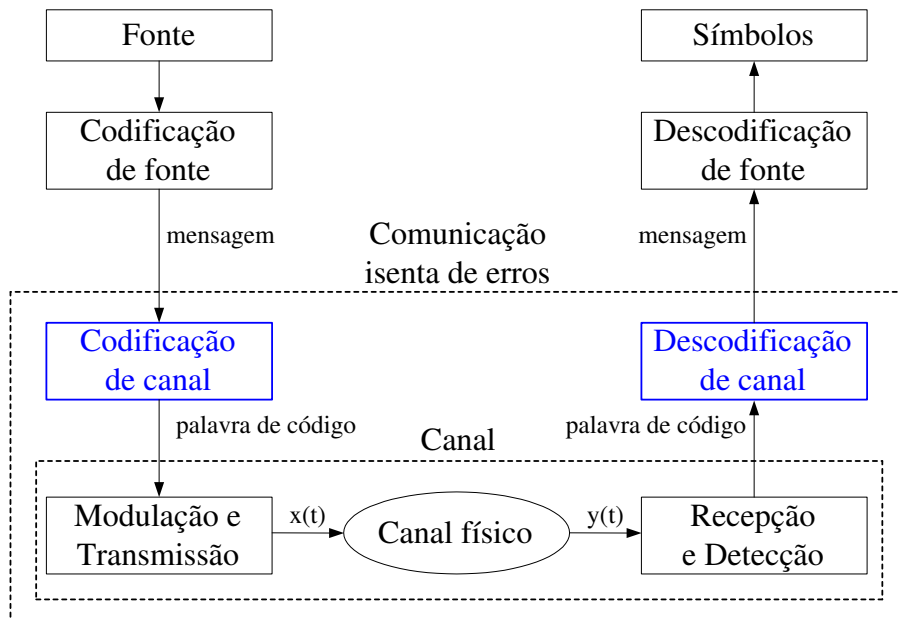


Figura 1: Enquadramento da codificação de fonte e codificação de canal no processo de comunicação/armazenamento.

relaciona a codificação de fonte e a codificação de canal, estabelecendo a terminologia utilizada. O codificador de fonte produz códigos que do ponto de vista do codificador de canal consiste na

mensagem a transmitir. O codificador de canal mapeia as mensagens em palavras de código. As palavras de código são enviadas para o canal físico de transmissão, após modulação. Estes

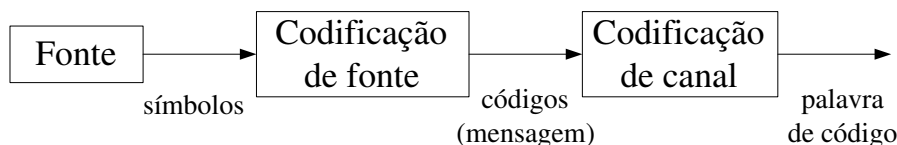


Figura 2: Relação entre codificação de fonte e codificação de canal.

codificadores têm objectivos contrários. O codificador de fonte representa os símbolos, através de um código, minimizando o comprimento médio deste, de tal forma que tenda para a entropia da fonte. Procura assim obter uma representação com pouca (ou nenhuma) redundância. O codificador de canal introduz redundância com o objectivo de recuperar a mensagem original após transmissão sobre canal físico ruidoso. O decodificador de canal recebe a palavra de código (eventualmente com erros) e procura recuperar a mensagem transmitida. Geralmente, o canal físico é inacessível, pelo que se justifica a introdução de redundância à priori na palavra a transmitir pelo canal.

Neste documento tratam-se códigos de canal, os processos de codificação e de decodificação, sem detalhar aspectos relativos à constituição do canal físico e ao tipo de modulação digital utilizado na transmissão sobre este.

3 Caracterização de canal

Nesta secção, caracterizam-se os canais de comunicação/armazenamento em termos genéricos. Apresentam-se as causas da existência de erros na comunicação. Abordam-se as características dos canais físicos e lógicos.

3.1 Canal físico

A correcção e/ou detecção de erros é necessária devido aos erros introduzidos no canal físico de transmissão ou de armazenamento. Geralmente, o canal é inacessível e como tal não é possível alterar características deste para evitar ou minimizar a existência de erros. Os erros são causados essencialmente pela conjugação de dois factores:

- atenuação do sinal transmitido;
- interferência.

O sinal contínuo $x(t)$ colocado na entrada do canal físico é sujeito a estes factores, de tal forma que no outro extremo do canal se tem um sinal distorcido $y(t)$. A detecção sobre $y(t)$ pode conduzir a bits errados.

O modelo AWGN (*Additive White Gaussian Noise*), apresentado na figura 3, é normalmente utilizado para representar o comportamento de canais físicos. Ao sinal eléctrico colocado na entrada do canal, é-lhe adicionado ruído branco, com distribuição de amplitude gaussiana. O valor β representa a atenuação imposta sobre $x(t)$.

O modelo de ruído branco, com densidade espectral plana ($S_n(f) = \eta$), justifica-se pelo facto de que a potência do ruído afecta de igual forma todas as frequências. A distribuição de amplitude gaussiana é justificada pelo teorema do limite central [12] que estabelece o seguinte:

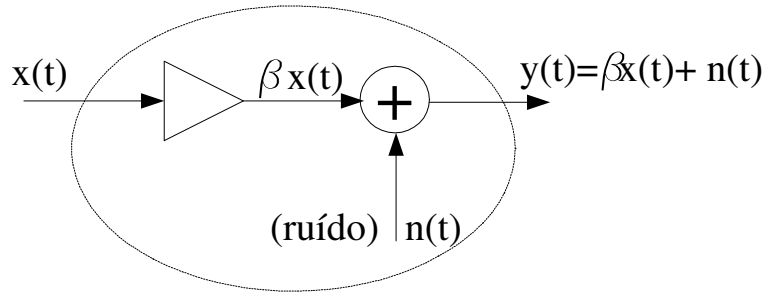


Figura 3: Representação do canal físico: modelo AWGN (*Aditive White Gaussian Noise*).

Dado um conjunto de g v.a. independentes e identicamente distribuídas (*iid*), a densidade de probabilidade da soma dessas variáveis tende para a distribuição gaussiana, quando $g \rightarrow \infty$.

A figura 4 mostra a função densidade de probabilidade (fdp) da soma de g v.a. independentes, com distribuição uniforme, média nula e variância unitária. Consideram-se os valores de $g \in \{2, 3, 7\}$.

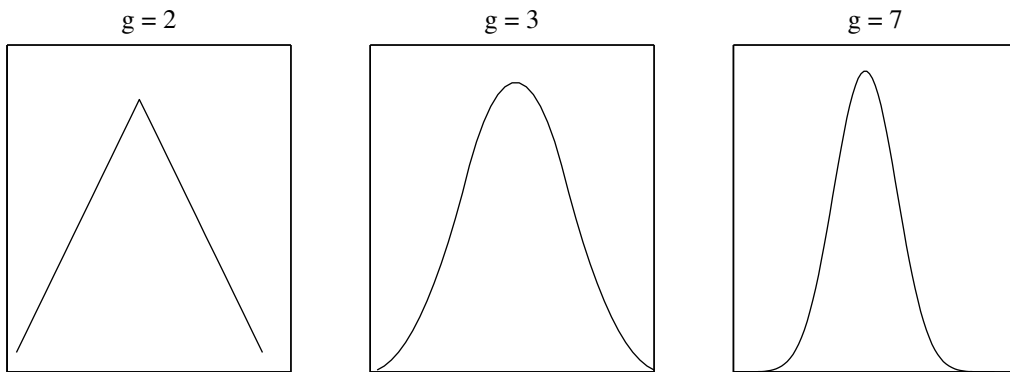


Figura 4: Ilustração do teorema do limite central: fdp da soma de $g \in \{2, 3, 7\}$ v.a. independentes e identicamente distribuídas.

O teorema do limite central é a razão principal pela qual muitos fenómenos são modelados através de v.a. com distribuição gaussiana. O ruído aditivo na comunicação é causado pela soma de contribuições elementares independentes de elevado número de termos. É portanto natural que seja modelado através da distribuição gaussiana.

3.2 Canal

Dado que se realiza comunicação binária, o canal pode ser analisado através de modelo discreto assente em v.a. e probabilidades. Assumindo que o canal não tem memória, ou seja, a transmissão de um bit é independente das transmissões dos bits anteriores, obtém-se o modelo de canal binário simétrico (*BSC - binary symmetric channel*). A designação de simétrico ilustra o facto de que a probabilidade de errar o bit 0 é igual à probabilidade de errar o bit 1. O BSC tem a representação apresentada na figura 5, em que X e Y são v.a. binárias que representam a entrada e a saída do canal, respectivamente. A probabilidade de transmitir um bit e detectar o outro designa-se por α . A matriz estocástica

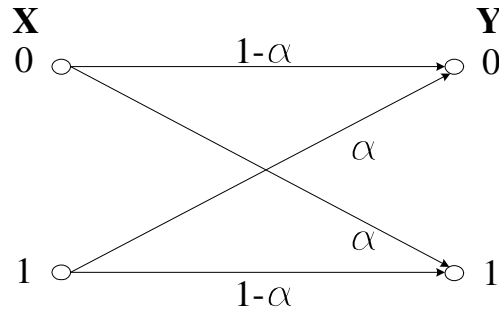


Figura 5: Representação do canal discreto binário: modelo BSC (*binary symmetric channel*).

$$\mathbf{P} = \begin{bmatrix} 1 - \alpha & \alpha \\ \alpha & 1 - \alpha \end{bmatrix}, \quad (1)$$

define as probabilidades de transição $p(y|x)$ associadas ao canal. Na diagonal principal de \mathbf{P} constam as probabilidades de não existir erro na transmissão. As probabilidades de transição $p(y|x)$ são apresentadas na tabela 1.

