

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

Códigos lineares de bloco cíclicos

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

17 Junho 2004

Versão 1.0

Índice

1	Introdução	1
2	Códigos lineares de bloco	1
3	Códigos lineares de bloco cíclicos	2
3.1	Palavras de código como polinómios	2
3.2	Polinómio gerador	3
3.2.1	Relação com a matriz geradora	4
3.3	Códigos cíclicos sistemáticos	4
3.3.1	Exemplos de polinómios geradores e respectivos códigos	5
3.3.2	Factorização de $X^n + 1$	6
3.3.3	Polinómios geradores standard	7
3.3.4	Códigos de Hamming cíclicos e não cíclicos	7
3.4	Capacidades de detecção e correcção	8
3.5	Codificação e descodificação - realização em <i>hardware</i>	9
3.5.1	Codificação	10
3.5.2	Descodificação	10
3.6	Famílias de códigos cíclicos e aplicações	13
3.7	Utilização do MATLAB - Communications Toolbox	14

1 Introdução

Este documento apresenta os códigos de bloco linear cíclicos. Na secção 2 revêem-se características dos códigos lineares de bloco, o seu tratamento matricial para a codificação e para a decodificação. A secção 3 introduz os códigos lineares de bloco cíclicos, como sub-classe dos códigos lineares de bloco. Analisam-se as palavras de código como polinómios. Definem-se códigos através do respectivo polinómio gerador. Apresentam-se códigos cíclicos na forma sistemática e não sistemática. Exemplifica-se a realização de codificadores e decodificadores em *hardware*. Elencam-se aplicações e cálculos em MATLAB.

2 Códigos lineares de bloco

Os códigos utilizados na codificação de canal, designam-se de bloco [1, 4, 5, 6], quando a mensagem e a palavra de código consistem em vectores (blocos) de bits. 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. Estes códigos designam-se habitualmente por (n,k) , sendo k a dimensão do vector mensagem e n a dimensão da palavra de código.

Existem 2^k mensagens e palavras de código diferentes. Cada mensagem corresponde a uma palavra de código. O codificador de canal consiste numa função que realiza o mapeamento entre a mensagem e a respectiva palavra de código:

$$Cod : \{0, 1\}^k \rightarrow \{0, 1\}^n.$$

O codificador calcula

$$q = n - k,$$

bits redundantes em função dos bits da mensagem. O código é também **linear** [1, 4, 5, 6] se respeitar as condições:

- o vector nulo é palavra do código $\mathbf{c} = [0 \ 0 \ 0 \ \dots \ 0]$;
- a soma modular de duas palavras do código é ainda uma palavra do código.

Obtêm-se assim os **códigos lineares de bloco**. 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 [6].

A codificação é realizada através do produto do vector mensagem \mathbf{m} de dimensões $1 \times k$ pela matriz geradora \mathbf{G} de dimensões $k \times n$, obtendo-se a palavra de código

$$\mathbf{c} = \mathbf{mG},$$

de dimensões $1 \times n$. As linhas da matriz \mathbf{G} são palavras de código, linearmente independentes, ou seja, nenhuma linha de \mathbf{G} pode ser obtida por combinação linear das outras linhas. A matriz \mathbf{G} gera 2^k vectores de um total possível de 2^n , gerando o sub-espaco vectorial de dimensão k . As palavras do código são elementos desse sub-espaco vectorial [5, 6].

A decodificação é realizada de forma matricial. A palavra recebida, com dimensões $1 \times n$, é multiplicada pela matriz de teste de paridade transposta, de dimensões $n \times q$, obtendo-se o síndroma

$$\mathbf{s} = \mathbf{cH}^T,$$

o qual é um vector de dimensões $1 \times q$. Para efectuar detecção de erros, verifica-se se o síndroma obtido é nulo. Caso seja nulo, não existem erros detectados na palavra recebida; caso contrário

existem erros detectados. Para efectuar correcção, utiliza-se o padrão de erro associado a cada síndrome, somando esse padrão de erro à palavra recebida. Estima-se assim a palavra de código transmitida e a respectiva mensagem a partir dessa palavra [1, 4, 5, 6].