$\mathbf{p}(y x)$	Y=0	Y=1
X=0	$1-\alpha$	α
X=1	α	$1-\alpha$

Tabela 1: Probabilidades de transição $p(y|x)$.

A presença do ruído no canal físico é reflectida no valor atribuído à probabilidade α . Existe erro de transmissão sobre o BSC, quando se transmite determinado bit e se detecta o outro. Desta forma, a probabilidade de erro (P_e) no BSC é dada por:

$$\begin{aligned} P_e &= p(Y = 0, X = 1) + p(Y = 1, X = 0) \\ &= p(Y = 0|X = 1)p(X = 1) + p(Y = 1|X = 0)p(X = 0) \\ &= \alpha p(X = 1) + \alpha p(X = 0) \\ &= \alpha, \end{aligned} \quad (2)$$

verificando-se que esta é independente da distribuição de probabilidades de X. A P_e define o BER (*Bit Error Rate*) do canal. O BER é normalmente utilizado para classificar a qualidade de serviços¹ tais como a transmissão de dados e voz, por exemplo.

A figura 6 apresenta dois modelos de canal, alternativos ao BSC, designados por BEC (*Binary Erasure Channel*) e Z (*Z-channel*). No BEC, a saída é ternária indicando que existem situações de erro para as quais não é possível detectar (decidir) o bit transmitido. No *Z-channel* apenas o bit 1 pode ser recebido em erro. O modelo BSC é mais realista do que estes dois modelos, sendo mais utilizado.

¹QOS-*Quality of Service*.

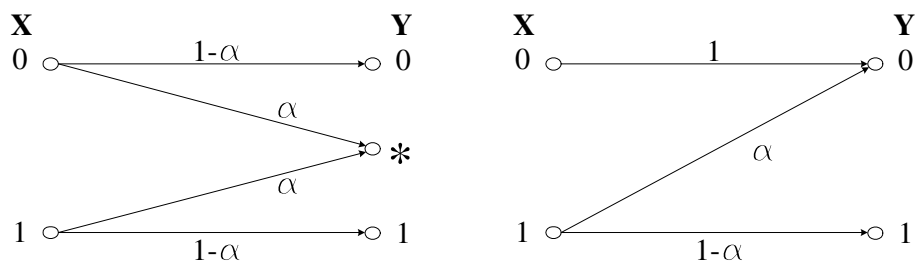


Figura 6: Modelos de canal: BEC (*Binary Erasure channel*) e Z (*Z-channel*).

3.3 Teorema da codificação de canal e capacidade de canal

A probabilidade de erro no canal determina a capacidade C de transferência de informação no canal. O segundo teorema de Shannon²[2, 7, 10], também designado de teorema da codificação de canal, afirma que:

Dada a capacidade C do canal, existe uma técnica de codificação tal que a informação pode ser transmitida no canal a um ritmo $R=C$, com probabilidade de erro arbitrariamente pequena. Se $R>C$, não é possível transmitir sem erros.

O teorema da codificação de canal estabelece que a capacidade de canal é o majorante para o ritmo de transferência de informação, com probabilidade de erro arbitrariamente pequena. Estabelece ainda que esta probabilidade de erro é atingível para qualquer ritmo abaixo da capacidade. No entanto, não define como se devem estabelecer os códigos. O trabalho de Hamming³ consiste precisamente numa forma sistemática de estabelecer códigos [4]. Para uma prova do teorema deve consultar [2, 7, 10].

A figura 7 apresenta o modelo genérico de canal discreto binário, no qual a entrada e a saída são representadas por v.a. discretas binárias.

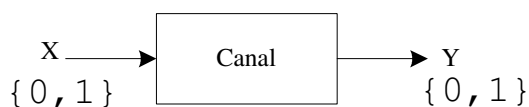


Figura 7: Modelo genérico de canal discreto binário.

3.3.1 Analogia com o teorema da codificação de fonte

Existe uma analogia entre o teorema de codificação de fonte [2, 7, 10] e o teorema da codificação de canal, na medida em que ambos estabelecem limites. O teorema de codificação de fonte estabelece que o comprimento médio L das palavras do código que representa, sem distorção, os símbolos de determinada fonte com entropia $H(X)$ deve verificar

$$L \geq H(X), \quad (3)$$

²Claude E. Shannon (1916-2001) <http://turnbull.mcs.st-and.ac.uk/~history/Mathematicians/Shannon.html>

³Richard W. Hamming (1915-1998) <http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Hamming.html>

ou seja, a entropia é o limite inferior admissível para o comprimento médio das palavras do código, sem distorção. No teorema da codificação de canal tem-se

$$R \leq C, \tag{4}$$

ou seja, a capacidade de canal é o limite superior do ritmo de transmissão de informação das palavras do código (de canal), para se obter probabilidade de erro arbitrariamente pequena.

3.3.2 Capacidade de transferência de informação

A capacidade de transferência de informação num canal é definida, através do segundo teorema de Shannon, como

$$C = \max_{p(x)} \{I(X; Y)\} \text{ [bit/symbol]}. \tag{5}$$

A equação (5) interpreta-se como a maximização da quantidade de informação transmitida no canal, para a distribuição de probabilidades existente na entrada do canal. Procura-se que a informação mútua, entre a v.a. de entrada e a v.a. de saída, seja máxima. Tendo em conta que a informação mútua é dada por

$$I(X; Y) = H(X) - H(X|Y), \tag{6}$$

verifica-se que o termo $H(X|Y)$ representa o equívoco introduzido pelo ruído existente no canal. Num canal sem ruído tem-se $H(X|Y) = 0$, ou seja, não há incerteza sobre X dado que se observa Y . Nesta situação, tem-se que $I(X; Y) = H(X) = H(Y)$. A figura 8 estabelece a relação entre $H(X)$, $H(Y)$, $H(X|Y)$ e $I(X; Y)$, para:

- canal sem ruído ($I(X; Y) = H(X) = H(Y)$);
- canal com ruído, no caso geral ($I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$);
- canal com ruído, no caso extremo ($I(X; Y) = 0$).

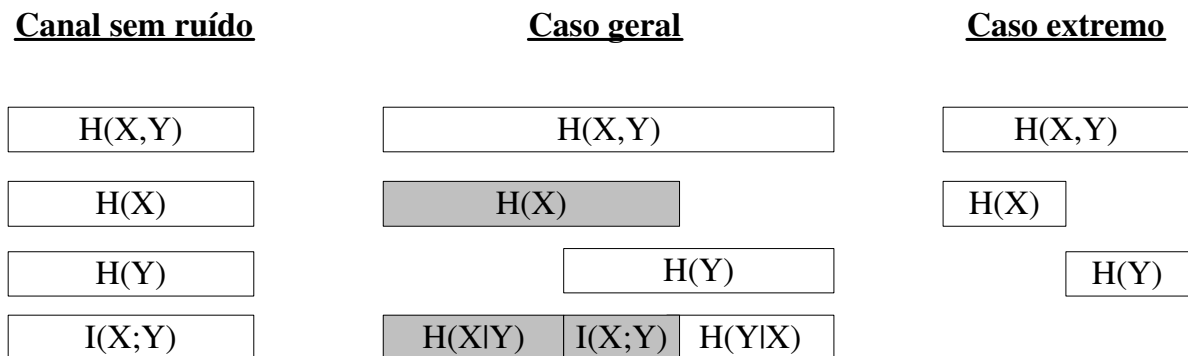


Figura 8: Relação entre entropia, entropia conjunta, entropia condicionada e informação mútua.

3.3.3 Capacidade do BSC

Nesta secção apresenta-se o cálculo da capacidade de transferência de informação sobre o BSC. Na equação (2), apresenta-se a P_e do BSC. A probabilidade de erro determina a capacidade de transferência de informação no canal. Tendo em conta as definições de capacidade de canal e de informação mútua, nas equações (5) e (6) tem-se

$$C = \max_{p(x)} \{H(X) - H(X|Y)\}. \tag{7}$$

Tendo em conta que o valor máximo de $H(X)$ é 1, dado que X representa os dígitos binários do código produzido por um codificador de fonte, tem-se

$$C = \max_{p(x)} \{1 - H(X|Y)\}, \quad (8)$$

sendo então necessário calcular o equívoco $H(X|Y)$ introduzido pelo ruído. Tendo em conta a definição de entropia condicionada

$$H(X|Y) = - \sum_x \sum_y p(x, y) \log_2 p(x|y), \quad (9)$$

verifica-se que é necessário determinar $p(x, y)$ e $p(x|y)$. Através da lei de Bayes, obtém-se

$$p(x, y) = p(y|x)p(x). \quad (10)$$

Dado que $H(X) = 1$, tem-se $p(X = 0) = p(X = 1) = \frac{1}{2}$. A tabela 2 apresenta $p(x, y)$, assim calculada. Para obter $p(x|y)$, aplica-se de novo a lei de Bayes, apresentada na equação (11),

$\mathbf{p(x,y)}$	Y=0	Y=1
X=0	$\frac{1-\alpha}{2}$	$\frac{\alpha}{2}$
X=1	$\frac{\alpha}{2}$	$\frac{1-\alpha}{2}$

Tabela 2: Probabilidade conjunta $p(x, y)$.

$$p(x|y) = \frac{p(x, y)}{p(y)}, \quad (11)$$

obtendo-se assim a tabela 3.

$\mathbf{p(x y)}$	Y=0	Y=1
X=0	$1-\alpha$	α
X=1	α	$1-\alpha$

Tabela 3: Probabilidades condicionadas $p(x|y)$.

Finalmente, aplicando a definição de entropia condicionada, apresentada na equação (9), tem-se que a capacidade do BSC é

$$\begin{aligned} C &= 1 - \left(-2 \frac{1-\alpha}{2} \log_2(1-\alpha) - 2 \frac{\alpha}{2} \log_2(\alpha) \right) \\ &= 1 - \left(-(1-\alpha) \log_2(1-\alpha) - \alpha \log_2(\alpha) \right) \\ &= 1 - \Omega(\alpha), \end{aligned} \quad (12)$$

em que $\Omega(\alpha)$ é a entropia da fonte binária cuja probabilidade de um dos símbolos é α (ver figura 19). A figura 9 mostra a capacidade do BSC, em função da sua P_e . Note-se que na situação em que $P_e = \alpha = \frac{1}{2}$ não há transferência de informação, embora exista comunicação. Por outro lado, com $\alpha = 0$ ou $\alpha = 1$, o canal é determinista e a quantidade de informação transmitida vale 1.

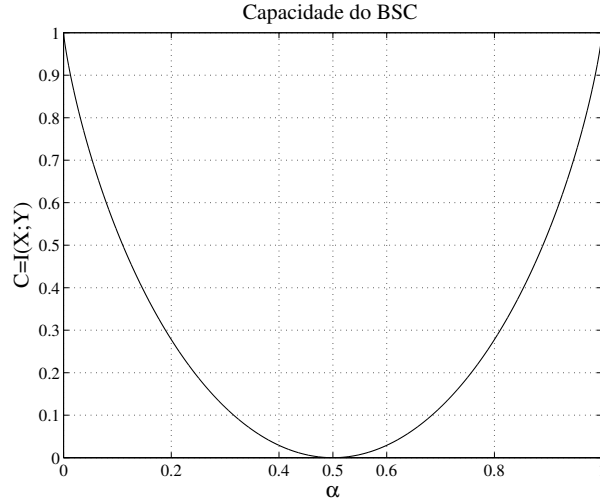


Figura 9: Capacidade de transferência de informação do BSC em função da probabilidade de erro.

3.3.4 Cálculo alternativo da capacidade do BSC

Outra forma de cálculo da capacidade do canal consiste na utilização da definição de informação mútua

$$I(X; Y) = \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}. \quad (13)$$

Tendo $p(x, y)$ na tabela 2 e sabendo que $p(X = 0) = p(X = 1) = \frac{1}{2}$, resta calcular $p(Y = 0)$ e $p(Y = 1)$. Tem-se que

$$\begin{aligned} p(Y = 0) &= p(Y = 0, X = 0) + p(Y = 0, X = 1) \\ &= p(Y = 0|X = 0)p(X = 0) + p(Y = 0|X = 1)p(X = 1) \\ &= (1 - \alpha)p(X = 0) + \alpha p(X = 1) \\ &= (1 - \alpha)\frac{1}{2} + \alpha\frac{1}{2} \\ &= \frac{1}{2}, \end{aligned} \quad (14)$$

e $p(Y = 1) = 1 - p(Y = 0) = \frac{1}{2}$. Desta forma obtém-se

$$\begin{aligned} I(X; Y) &= 2\frac{1 - \alpha}{2} \log_2 \left(\frac{\frac{1 - \alpha}{2}}{\frac{1}{2}\frac{1}{2}} \right) + 2\frac{\alpha}{2} \log_2 \left(\frac{\frac{\alpha}{2}}{\frac{1}{2}\frac{1}{2}} \right) \\ &= (1 - \alpha) \log_2(2(1 - \alpha)) + \alpha \log_2(2\alpha) \\ &= (1 - \alpha)(\log_2 2 + \log_2(1 - \alpha)) + \alpha(\log_2 2 + \log_2 \alpha) \\ &= (1 - \alpha)(1 + \log_2(1 - \alpha)) + \alpha(1 + \log_2 \alpha) \\ &= 1 + \log_2(1 - \alpha) - \alpha - \alpha \log_2(1 - \alpha) + \alpha + \alpha \log_2(\alpha) \\ &= 1 + (1 - \alpha) \log_2(1 - \alpha) + \alpha \log_2(\alpha) \\ &= 1 - \Omega(\alpha) \end{aligned} \quad (15)$$

4 Códigos de codificação de canal

Nesta secção apresentam-se os códigos utilizados na codificação de canal, bem como as suas propriedades e características. Consideram-se os códigos de bloco, nos quais a mensagem e a palavra de código consistem em vectores (blocos) de bits.

4.1 Caracterização do codificador e decodificador

Seja $\mathbf{m} = [m_0 \ m_1 \ \dots \ m_{k-1}]$ o vector mensagem com k bits e $\mathbf{c} = [c_0 \ c_1 \ \dots \ c_{n-1}]$ a palavra de código com n ($> k$) bits. Existem 2^k mensagens e palavras de código diferentes. Cada mensagem corresponde a uma palavra de código.

4.1.1 Codificador

O codificador de canal consiste numa função que realiza o mapeamento entre a mensagem e a respectiva palavra de código:

$$\text{Cod} : \{0, 1\}^k \rightarrow \{0, 1\}^n. \quad (16)$$

O codificador calcula

$$q = n - k, \quad (17)$$

bits redundantes em função dos bits da mensagem. Os bits redundantes (também designados por bits de paridade) são concatenados aos de mensagem. A forma como são associados aos bits da mensagem, classifica o código como sistemático ou não sistemático. Na forma sistemática, os bits redundantes são concatenados no início ou no final dos bits de mensagem. Na forma não sistemática, os bits redundantes são entrelaçados com os bits de mensagem. Designando por $\{b_0, b_1, \dots, b_{q-1}\}$ os q bits redundantes, apresentam-se alguns exemplos destas formas:

- sistemática

$$\begin{aligned} \mathbf{c} &= [m_0 \ m_1 \ \dots \ m_{k-1} \ \dots \ b_0 \ b_1 \ \dots \ b_{q-1}], \\ \mathbf{c} &= [b_0 \ b_1 \ \dots \ b_{q-1} \ \dots \ m_0 \ m_1 \ \dots \ m_{k-1}]; \end{aligned}$$

- não sistemática

$$\begin{aligned} \mathbf{c} &= [m_0 \ b_0 \ b_1 \ m_1 \ \dots \ m_{k-1} \ \dots \ b_2 \ \dots \ b_{q-1}], \\ \mathbf{c} &= [b_0 \ b_1 \ m_0 \ b_2 \ m_1 \ \dots \ m_{k-1} \ \dots \ b_3 \ \dots \ b_{q-1}]. \end{aligned}$$

A introdução de bits redundantes conduz ao afastamento entre as 2^k palavras de código, tal como se ilustra na figura 10, com $k = 2$ e $n = 3$.

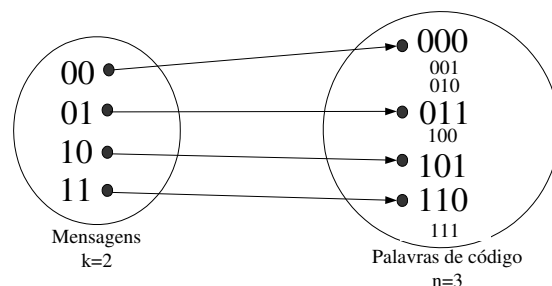


Figura 10: Mapeamento entre mensagens e palavras de código.

4.1.2 Descodificador

O decodificador realiza a sequência de acções:

1. recebe a palavra \mathbf{y} (possivelmente com erros);
2. estima a palavra de código \mathbf{c} que lhe deu origem;
3. estima a mensagem \mathbf{m} correspondente.

A figura 11 ilustra as acções do codificador, decodificador e do canal. A mensagem \mathbf{m} é transformada na palavra de código \mathbf{c} ; esta é enviada através do canal, obtendo-se $\mathbf{y} = \mathbf{c} + \mathbf{e}$ na saída deste, em que \mathbf{e} representa o padrão de erro adicionado à palavra de código. Caso \mathbf{e} seja diferente do vector nulo, a palavra recebida tem erros. O decodificador procura estimar a palavra de código transmitida e a respectiva mensagem $\hat{\mathbf{m}}$.



Figura 11: Sequência de acções realizada pelo codificador e pelo decodificador de canal.