As matrizes \mathbf{H}^T e \mathbf{G} são ortogonais, o que implica que as linhas de \mathbf{G} (dimensões $k \times n$) e \mathbf{H} (dimensões $q \times n$) geram sub-espacos vectoriais ortogonais, de dimensão k e q , respectivamente. O espaco vectorial de dimensão n é assim decomposto em dois sub-espacos de dimensões k e q , ($n = q + k$) [6].

3 Códigos lineares de bloco cíclicos

Os códigos lineares de bloco cíclicos constituem uma sub-classe dos códigos lineares de bloco¹. Para além de possuírem as características enunciadas acima para os códigos lineares de bloco, verificam ainda a propriedade de que uma rotação cíclica de qualquer ordem sobre qualquer palavra do código produz outra palavra do código. Por exemplo, aplicando a rotação cíclica de ordem 1 para a esquerda sobre a palavra

$$\mathbf{c} = [c_{n-1} \ c_{n-2} \ c_{n-3} \ \dots \ c_0],$$

obtem-se

$$\mathbf{c}' = [c_{n-2} \ c_{n-3} \ \dots \ c_0 \ c_{n-1}].$$

Aplicando mais outra rotação para a esquerda, a palavra de código fica

$$\mathbf{c}'' = [c_{n-3} \ \dots \ c_0 \ c_{n-1} \ c_{n-2}].$$

3.1 Palavras de código como polinómios

O estudo dos códigos cíclicos é facilitado, caso as palavras de código sejam analisadas como polinómios. Seja a palavra de código

$$\mathbf{c} = [c_{n-1} \ c_{n-2} \ \dots \ c_1 \ c_0].$$

Esta palavra escrita na forma polinomial é dada por

$$c(X) = c_{n-1}X^{n-1} + c_{n-2}X^{n-2} + \dots + c_1X + c_0, \quad (1)$$

com $c_i \in \{0, 1\}$. Consideram-se as seguintes operações sobre polinómios: multiplicação por X e X^k , adição de polinómios e rotação. Sejam $c(X)$ e $d(X)$ dois polinómios correspondentes a palavras de código.

- **Multiplicação** - sobre o polinómio $c(X)$, definido na equação (1) a multiplicação por X resulta em

$$c(X)X = c_{n-1}X^n + c_{n-2}X^{n-1} + \dots + c_1X^2 + c_0X,$$

e a multiplicação por X^k vale

$$c(X)X^k = c_{n-1}X^{n-1+k} + c_{n-2}X^{n-2+k} + \dots + c_1X^{1+k} + c_0X^k.$$

¹Um código cíclico não é necessariamente linear. No entanto, caso não seja linear, a sua estrutura é menos interessante do ponto de vista prático [6].

- **Adição** - a adição de dois polinómios $c(X)$ e $d(X)$, do mesmo grau, em aritmética de módulo 2, resulta em

$$c(X) + d(X) = (c_{n-1} + d_{n-1})X^{n-1} + (c_{n-2} + d_{n-2})X^{n-2} + \dots + (c_1 + d_1)X + (c_0 + d_0).$$

Caso $c(X) = d(X)$, a soma dá o polinómio (vector) nulo.

- **Rotação** - a rotação cíclica para a esquerda do polinómio $c(X)$ é dada por

$$c'(X) = c_{n-2}X^{n-1} + \dots + c_1X^2 + c_0X + c_{n-1}.$$

A rotação é obtida, analiticamente pela operação

$$c'(X) = c(X)X + c_{n-1}(X^n + 1),$$

a qual consiste na multiplicação do polinómio por X , para deslocar uma posição para a esquerda

$$c(X)X = c_{n-1}X^n + c_{n-2}X^{n-1} + \dots + c_1X^2 + c_0X,$$

e em seguida, para anular o termo $c_{n-1}X^n$ e obter c_{n-1} , é necessário somar estes dois termos à expressão anterior, obtendo-se

$$c(X)X + c_{n-1}X^n + c_{n-1} = \underbrace{c_{n-1}X^n + c_{n-1}X^n}_0 + c_{n-2}X^{n-1} + \dots + c_1X^2 + c_0X + c_{n-1}.$$

Verifica-se finalmente que

$$\begin{aligned} c'(X) &= c(X)X + c_{n-1}(X^n + 1) \\ &= c_{n-2}X^{n-1} + \dots + c_1X^2 + c_0X + c_{n-1}. \end{aligned} \quad (2)$$

3.2 Polinómio gerador

As palavras de um código cíclico (n,k) são obtidas a partir do seu polinómio gerador $g(X)$, de coeficientes binários. Cada palavra de código $c_i(X)$ é escrita na forma

$$c_i(X) = f_i(X)g(X), \quad (3)$$

em que $f_i(X)$ é outro polinómio de grau $k - 1$, único para cada palavra do código. Contudo, não é forçoso que $f_i(X)$ seja o polinómio mensagem, ou seja, os bits de mensagem na forma polinomial. Para $g(X)$ ser polinómio gerador de um código de bloco cíclico (n,k) , com $q = n - k$ bits redundantes, deve ter as seguintes propriedades [4, 6]:

1. grau q ;
2. ser factor de $X^n + 1$ (ou de $X^n - 1$, em aritmética de módulo 2).

A primeira condição implica que o polinómio seja da forma $g(X) = X^q + \dots$. A segunda indica que o resto da divisão de $X^n + 1$ por $g(X)$ deve ser zero:

$$\text{rem} \left(\frac{X^n + 1}{g(X)} \right) = 0, \quad (4)$$

em que *rem* representa o resto da divisão dos dois polinómios. A equação (4) pode ainda ser expressa de forma equivalente²

$$\frac{X^n + 1}{g(X)} = h(X) + \frac{0}{g(X)} \quad (=) \quad X^n + 1 = g(X)h(X). \quad (5)$$

Esta condição implica que o termo de grau zero de $g(X)$ vale 1, concluindo-se que o polinómio gerador é da forma $g(X) = X^q + \dots + 1$. O polinómio $h(X)$, apresentado na equação (5), designa-se por polinómio de teste de paridade.

3.2.1 Relação com a matriz geradora

As palavras dos códigos de bloco lineares são obtidas a partir da combinação linear das linhas da matriz geradora \mathbf{G} . As palavras dos códigos de bloco lineares cíclicos, são estabelecidas a partir do polinómio gerador do código. Por outro lado, um código cíclico, dado que é linear, também possui matriz geradora. Esta pode ser obtida a partir do polinómio gerador, efectuando sucessivas rotações sobre este.

O polinómio $g(X)$, descreve o código estabelecendo-se a matriz geradora a partir deste polinómio. A partir do polinómio de teste de paridade $h(X)$, obtem-se a matriz de teste de paridade \mathbf{H} , tal que $\mathbf{GH}^T = \mathbf{0}$.

3.3 Códigos cíclicos sistemáticos

À semelhança dos códigos lineares de bloco, os códigos cíclicos também se apresentam na forma sistemática ou não sistemática. A condição de código sistemático em que as palavras de código estão organizadas em blocos de mensagem e de bits de paridade, assumindo por exemplo, a forma $\mathbf{c} = [m_0 \ m_1 \ \dots \ m_{k-1} \ b_0 \ b_1 \ \dots \ b_{q-1}]$, na forma polinomial é

$$c_i(X) = m_i(X)X^q + b(X), \quad (6)$$

em que $b(X)$ é o polinómio que representa os bits de paridade. A multiplicação por X^q desloca os bits de mensagem, para a esquerda em q posições. As equações (3) e (6), definem as condições de cíclico e cíclico e sistemático, respectivamente. Desta forma, para o código ser cíclico e sistemático deve verificar simultaneamente as duas condições

$$c_i(X) = f_i(X)g(X) = m_i(X)X^q + b(X), \quad (7)$$

para todas as palavras de código $c_i(X)$. A seguir, determina-se a expressão de $b(X)$ (bits de paridade); dividindo ambos os termos de (7) por $g(X)$ obtém-se

$$\begin{aligned} \frac{f_i(X)g(X)}{g(X)} &= \frac{m_i(X)X^q + b(X)}{g(X)} \\ (=) f_i(X) &= \frac{m_i(X)X^q}{g(X)} + \frac{b(X)}{g(X)} \end{aligned} \quad (8)$$