Caso a palavra recebida \mathbf{y} não pertença ao código, então ocorreu um ou mais erros na transmissão. Para lidar com a existência de erros, o decodificador funciona num dos modos:

- detecção;
- correcção;
- detecção e correcção.

No modo de detecção, quando o decodificador detecta que existe erro na palavra recebida, reporta esse erro e envia um pedido de retransmissão. No modo de correcção, sempre que detectada a presença de um (ou mais) erro(s) procura-se corrigir esse(s) erro(s). O primeiro modo designa-se por ARQ (*Automatic Repeat ReQuest*), enquanto que o segundo toma o nome de FEC (*Forward Error Correction*). É ainda possível funcionar em modo misto (detecção e correcção) procurando corrigir os erros para os quais existe redundância suficiente para o fazer, e pedir a retransmissão da palavra de código sempre que forem detectados erros para os quais não há capacidade de correcção.

As características dos códigos, nomeadamente as capacidades de detecção e correcção de erros determinam os modos de funcionamento do decodificador.

4.2 Características dos códigos de bloco

Nesta secção analisam-se as características dos códigos de bloco, considerando que existem k bits de mensagem e n bits na palavra de código. Estes códigos são habitualmente referidos como (n,k) . As características a considerar são as seguintes:

- **Code rate (ritmo):** $R = \frac{k}{n} = \frac{k}{k+q}$; é a medida de eficiência do código, porque representa o quociente do número de bits de informação sobre o número total de bits transmitidos;
- **Distância de Hamming (dH)** entre duas palavras do código define-se como o número de bits que varia entre duas palavras.

- **Distância mínima** (d_{\min}): é a menor distância de Hamming entre duas quaisquer palavras do código; depende do número de bits redundantes, tal que $d_{\min} \leq q + 1$, com $q = n - k$.
- **Capacidade de detecção**: detecta todos os padrões até l erros, com $l \leq d_{\min} - 1$.
- **Capacidade de correção**: corrige todos os padrões até t erros, com $t \leq \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- **Capacidade de detecção e correção**: detecta até l erros e corrige até t erros com $d_{\min} \geq l + t + 1$ e $l > t$.

As capacidades de detecção e correção são obtidas à custa da introdução de redundância e dependem da distância mínima do código. Aumentar a distância mínima melhora as capacidades de detecção e correção, mas em contrapartida diminui a eficiência do código. Os critérios de desenho dos códigos de codificação de canal são:

- dado o R maximizar d_{\min} ;
- dada a d_{\min} minimizar R .

O desenho de códigos eficientes constitui um problema complexo. Em seguida, são analisados os códigos de repetição e de bit de paridade par, para os quais a abordagem de desenho é relativamente simples. Serão ainda abordados os códigos lineares enquanto sub-classe de códigos, caracterizados por possuírem construção fácil, baixa complexidade de codificação e de decodificação e capacidades razoáveis de detecção e correção.

Os códigos de bloco (n,k) também podem ser representados na forma (n,M,d) , sendo n o comprimento das palavras, $M = 2^k$ o número de palavras de código e d a distância mínima do código [14].

4.3 Código de repetição (3,1)

A repetição é uma forma simples de introduzir redundância na mensagem. Considerando mensagens com $k=1$ bit e introduzindo dois bits de redundantes iguais que constituem repetição da mensagem, tem-se o código (3,1) apresentado na tabela 4.

Mensagem	Palavra de Código
0	000
1	111

Tabela 4: Mensagens e palavras de código para o código de repetição (3,1).

A figura 12 ilustra o mapeamento entre o espaço das mensagens e das palavras de código. Note-se o afastamento entre as palavras do código.

A decodificação deste código é realizada por maioria, ou seja, se a maioria dos bits da palavra de código recebida valer 1, então decodifica-se a mensagem 1, caso contrário, decodifica-se a mensagem 0. Este código tem $d_{\min}=3$ e como tal:

- detecta todos os erros de 1 e 2 bit;
- corrige todos os erros de 1 bit.

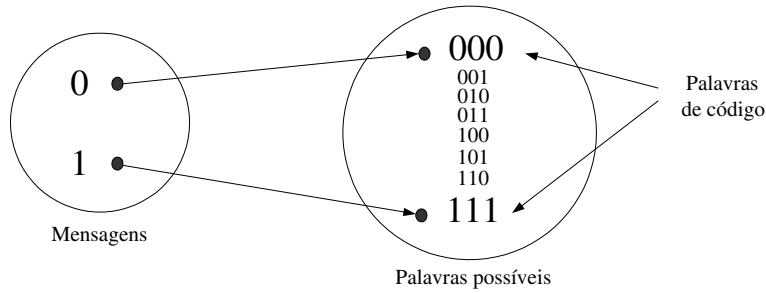


Figura 12: Código de repetição (3,1): mapeamento entre mensagens e palavras de código.

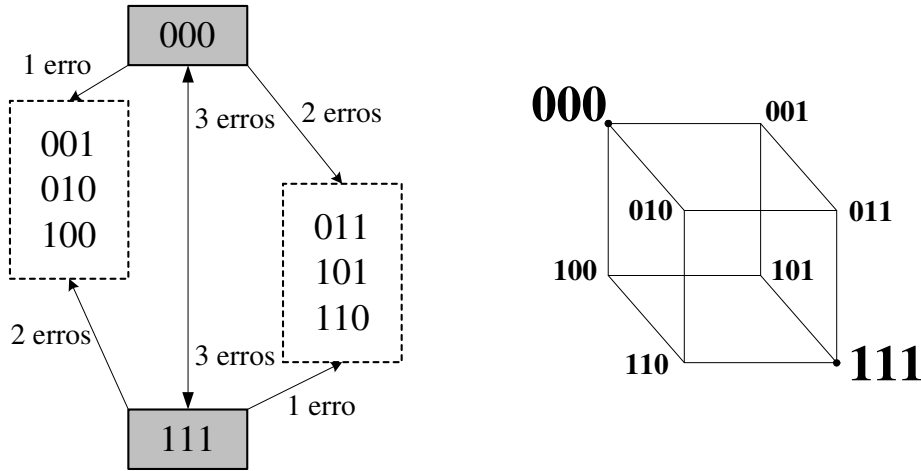


Figura 13: Ilustração das capacidades de detecção e correção de erros do código de repetição (3,1).

A figura 13 ilustra estas capacidades de detecção e correção de erros, evidenciando o afastamento entre as palavras do código. Este código pode funcionar nos modos de FEC e ARQ.

Considerando que se utiliza este código sobre um BSC com $P_e = \alpha = 10^{-5}$ tem-se que a probabilidade de errar 1 bit, sobre uma palavra de 3 bits, é dada por

$$P(1, 3) = C_1^3 \alpha^1 (1 - \alpha)^2 = \frac{3!}{2!1!} \alpha (1 - \alpha)^2 = 3\alpha - 6\alpha^2 + 3\alpha^3 \approx 3 \times 10^{-5}, \quad (18)$$

em que C_1^3 representa combinações de três um a um. A probabilidade de errar 2 bits é

$$P(2, 3) = C_2^3 \alpha^2 (1 - \alpha)^1 = \frac{3!}{1!2!} \alpha^2 (1 - \alpha) = 3\alpha^2 - 3\alpha^3 \approx 3 \times 10^{-10}, \quad (19)$$

Finalmente, a situação extrema de errar os 3 bits da palavra ocorre com probabilidade

$$P(3, 3) = C_3^3 \alpha^3 (1 - \alpha)^0 = \frac{3!}{0!3!} \alpha^3 = \alpha^3 = 10^{-15}, \quad (20)$$

concluindo-se que $P(3, 3) \ll P(2, 3) \ll P(1, 3)$. Verifica-se que a capacidade de correção até 1 bit errado é adequada nesta situação.

4.4 Código bit de paridade par (3,2)

O código bit de paridade par (3,2), consiste em adicionar um bit de paridade no final da mensagem. Este bit é a soma módulo 2 dos bits da mensagem obtendo-se assim a palavra

Mensagem	Palavra de Código
00	000
01	011
10	101
11	110

Tabela 5: Mensagens e palavras de código para o código bit de paridade par (3,2).

de código $\mathbf{c} = [m_0 \ m_1 \ m_0 \oplus m_1]$. A tabela 5 apresenta as palavras de código. Este código tem $d_{\min}=2$ e detecta a presença de 1 e 3 bits errados⁴, não tem capacidade de correcção e como tal não pode ser utilizado no modo FEC. A descodificação é realizada recalculando a paridade da mensagem recebida, comparando-a com a paridade transmitida; se forem iguais não se detectam erros, caso contrário são detectados 1 ou 3 erros, na palavra recebida. A figura 14 ilustra a disposição relativa das palavras de código, evidenciando que $d_{\min} = 2$, uma vez que entre quaisquer duas palavras do código, é sempre necessário percorrer duas arestas do cubo. Por outro lado, verifica-se que a existência de 3 erros sobre qualquer palavra de código é sempre detectável, uma vez que resulta numa palavra que não pertence ao código.

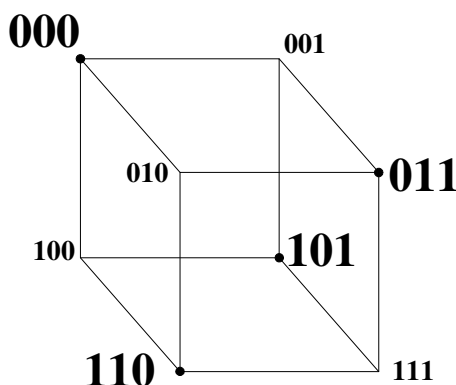


Figura 14: Palavras do código de bit de paridade par (3,2).

4.5 Códigos lineares

Tal como constatado anteriormente, o desenho de códigos eficientes é um problema complexo. Procura-se maximizar a distância mínima do código (d_{\min}), tendo como restrição o rate R . Em alternativa, podemos maximizar R com restrição d_{\min} . São também problemas adicionais a memória ocupada e complexidade do codificador e do decodificador.

Recorrendo aos conceitos de estrutura algébrica e espaço vectorial [3, 6, 11, 14] definem-se os códigos lineares. As palavras de código são elementos de determinado sub-espaço vectorial. Obtêm-se assim os **códigos lineares de bloco** [1, 5, 13, 14]. Designam-se de **bloco** porque todas as palavras têm a mesma dimensão e **lineares** porque:

- o vector nulo é palavra do código $\mathbf{c} = [0 \ 0 \ 0 \ \dots \ 0]$;

⁴O valor $d_{\min}=2$ garante a detecção de todos os erros de 1 bit, para qualquer código. Neste código, é ainda possível detectar 3 bits errados.

- a soma modular de duas palavras do código é ainda uma palavra do código.

Os códigos lineares são um sub-conjunto de todos os códigos, apresentando as vantagens de requererem pouca memória e os codificadores e decodificadores são constituídos por operações simples.

4.5.1 Características

O peso de Hamming (\mathbf{w}) de uma palavra define-se como o número de bits não nulos nessa palavra. Sejam c_i e c_j duas palavras distintas de um código linear de bloco. A d_{\min} do código é dada por

$$d_{\min} = \min_{i \neq j} dH(c_i, c_j). \quad (21)$$

Dado que o código é linear, tem-se que a soma modular de duas palavras dada por

$$c_k = c_i \oplus c_j, \quad (22)$$

é ainda outra palavra do código, diferente do vector nulo. Desta forma, a d_{\min} do código é dada por

$$d_{\min} = \min \mathbf{w}(c_k), \quad (23)$$

sendo c_k qualquer palavra de código diferente do vector nulo. O peso de Hamming desta palavra corresponde ao número de bits em que c_i e c_j diferem.

Os códigos de repetição e de bit de paridade par também são lineares. As tabelas 6 e 7 apresentam as palavras destes códigos e os respectivos pesos de Hamming, através dos quais se confirma que:

- para o código de repetição (3,1) tem-se $d_{\min} = 3$, $l = 2$ e $t = 1$.
- para o código de bit de paridade par (3,2) tem-se $d_{\min} = 2$, $l = 1$ e $t = 0$.

Mensagem	Palavra de Código	Peso de Hamming
0	000	0
1	111	3

Tabela 6: Mensagens, palavras de código e respectivo peso de Hamming para o código de repetição (3,1).

Mensagem	Palavra de Código	Peso de Hamming
00	000	0
01	011	2
10	101	2
11	110	2

Tabela 7: Mensagens, palavras de código e respectivo peso de Hamming para o código bit de paridade par (3,2).

4.5.2 Códigos de Hamming

Os códigos de Hamming⁵[1, 2, 4, 5] constituem uma família de códigos lineares de bloco, desenhada com o critério de possuírem $d_{\min} = 3$, corrigindo todos os erros de 1 bit. A motivação deste critério de desenho está no facto de que sobre um BSC, a probabilidade de errar 2 bits na mesma palavra de código é muito menor do que a probabilidade de errar apenas 1 bit: $P(2, n) \ll P(1, n)$. Os códigos de Hamming são definidos por um parâmetro inteiro $r (\geq 2)$ tal que: $(n, k) = (2^r - 1, 2^r - 1 - r)$. Com $r = 3$ tem-se $k=4$ e $n=7$; estabelecendo as equações de paridade

$$\begin{aligned} b_0 &= m_1 \oplus m_2 \oplus m_3, \\ b_1 &= m_0 \oplus m_1 \oplus m_3, \\ b_2 &= m_0 \oplus m_2 \oplus m_3, \end{aligned} \quad (24)$$

obtêm-se as palavras de código na forma sistemática

$$\mathbf{c} = [m_0 \ m_1 \ m_2 \ m_3 \ b_0 \ b_1 \ b_2]. \quad (25)$$

A figura 15 ilustra os bits de paridade utilizados no código Hamming (7,4). Verifica-se que cada

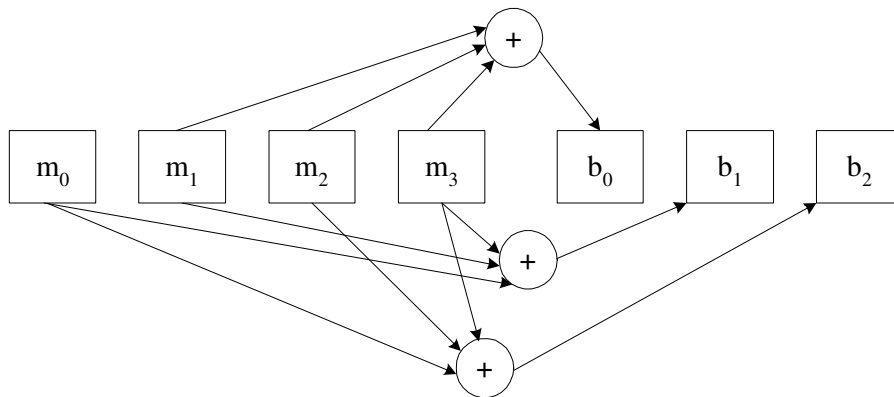


Figura 15: Ilustração dos bits de paridade no código Hamming (7,4).

palavra de código pode então ser escrita na forma matricial

$$\mathbf{c} = m\mathbf{G} = m [I_4 \mid \mathbf{P}] = \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (26)$$

A matriz \mathbf{P} de dimensões $k \times q$ designa-se por sub-matriz geradora de paridade; cada coluna de \mathbf{P} estabelece uma equação de paridade, ou seja, cada um dos bits $\{b_0, b_1, b_2\}$ da equação (24). A matriz \mathbf{G} de dimensões $k \times n$ designa-se de matriz geradora do código, uma vez que estabelece todas as palavras do código. Cada linha de \mathbf{G} é uma palavra do código. Todas as palavras do código são obtidas por combinação linear das linhas de \mathbf{G} , os coeficientes da combinação linear são os bits da mensagem. A tabela 8 apresenta as $2^4 = 16$ mensagens, as palavras do código e o respectivo peso de Hamming. Verifica-se que, à excepção do vector nulo, o menor peso de Hamming de todas as palavras vale 3, logo o código tem $d_{\min} = 3$. Note-se que este é um código de Hamming. Estabelecendo outra sub-matriz geradora de paridade \mathbf{P} , com pelo menos dois bits a 1 por cada linha, obtêm-se outro código de Hamming.

⁵Richard W. Hamming (1915-1998) <http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Hamming.html>

Mensagem	Palavra de Código	Peso de Hamming
0000	0000000	0
0001	0001111	4
0010	0010101	3
0011	0011010	3
0100	0100110	3
0101	0101001	3
0110	0110011	4
0111	0111100	4
1000	1000011	3
1001	1001100	3
1010	1010110	4
1011	1011001	4
1100	1100101	4
1101	1101010	4
1110	1110000	3
1111	1111111	7

Tabela 8: Mensagens, palavras de código e peso de Hamming para o código Hamming (7,4).

4.5.3 Sub-espço vectorial

Na perspectiva vectorial, tem-se que \mathbf{G} é um conjunto de vectores linearmente independentes, ou seja, nenhuma linha de \mathbf{G} pode ser obtida por combinação linear das outras linhas. A matriz \mathbf{G} gera $2^4 = 16$ vectores de um total possível de $2^7 = 128$ e como tal é a base de sub-espço vectorial. As palavras do código são elementos desse sub-espço vectorial. Note-se a economia de memória na codificação: para obter as 16 palavras de código, basta armazenar 4 palavras. Considerando o código de Hamming (15,11), com $2^{11} = 2048$ palavras, verifica-se que a matriz geradora é constituída apenas por $k=11$ palavras.

As matrizes geradores para os códigos de repetição (3,1) e de bit de paridade par (3,2), são

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad (27)$$

respectivamente.