Em aritmética de módulo 2, a equação (8) assume a forma

$$\frac{m_i(X)X^q}{g(X)} = f_i(X) + \frac{b(X)}{g(X)}. \quad (9)$$

²Tendo em conta a divisão de polinómios: $\frac{\alpha(x)}{\beta(x)} = q(x) + \frac{r(x)}{\beta(x)}$.

Verifica-se então que, em aritmética módulo 2

$$b(X) = \text{rem} \left(\frac{m_i(X)X^q}{g(X)} \right), \quad (10)$$

sendo normalmente designado por CRC (*Cyclic Redundancy Check*) [4, 6]. A palavra de código linear de bloco cíclico e sistemático é então expressa na forma

$$c_i(X) = m_i(X)X^q + b(X) = m_i(X)X^q + \text{rem} \left(\frac{m_i(X)X^q}{g(X)} \right). \quad (11)$$

O CRC consiste no resto da divisão da mensagem deslocada de q bits para a esquerda pelo polinómio gerador. Tendo em conta a equação (11), podemos assim sumarizar a obtenção do CRC:

1. multiplicar o polinómio mensagem $m_i(X)$ por X^q , obtendo $m_i(X)X^q$;
2. dividir $m_i(X)X^q$ pelo polinómio gerador $g(X)$ e obter $b(X)$;
3. somar $b(X)$ a $m_i(X)X^q$ obtendo $c_i(X)$.

3.3.1 Exemplos de polinómios geradores e respectivos códigos

O polinómio $g(X) = X + 1$ gera um código cíclico (3,2). A figura 1 mostra a divisão $\frac{X^3+1}{X+1}$, a qual tem resto zero. A tabela 1 apresenta as palavras de código. Verifica-se que se trata

$$\begin{array}{r|l} 1 & 0 & 0 & 1 & & 1 & 1 \\ \hline 1 & 1 & & & & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & & & & & \\ & 1 & 1 & & & & & \\ \hline & 0 & 1 & 1 & & & & \\ & & 1 & 1 & & & & \\ \hline & & 0 & 0 & & & & \end{array}$$

Figura 1: Divisão de polinómios; polinómio gerador do código (3,2).

do código de bloco (3,2) de bit de paridade par. Para este código, podemos estabelecer, entre

Palavras do código (3,2)
000
011
110
101

Tabela 1: Palavras de código do código cíclico (3,2).

outras, as matrizes geradoras

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{G} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \quad (12)$$

A partir da divisão apresentada na figura 1, concluímos que o polinómio $X^2 + X + 1$ também é factor de $X^3 + 1$ e como tal gera um código cíclico (3,1), cujas palavras são [0 0 0] e [1 1 1]. Este é o código de repetição (3,1).

$$\frac{X^3 + 1}{X + 1} = X^2 + X + 1 \quad (=) \quad X^3 + 1 = (X + 1)(X^2 + X + 1).$$

O polinómio $g(X) = X^3 + X + 1$ gera um código cíclico. A figura 2 ilustra a divisão $\frac{X^7+1}{X^3+X+1}$. Verifica-se que a divisão apresenta resto nulo e como tal garante-se que $g(X)$ gera um código

$$\begin{array}{r} 10000001 \quad | \quad 1011 \\ \underline{1011} \quad 10111 \\ 001100 \\ \underline{1011} \\ 01110 \\ \underline{1011} \\ 01011 \\ \underline{1011} \\ 0000 \end{array}$$

Figura 2: Divisão de polinómios; polinómio gerador do código (7,4).

cíclico (7,4). A partir do polinómio gerador, podemos estabelecer a matriz geradora do código

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad (13)$$

na forma não sistemática, efectuando sucessivas rotações sobre este, obtendo palavras de código linearmente independentes. Para obter a matriz geradora na forma sistemática e dado que o código é linear efectuam-se adições de palavras de código, obtendo a forma desejada para a matriz, garantindo que as palavras são linearmente independentes. A matriz

$$\mathbf{G}_{\text{sist}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad (14)$$

define a forma sistemática do código. A primeira linha de \mathbf{G}_{sist} é obtida pela soma da 1ª, 3ª e 4ª linhas de \mathbf{G} , enquanto que a segunda linha consiste na soma da 2ª e 4ª linhas \mathbf{G} .

Pela análise da figura 2, verifica-se que

$$\frac{X^7 + 1}{X^3 + X + 1} = X^4 + X^2 + X + 1 \quad (=) \quad X^7 + 1 = (X^3 + X + 1)(X^4 + X^2 + X + 1),$$

ou seja, o polinómio $X^4 + X^2 + X + 1$ gera um código cíclico (7,3).

3.3.2 Factorização de $X^n + 1$

O polinómio gerador de um código (n,k) é factor de $X^n + 1$ e como tal a factorização de $X^n + 1$ em polinómios de coeficientes binários assume especial importância, para diferentes valores de

$X^n + 1$	Produto de polinómios
$X^{11} + 1$	$X^{11} + 1 = (X + 1)(X^{10} + X^9 + \dots + X + 1)$
$X^9 + 1$	$X^9 + 1 = (X + 1)(X^2 + X + 1)(X^6 + X^3 + 1)$
$X^7 + 1$	$X^7 + 1 = (X + 1)(X^3 + X^2 + 1)(X^3 + X + 1)$
$X^5 + 1$	$X^5 + 1 = (X + 1)(X^4 + X^3 + X^2 + X + 1)$
$X^3 + 1$	$X^3 + 1 = (X + 1)(X^2 + X + 1)$

Tabela 2: Factorização de $X^n + 1$ em polinómios de coeficientes binários.

n. A tabela 2 apresenta exemplos dessa factorização. Através da consulta da tabela obtêm-se polinómios geradores de diferentes graus. Por exemplo, os polinómios $X^3 + X^2 + 1$ e $X^3 + X + 1$ geram códigos com $q = 3$ bits redundantes. Dado que são factor de $X^7 + 1$ ambos geram códigos $(7, 7 - 3) = (7, 4)$. O polinómio $(X + 1)$ também gera um código cíclico com 1 bit redundante - código $(7,6)$, por exemplo. A factorização assegura que o polinómio gera um código cíclico; no entanto, não assegura que este seja um *bom* código cíclico, no que diz respeito à sua distância mínima.

3.3.3 Polinómios geradores standard

A tabela 3 apresenta polinómios geradores standard [6]. O número de bits redundantes do

Código CRC	Polinómio gerador
CRC4 code	$g(X) = X^4 + X^3 + X^2 + X + 1$
CRC7 code	$g(X) = X^7 + X^6 + X^4 + 1$
CRC12 code	$g(X) = X^{12} + X^{11} + X^3 + X^2 + X + 1$
CRC16 code	$g(X) = X^{16} + X^{15} + X^2 + 1$
CRC-CCITT ³ code	$g(X) = X^{16} + X^{12} + X^5 + 1$
CRC32 code	$g(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + \dots$ $\dots + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Tabela 3: Alguns polinómios geradores standard.

código corresponde ao grau do polinómio.