4.6 Descodificação

Os códigos de bloco linear caracterizam-se por ter codificação e descodificação com baixa complexidade, ocupando pouca memória. A codificação consiste em multiplicar o vector mensagem (de dimensões $1 \times k$) pela matriz geradora (de dimensões $k \times n$) do código obtendo

$$\mathbf{c} = \mathbf{m}\mathbf{G}. \quad (28)$$

A descodificação assume uma forma semelhante. A palavra de código recebida é também multiplicada por uma matriz, de dimensões $n \times q$, que designaremos por \mathbf{H}^T . Desta multiplicação obtém-se o vector

$$\mathbf{s} = \mathbf{c}\mathbf{H}^T, \quad (29)$$

de dimensões $1 \times q$, designado por síndrome. A matriz \mathbf{H}^T é ortogonal à matriz geradora tal que

$$\mathbf{GH}^T = \underbrace{\begin{bmatrix} 0 & \dots & 0 \\ \vdots & \dots & \vdots \\ 0 & \dots & 0 \end{bmatrix}}_{\text{Matriz nula com } k \times q \text{ bits}}. \quad (30)$$

A matriz \mathbf{H} designa-se por matriz de teste de paridade. Dado que a palavra de código é escrita na forma $\mathbf{c} = \mathbf{mG}$, tem-se que

$$\mathbf{cH}^T = \mathbf{mGH}^T = \underbrace{[0 \ 0 \ \dots \ 0]}_{\text{Vector nulo com } q \text{ bits}} \quad (31)$$

para qualquer palavra do código \mathbf{c} . Caso o decodificador receba a palavra $\mathbf{y} = \mathbf{c} + \mathbf{e}$, a qual não pertence ao código, dado que \mathbf{e} é o padrão de erro adicionado à palavra de código \mathbf{c} , o valor de \mathbf{s} é não nulo. Neste caso tem-se

$$\mathbf{s} = \mathbf{yH}^T = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = (\mathbf{mG} + \mathbf{e})\mathbf{H}^T = \underbrace{\mathbf{mGH}^T}_{[0 \ 0 \ \dots \ 0]} + \mathbf{eH}^T = \mathbf{eH}^T, \quad (32)$$

verificando-se que o síndrome apenas depende do padrão de erro \mathbf{e} . A matriz \mathbf{H}^T é construída de acordo com a forma da matriz geradora \mathbf{G} . A funcionalidade de \mathbf{H}^T consiste em recalculer os bits de paridade sobre os bits de mensagem e comparar esses bits de paridade com aqueles transmitidos na palavra. Quando a matriz geradora \mathbf{G} assume a forma $\mathbf{G} = [\mathbf{I}_k \ \mathbf{P}]$, tem-se que $\mathbf{H}^T = \begin{bmatrix} \mathbf{P} \\ \mathbf{I}_q \end{bmatrix}$. Para o código de Hamming (7,4) apresentado na equação (24), tem-se que

$$\mathbf{H}^T = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (33)$$

No caso dos códigos de repetição (3,1) e bit de paridade par (3,2), estas matrizes são

$$\mathbf{H}^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{H}^T = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (34)$$

respectivamente.

4.6.1 Tabela de síndromas

Cada linha de \mathbf{H}^T representa um síndrome do código. Associado a cada síndrome, está um padrão de 1 bit em erro. Sempre que esse padrão de erro se verificar sobre a palavra, obtém-se o respectivo síndrome, tal como se constata na equação (32). A tabela 9 mostra os síndromas e os respectivos padrões de erro, para o código Hamming (7,4). Ao síndrome nulo, corresponde a ausência de erros detectados. O número de síndromas é dado por $2^q - 1 = 2^3 - 1 = 7$, tantos quantos os padrões de erro de 1 bit. Devido a este facto, os códigos de Hamming são designados de códigos perfeitos [14].

Síndrome (s)	Padrão de erro (e)
000	0000000
011	1000000
110	0100000
101	0010000
111	0001000
100	0000100
010	0000010
001	0000001

Tabela 9: Tabela de síndromas e respectivos padrões de erro, para o código Hamming (7,4).

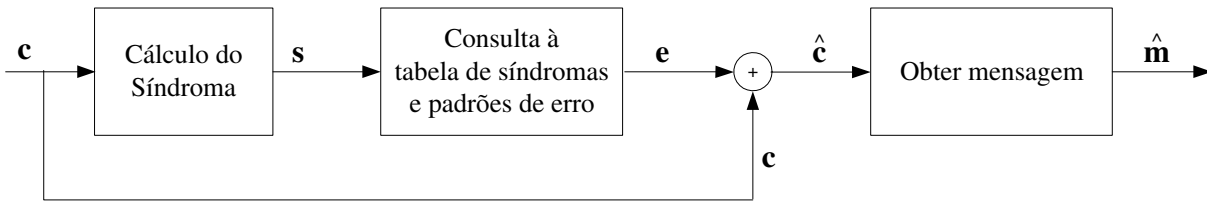


Figura 16: Ilustração da descodificação baseada em síndrome.

A figura 16 ilustra o processo de descodificação e consulta à tabela de síndromas para correcção. Em função do síndrome obtido, extrai-se o respectivo padrão de erro e soma-se este à palavra de código recebida. A partir desta estima-se a mensagem. Para funcionamento no modo de detecção, basta verificar se o síndrome é ou não nulo. Caso o síndrome seja nulo, não são detectados erros. Caso seja não nulo, detecta-se a presença de erros.

Note-se que, à semelhança do codificador, também o descodificador necessita de pouca memória: basta possuir a matriz \mathbf{H}^T os padrões de erro (ou, em alternativa, a informação de qual o bit errado para cada síndrome).

4.6.2 Exemplos de descodificação

Exemplifica-se a descodificação e a correcção de erros, utilizando as palavras $\mathbf{c} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ e $\mathbf{c} = [0\ 0\ 0\ 1\ 1\ 1\ 1]$ do código de Hamming (7,4), cuja matriz \mathbf{H}^T é apresentada na equação (33). A tabela 10 apresenta exemplos de descodificação, sobre estas palavras, na presença de 0, 1, 2 e 3 bits errados. Verifica-se que para as situações em que existe 1 bit errado, a correcção é realizada. Nas situações em que existem 2 bits errados, acima da capacidade de correcção do código, o mecanismo de correcção baseado em síndrome introduz mais um erro e a mensagem estimada está errada. Caso existam 3 erros, também não é possível descodificar a mensagem correcta. Na última linha da tabela verifica-se que após o erro, obteve-se outra palavra do código. Nesta situação o síndrome calculado é nulo. Note-se que se o descodificador funcionar em modo de detecção de erros, são detectados todos os erros de 1 bit e 2 bit por palavra de código. É ainda possível detectar alguns erros de 3 bit, sempre que o síndrome obtido é não nulo.

4.6.3 Cálculo da distância mínima

No processo de descodificação as palavras recebidas são multiplicadas por \mathbf{H}^T . As palavras de código são ortogonais a \mathbf{H}^T e a multiplicação das palavras de código por \mathbf{H}^T , consiste na

Situação	m	c	e	y=c+e	s	ĉ	ĉm
0 erros	0001	0001111	0000000	0001111	000	0001111	0001
0 erros	0000	0000000	0000000	0000000	000	0000000	0000
1 erro	0001	0001111	1000000	1001111	011	0001111	0001
1 erro	0000	0000000	0000001	0000001	001	0000000	0000
2 erros	0001	0001111	0001010	0000101	101	0010101	0010
2 erros	0000	0000000	1100000	1100000	101	1110000	1110
3 erros	0001	0001111	1010100	1011011	010	1011001	1011
3 erros	0000	0000000	1110000	1110000	000	1110000	1110

Tabela 10: Exemplos de descodificação em modo correcção, na presença de erros, para o código Hamming (7,4).

combinação linear das linhas desta matriz; estas linhas são as que correspondem aos bits com o valor 1 na palavra de código. Conclui-se que a d_{\min} do código pode ser obtida a partir das linhas de \mathbf{H}^T , como o número mínimo de linhas que é necessário somar para obter o vector nulo [5].

4.7 Código de Hamming (7,4) não sistemático

O código de Hamming (7,4), inicialmente proposto [2, 4, 14], tem a forma não sistemática. A matriz geradora é

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (35)$$

As palavras do código, organizadas na forma $\mathbf{c} = [b_0 \ b_1 \ m_0 \ b_2 \ m_1 \ m_2 \ m_3]$, constam da tabela 11

A matriz de teste de paridade e a sua transposta são

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{H}^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad (36)$$

respectivamente. A tabela 12 apresenta os síndromas e os respectivos padrões de 1 bit em erro. Pela análise da tabela, constata-se que o valor numérico do síndrome, entendido como número em binário natural a 3 bits, representa a posição do bit errado.

4.8 Códigos lineares modificados

Por vezes existem restrições que levam a modificações do comprimento das palavras de código a enviar por um determinado canal. Uma situação exemplificativa é a comunicação série; quando se pretende transmitir dados via canal série utilizando um UART, geralmente os dados devem ser enviados em blocos de 8 bits. Se pretendermos enviar palavras de um código Hamming (7,4) é vantajoso utilizar o “oitavo bit” como bit de paridade.

Mensagem	Palavra de Código	Peso de Hamming
0000	0000000	0
0001	1101001	4
0010	0101010	3
0011	1000011	3
0100	1001100	3
0101	0100101	3
0110	1100110	4
0111	0001111	4
1000	1110000	3
1001	0011001	3
1010	1011010	4
1011	0110011	4
1100	0111100	4
1101	1010101	4
1110	0010110	3
1111	1111111	7

Tabela 11: Mensagens, palavras de código e peso de Hamming para o código Hamming (7,4) não sistemático.

Valor	Síndrome (s)	Padrão de erro (e)
0	000	0000000
1	001	1000000
2	010	0100000
3	011	0010000
4	100	0001000
5	101	0000100
6	110	0000010
7	111	0000001

Tabela 12: Tabela de síndromas e respectivos padrões de erro, para o código Hamming (7,4) não sistemático.