3.3.4 Códigos de Hamming cíclicos e não cíclicos

Os códigos de Hamming⁴ [1, 2, 3, 4] podem assumir a forma cíclica, escolhendo as equações de paridade de forma criteriosa. Os códigos de Hamming são definidos por um parâmetro inteiro r (≥ 2) tal que: $(n, k) = (2^r - 1, 2^r - 1 - r)$; com $r = 3$ tem-se $k=4$ e $n=7$. O código de Hamming $(7,4)$ com as equações de paridade

$$\begin{aligned}
 b_0 &= m_1 \oplus m_2 \oplus m_3, \\
 b_1 &= m_0 \oplus m_1 \oplus m_3, \\
 b_2 &= m_0 \oplus m_2 \oplus m_3,
 \end{aligned}
 \tag{15}$$

⁴Richard W. Hamming (1915-1998) <http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Hamming.html>

definido pela matriz geradora

$$\mathbf{G} = \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (16)$$

é não cíclico. Por exemplo, a rotação para a direita, da segunda linha de \mathbf{G} , não produz a terceira linha de \mathbf{G} . Os códigos de Hamming podem assumir forma cíclica. O polinómio

$$g(X) = X^3 + X + 1$$

de grau $q=3$ gera um código de Hamming (7,4) cíclico. Na figura 2 e na tabela 2 verifica-se que este polinómio é factor de $X^7 + 1$. A matriz geradora na forma sistemática, obtida a partir deste polinómio, apresentada na equação (14) gera um código de Hamming (7,4) cíclico. A tabela 4 apresenta todas as palavras de código.

Palavra de Código	Peso de Hamming
0000000	0
0001011	3
0010110	3
0011101	4
0100111	4
0101100	3
0110001	3
0111010	4
1000101	3
1001110	4
1010011	4
1011000	3
1100010	3
1101001	4
1110100	4
1111111	7

Tabela 4: Palavras de código e respectivo peso de Hamming para o código Hamming (7,4) cíclico sistemático, gerado por $g(X) = X^3 + X + 1$.

O polinómio $g(X) = X^3 + X^2 + 1$ também gera um código cíclico de Hamming (7,4) [6]. Procedendo de forma idêntica à anterior, obtêm-se todas as palavras de código, as quais se apresentam na tabela 5.

3.4 Capacidades de detecção e correcção

Dado que o código cíclico é um código linear de bloco, as capacidades de detecção e correcção de erros dos códigos, em função da distância mínima, mantêm-se relativamente aos códigos lineares de bloco. No entanto, dada a sua estrutura, os códigos cíclicos são adequados para a detecção de erros, nomeadamente de *burst* de erros, na situação em que $n \gg 1$. Um *burst* de comprimento B, numa palavra de n bits, é uma sequência contígua de B bits, na qual o primeiro e o último bit são recebidos em erro. Os códigos cíclicos (n,k) detectam [4]:

Palavra de Código	Peso de Hamming
0000000	0
0001101	3
0010111	4
0011010	3
0100011	3
0101110	4
0110100	3
0111001	4
1000110	3
1001011	4
1010001	3
1011100	4
1100101	4
1101000	3
1110010	4
1111111	7

Tabela 5: Palavras de código e respectivo peso de Hamming para o código Hamming (7,4) cíclico sistemático, gerado por $g(X) = X^3 + X^2 + 1$.

- todos os *burst* de comprimento igual ou inferior a $n - k$;
- uma fracção dos *burst* de comprimento $n - k + 1$; esta fracção é $1 - 2^{-(n-k-1)}$;
- uma fracção dos *burst* de comprimento maior que $n - k + 1$; esta fracção é $1 - 2^{-(n-k)}$;
- todas as combinações de $d_{\min} - 1$ ou menos erros;
- todos os padrões de erro com número ímpar de erros, se o polinómio gerador possuir um número par de coeficientes não nulos.

Por exemplo, considerando o polinómio **CRC7 code** $g(X) = X^7 + X^6 + X^4 + 1$, apresentado na tabela 3, verifica-se que este detecta:

- todos os *burst* de comprimento igual ou inferior a 7;
- $1 - 2^{-(7-1)} = 98.44$ % dos *burst* de comprimento 8;
- $1 - 2^{-7} = 99.22$ % dos *burst* de comprimento maior que 8;
- todos os padrões de erro com número ímpar de erros.