4.8.1 Extensão

A extensão de um código linear de bloco (n,k) consiste em acrescentar mais um bit de redundância, passando a ter a representação $(n+1,k)$. Para os mesmos bits de mensagem existe mais um bit de controlo de paridade, o que aumenta a capacidade de controlo de erros, mas diminui o code rate (eficiência) do código. Em termos de espaço vectorial, o número de vectores que constituem a matriz geradora mantém-se. O comprimento das palavras aumenta de um. Se for efectuada a extensão de um código de Hamming (7,4) obtém-se um código (8,4). Um exemplo de extensão é acrescentar um bit de redundância que efectua a paridade de todos os bits anteriores que constituem a palavra. Seguindo este critério a matriz geradora do código de

Hamming estendido é

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \quad (37)$$

A tabela 13 apresenta as $2^4 = 16$ mensagens, as palavras do código e o respectivo peso de Hamming, para este código. Verifica-se que o código tem $R = \frac{4}{8} = 0.5$, $d_{\min}=4$, tem $l=3$ e $t=1$.

Mensagem	Palavra de Código	Peso de Hamming
0000	00000000	0
0001	00011110	4
0010	00101011	4
0011	00110101	4
0100	01001101	4
0101	01010011	4
0110	01100110	4
0111	01111000	4
1000	10000111	4
1001	10011001	4
1010	10101100	4
1011	10110010	4
1100	11001010	4
1101	11010100	4
1110	11100001	4
1111	11111111	8

Tabela 13: Mensagens, palavras de código e peso de Hamming para o código Hamming (8,4).

4.8.2 Redução

A redução de um código (n,k) consiste em retirar um bit da mensagem, mantendo o número de bits redundantes, o que resulta num código $(n-1,k-1)$. Para o código de Hamming (7,4), retirando o último bit de mensagem, obtém-se o código (6,3) cuja matriz geradora é

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \quad (38)$$

As palavras de código constam da tabela 14. O código tem $R = \frac{3}{6} = 0.5$, $d_{\min}=3$, $l=2$ e $t=1$. Visto que se diminui o número de bits de mensagem mantendo o número de bits de redundância, a capacidade de controlo de erros do código reduzido será sempre igual ou superior à do código inicial.

4.8.3 Código dual

Seja Cod um código (n,k) e Cod_2 outro código de dimensões $(n,n-k)$. Os códigos Cod e Cod_2 dizem-se duais, caso possuam as seguintes características [14]:

Mensagem	Palavra de Código	Peso de Hamming
000	000000	0
001	001101	3
010	010110	3
011	011011	4
100	100011	3
101	101110	4
110	110101	4
111	111000	3

Tabela 14: Mensagens, palavras de código e peso de Hamming para o código (6,3).

- a matriz geradora de Cod é a matriz de teste de paridade de Cod_2 ;
- a matriz de teste de paridade de Cod é a matriz geradora de Cod_2 .

As palavras de ambos os códigos são ortogonais entre si. Analisando as palavras de código enquanto elementos de sub-espaço vectorial, verifica-se que para qualquer código, as matrizes \mathbf{G} e \mathbf{H} geram espaços vectoriais ortogonais e ambos contêm o vector nulo.

A equação (30) mostra que todos os produtos internos entre as linhas da matriz \mathbf{G} e as colunas de matriz \mathbf{H}^T são nulos. Como as colunas de \mathbf{H}^T são as linhas de \mathbf{H} , então as linhas de \mathbf{G} são ortogonais a todas as linhas de \mathbf{H} . Considerando um código de Hamming (7,4) com matrizes geradora e de teste de paridade definidas por

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad (39)$$

respectivamente, verifica-se que \mathbf{H} gera um código (7,3), uma vez que tem dimensões 3×7 , e é constituída por vectores linearmente independentes.

Em termos de espaços vectoriais podemos verificar que temos um espaço vectorial V de dimensão 7. Este espaço vectorial é decomposto em dois subespaços vectoriais ortogonais. As palavras do código (7,4) pertencem ao sub-espaço de dimensão 4, enquanto que as palavras do código (7,3) pertencem ao sub-espaço de dimensão 3. A dimensão de um espaço define-se como o número máximo de vectores linearmente independentes existentes nesse espaço [14]. A figura 17 ilustra esta decomposição.

Outras formas de modificação de códigos lineares estão em [14].

4.9 Análise comparativa de códigos

A tabela 15 apresenta uma análise comparativa dos códigos apresentados, relativamente ao code rate e às capacidades de detecção e correcção de erros. Para os códigos de repetição, verifica-se que com o aumento do número de bits redundantes, as capacidades de detecção e correcção de erros aumentam e a eficiência do código diminui. Os códigos de bit de paridade têm as mesmas capacidades de detecção e correcção, diferindo na eficiência. O mesmo se passa para os códigos de Hamming: dado que possuem sempre $d_{\min} = 3$, tem-se que para estes códigos apenas muda a eficiência.

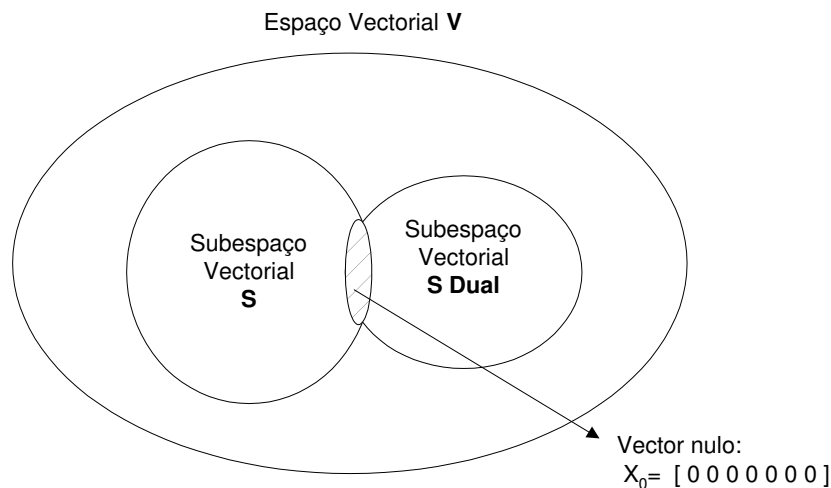


Figura 17: Decomposição dum espaço vectorial em dois sub-espaços.

Código	Rate	d_{\min}	Detecção (l)	Correcção (t)
Repetição (2,1)	0.5	2	1	0
Repetição (3,1)	0.333	3	2	1
Repetição (4,1)	0.25	4	3	1
Repetição (5,1)	0.2	5	4	2
Paridade (3,2)	0.666	2	1	0
Paridade (8,7)	0.875	2	1	0
Hamming (7,4) r=3	0.571	3	2	1
Hamming (15,11) r=4	0.733	3	2	1
Hamming (31,26) r=5	0.838	3	2	1

Tabela 15: Análise comparativa de códigos.

4.10 Aplicações

Nesta secção elencam-se aplicações dos códigos detectores e correctores de erros, referidos neste texto.

- **Comunicação série assíncrona** - 1 bit de paridade por cada byte;
- **Memórias RAM** - utilizam 1 bit de paridade por cada byte, ou mais do que 1 bit de paridade por cada byte, no caso da ECC (*Error Correcting Code*) RAM;
- **Teletexto** - Hamming (8,4), extensão do Hamming (7,4);
- **Discos rígidos** - Código de Hamming, com bits de paridade por sector;
- **RAID (*Redundant Array of Independent Disks*)**, o qual é constituído por vários níveis:
 - . RAID 1 - *mirroring*; código de repetição
 - . RAID 2 - *Hamming system*; no caso do Hamming (7,4) usa 7 discos rígidos (4 dados + 3 paridade)

. RAID 3 - *parallel transfer with parity drive*; usa código bit de paridade, no qual existem vários discos de dados e um de paridade.

- **Bluetooth (comunicação sem fios)** - usa código de repetição (3,1) para o *packet header* e Hamming modificado (15,10) para a *application data*.

5 Utilização do MATLAB - Communications Toolbox

Apresentam-se exemplos de utilização do MATLAB, nomeadamente algumas funcionalidades da Communications Toolbox, para codificação, decodificação, detecção e correcção de erros. Verifica-se experimentalmente o teorema da codificação de canal.

5.1 Codificação e decodificação

O troço de código seguinte ilustra o estabelecimento da matriz geradora e detectora de paridade para um código de Hamming (7,4), utilizando a função `hammgen`. O parâmetro de entrada é o factor r de desenho do código, tal que $(n, k) = (2^r - 1, 2^r - 1 - r)$. A função `hammgen` retorna também os valores de n e k .

```
>> r=3; [H,G,n,k] = hammgen(r)
H =  1     0     0     1     0     1     1
     0     1     0     1     1     1     0
     0     0     1     0     1     1     1

G =  1     1     0     1     0     0     0
     0     1     1     0     1     0     0
     1     1     1     0     0     1     0
     1     0     1     0     0     0     1

n =  7

k =  4
```

Note-se que a matriz geradora está na forma sistemática $\mathbf{G} = [P \mid I_4]$. A sub-matriz geradora de paridade \mathbf{P} difere da apresentada na equação (26).

Exemplifica-se agora a obtenção de uma palavra de código, de duas formas distintas: realizando a multiplicação do vector mensagem pela matriz geradora; somando as duas linhas correspondentes da matriz geradora.

```
>> msg = [0 0 1 1];
>> c = mod(msg*G,2)
c =  0     1     0     0     0     1     1

>> c = mod(G(3,:) + G(4,:),2)
c =  0     1     0     0     0     1     1
```

Sobre a palavra de código, introduz-se um erro, trocando o segundo bit desta, através da soma com um padrão de erro. Calcula-se o síndrome e verifica-se que este é a segunda linha de \mathbf{H}^T , correspondendo ao padrão de erro somado à palavra.

```

>> e = [0 1 0 0 0 0 0]; y = mod(c + e,2)
y = 0      0      0      0      0      1      1

>> s=mod(y*H',2)
s = 0      1      0

>> H'
ans =
     1     0     0
     0     1     0
     0     0     1
     1     1     0
     0     1     1
     1     1     1
     1     0     1

```

5.2 Verificação do teorema da codificação de canal

Para verificar experimentalmente a relação entre R e C , estabelecida pelo teorema da codificação de canal, realiza-se a seguinte experiência:

- estabelece-se um código de Hamming (15,11) com $R = \frac{11}{15} = 0.733$;
- simula-se o BSC com P_e configurável;
- dado que a P_e determina a capacidade do canal, os valores de P_e escolhidos fazem com que C seja inferior ou superior a R ;
- transmitem-se L ($\gg 1$) palavras do código de Hamming (15,11), através do BSC, o que resulta na passagem de $15L$ bits;
- contabiliza-se o total de palavras com $\{0, 1, 2, 3, \dots, 15\}$ bits em erro, para as L transmissões.

A tabela 16 mostra os valores de C , para os valores de P_e em teste. A figura 18 apresenta

P_e	0.001	0.01	0.05	0.08
C	.989	.919	.714	.598

Tabela 16: Capacidade do BSC, para determinados valores da P_e .

os histogramas do número de palavras com $\{0, 1, 2, 3, \dots, 15\}$ bits em erro, para as $L=100000$ transmissões, através do BSC, para as probabilidades de erro consideradas na tabela 16. Dado que o código de Hamming tem $d_{\min}=3$, $l=2$ e $t=1$, verifica-se que apenas com a existência de 3 ou mais erros na palavra, é que não é possível transmitir sem erros (em modo de detecção). Verifica-se que, nos últimos dois gráficos, com $C=.714$ e $C=.598$, ($< R=0.733$), existem 3 e mais erros nas palavras recebidas, não sendo possível transmitir com P_e arbitrariamente pequena, nestas condições. Nos dois primeiros gráficos, com $C=.989$ e $C=.919$, ($> R=0.733$), o número de palavras com número de erros igual ou superior a 3 é desprezável, obtendo-se probabilidade de erro muito baixa. A tabela 17 apresenta os valores dos histogramas da figura 18, considerando apenas até 8 bits em erro, dado que o maior número de erros obtido numa palavra foi 7.

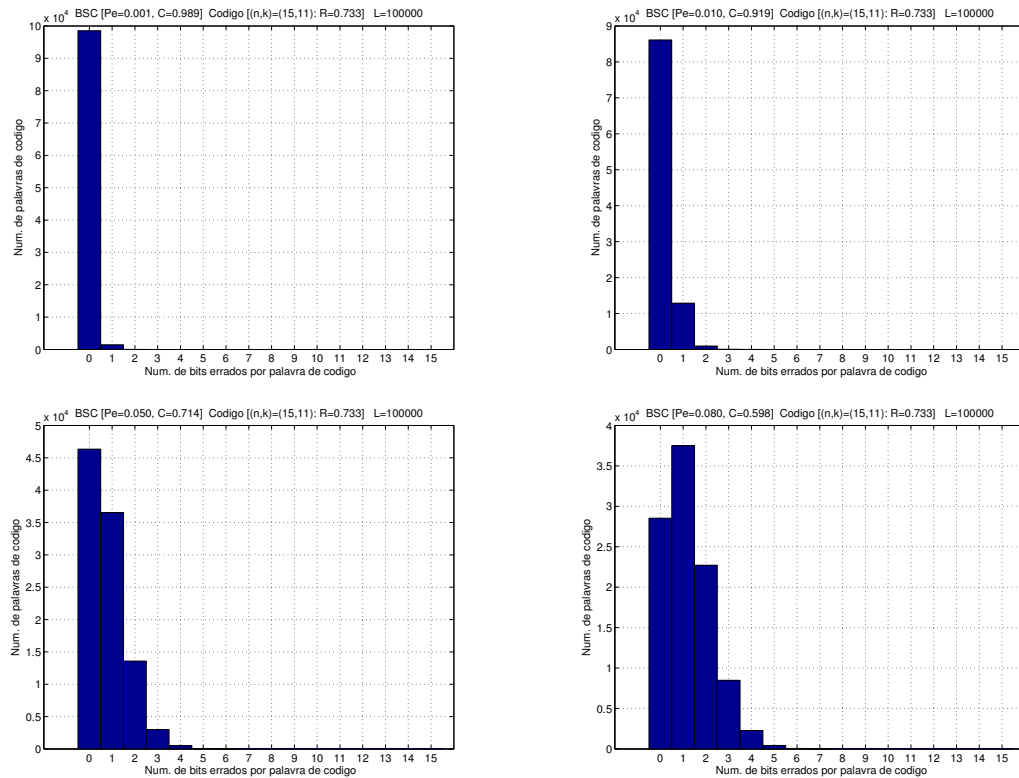


Figura 18: Histogramas do número de palavras com $\{0, 1, 2, 3, \dots, 15\}$ bits em erro após 100000 transmissões através de BSC com $P_e \in \{0.001, 0.01, 0.05, 0.08\}$

C	0 erros	1 erro	2 erros	3 erros	4 erros	5 erros	6 erros	7 erros	8 erros
.989	98533	1454	13	0	0	0	0	0	0
.919	86109	12886	957	45	3	0	0	0	0
.714	46332	36534	13577	3004	499	52	2	0	0
.598	28526	37495	22714	8491	2275	429	65	5	0

Tabela 17: Número de palavras do código Hamming (15,11), recebidas com $\{0, 1, 2, 3, \dots, 8\}$ bits em erro após 100000 transmissões através de BSC, tal como ilustrado na figura 18.

6 Apêndice: Teoria da Informação

Nesta apêndice apresentam-se as expressões de algumas medidas da teoria da informação, para complementar a exposição, realizada ao longo do texto.

Entropia

$$H(X) = - \sum_x p(x) \log_2 p(x) \quad (40)$$

Entropia da fonte binária

Para uma fonte binária em que um dos símbolos tem probabilidade α tem-se que a entropia é dada por

$$\Omega(\alpha) = -(1 - \alpha) \log_2(1 - \alpha) - \alpha \log_2(\alpha), \quad (41)$$

representada na figura 19.

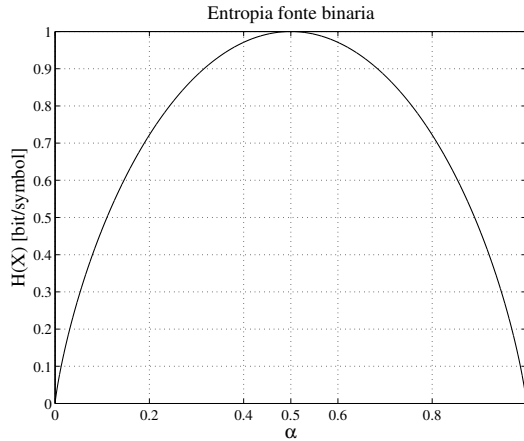


Figura 19: Entropia da fonte binária.

Entropia condicionada

$$H(X|Y) = - \sum_x \sum_y p(x, y) \log_2 p(x|y) \quad (42)$$

Entropia conjunta

$$H(X, Y) = - \sum_x \sum_y p(x, y) \log_2 p(x, y) \quad (43)$$

Informação mútua

$$I(X; Y) = - \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \quad (44)$$

Referências

- [1] A. B. Carlson. *Communication Systems*. McGraw-Hill, 1986.
- [2] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- [3] D. Dummit and R. Foote. *Abstract Algebra*. Prentice Hall, 1991.
- [4] R. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 1950.
- [5] S. Haykin. *Communication Systems*. John Wiley & Sons, 1994.
- [6] L. Magalhães. *Algebra Linear*. Texto Editora, 1997.
- [7] D. McKay. *Information Theory, Inference and Learning Algorithms*. Cambridge Press, 2003.
- [8] D. Salomon. *Data Compression - The complete reference*. Springer-Verlag, New York, December 1997.
- [9] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann, 2nd edition, March 2000.
- [10] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July, October 1948.

- [11] G. Strang and K. Borre. *Linear Algebra, Geodesy, and GPS*. Wellesley-Cambridge Press, 1997.
- [12] Y. Viniotis. *Probability and Random Processes*. McGraw-Hill International Editions, 1997.
- [13] D. Welsh. *Codes and Cryptography*. Oxford Science Publications, 1988.
- [14] S. Wicker. *Error Control Systems*. Prentice Hall, 1995.