3.5 Codificação e descodificação - realização em *hardware*

Uma vantagem dos códigos cíclicos em relação aos códigos de bloco, é a realização do codificador e do descodificador por *hardware*, utilizando *flip-flops*, organizados na estrutura *shift-register*. Esta realização não necessita de cálculo matricial para obter o síndrome, sendo eficiente relativamente à memória ocupada. É ainda eficiente relativamente ao tempo de execução.

Bit mensagem m_i	b_2	b_1	b_0	Bit na saída
	0	0	0	
1	0	1	1	1
1	1	0	1	1
0	0	0	1	0
0	0	1	0	0
	1	0	0	0
	0	0	0	1
	0	0	0	0

Tabela 6: Cálculo de palavra de código cíclico Hamming (7,4) com polinómio gerador $g(X) = X^3 + X + 1$ e mensagem $m(X) = X^3 + X^2$; realização do codificador em *hardware*.

Bit mensagem m_i	b_2	b_1	b_0	Bit na saída
	0	0	0	
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	0	1	1	1
	1	1	0	0
	1	0	0	1
	0	0	0	1

Tabela 7: Cálculo de palavra de código cíclico Hamming (7,4) com polinómio gerador $g(X) = X^3 + X + 1$ e mensagem $m(X) = 1$; realização do codificador em *hardware*.

ou seja, obtém-se resto nulo na divisão da palavra de código pelo polinómio gerador. Caso a palavra recebida tenha erros, então pode ser escrita na forma

$$y_i(X) = c_i(X) + e(X),$$

em que $e(X)$ é o polinómio que representa o padrão de erro. Neste caso, a divisão da palavra de código pelo polinómio gerador é dada por

$$\frac{c_i(X) + e(X)}{g(X)} = \frac{c_i(X)}{g(X)} + \frac{e(X)}{g(X)}.$$

O resto desta divisão fica

$$s(X) = \text{rem} \left(\frac{c_i(X)}{g(X)} + \frac{e(X)}{g(X)} \right) = \text{rem} \left(\frac{e(X)}{g(X)} \right),$$

dado que a parcela $\frac{c_i(X)}{g(X)}$ tem resto nulo; o polinómio $s(X)$ designa-se de síndrome e depende apenas do padrão de erro $e(X)$. Caso este seja nulo, o síndrome também é nulo.

A descodificação baseia-se então na divisão da palavra de código pelo polinómio gerador, obtendo assim o síndrome correspondente. O descodificador pode então ser visto como um calculadora de síndromas. A figura 4 apresenta a calculadora de síndromas para o código cujo codificador se apresentou na figura 3 (com polinómio gerador $g(X) = X^3 + X + 1$). Este

descodificador é descrito pelas expressões *booleanas*

$$\begin{aligned} s_0 &= \text{input} \oplus s_2^*, \\ s_1 &= s_0^* \oplus s_2^*, \\ s_2 &= s_1^*, \end{aligned} \tag{18}$$

em que s_i^* representa o estado anterior do *flip-flop* designado por s_i . O funcionamento do

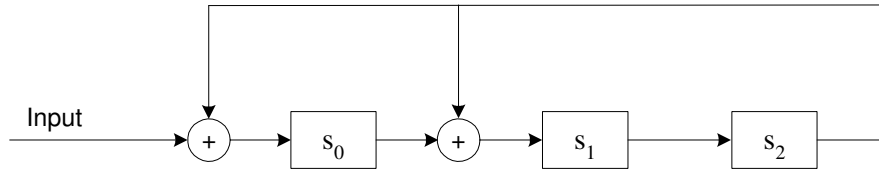


Figura 4: Cálculo de síndroma para o código cíclico Hamming (7,4) com polinómio gerador $g(X) = X^3 + X + 1$; realização do descodificador em *hardware*.

descodificador é o seguinte:

1. Ler os n bits da palavra de código (*shift-in*), pela mesma ordem com que foram enviados pelo codificador;
2. O síndroma corresponde ao conteúdo do *shift register*.

A tabela 8 apresenta a evolução do *shift-register* ao longo do processo de descodificação (cálculo do síndroma) da palavra de código $c(X) = X^6 + X^5 + X$, ou na forma de vector $\mathbf{c} = [1\ 1\ 0\ 0\ 0\ 1\ 0]$. A última linha da tabela contém o síndroma. Dado que a palavra pertence ao código, o síndroma obtido é nulo. A tabela 9 apresenta o cálculo do síndroma na situação em que a palavra de

Input	s_0	s_1	s_2
	0	0	0
1	1	0	0
1	1	1	0
0	0	1	1
0	1	1	1
0	1	0	1
1	0	0	0
0	0	0	0

Tabela 8: Cálculo do síndroma para palavra de código cíclico Hamming (7,4) com polinómio gerador $g(X) = X^3 + X + 1$; realização do descodificador em *hardware*.

código tem o último bit em erro $\mathbf{c} = [1\ 1\ 0\ 0\ 0\ 1\ 1]$. O síndroma obtido é $\mathbf{s} = [s_0\ s_1\ s_2] = [1\ 0\ 0]$. O cálculo do síndroma para a palavra $\mathbf{c} = [0\ 0\ 1\ 0\ 0\ 0\ 0]$ é apresentado na tabela 10. Esta palavra é obtida por troca do terceiro bit sobre o vector nulo⁵, portanto tem o terceiro bit errado. O síndroma obtido é $\mathbf{s} = [s_0\ s_1\ s_2] = [0\ 1\ 1]$.

⁵Palavra de código, dado que o código é linear.

Input	s_0	s_1	s_2
	0	0	0
1	1	0	0
1	1	1	0
0	0	1	1
0	1	1	1
0	1	0	1
1	0	0	0
1	1	0	0

Tabela 9: Cálculo do síndrome para palavra não pertencente ao código cíclico Hamming (7,4) com polinómio gerador $g(X) = X^3 + X + 1$; realização do decodificador em *hardware*.

Input	s_0	s_1	s_2
	0	0	0
0	0	0	0
0	0	0	0
1	1	0	0
0	0	1	0
0	0	0	1
0	1	1	0
0	0	1	1

Tabela 10: Cálculo do síndrome para palavra não pertencente ao código cíclico Hamming (7,4) com polinómio gerador $g(X) = X^3 + X + 1$; realização do decodificador em *hardware*.

A matriz geradora deste código está apresentada na equação (14). A matriz de teste de paridade transposta é

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

A tabela 11 apresenta os síndromas e os respectivos padrões de erro, para este código. Note-se que os síndromas estão apresentados na forma $\mathbf{s} = [s_2 \ s_1 \ s_0]$. Analisando a tabela, verifica-se que os síndromas obtidos nos cálculos apresentados nas tabelas 9 e 10, correspondem aos padrões de erro introduzidos sobre as mesmas.

3.6 Famílias de códigos cíclicos e aplicações

Para além dos polinómios geradores standard apresentados na tabela 3, existem outros que geram códigos cíclicos com capacidades de detecção de erros, com interesse prático. Por exemplo, o codificador de fonte WinRar⁶ realiza CRC32 com polinómio de 32 bit. O CRC a 32 bit é

⁶<http://home.zcu.cz/~squelch/WinRAR/TechNote.txt>

Síndromas = $[s_2 \ s_1 \ s_0]$	Padrão de erro (e)
000	0000000
101	1000000
111	0100000
110	0010000
011	0001000
100	0000100
010	0000010
001	0000001

Tabela 11: Tabela de síndromas e respectivos padrões de erro, para o código Hamming (7,4).

também utilizado pelo codificador de fonte `pkzip` e pelo protocolo `ZMODEM`⁷.

O polinómio designado por **CRC-CCITT** na tabela 3 é utilizado no protocolo X.25, enquanto que o polinómio **CRC32 code** é utilizado pelo protocolo *Ethernet*⁸.

A família de códigos BCH (Bose-Chaudhuri-Hocqunghem) e os códigos Reed-Solomon (RS) [4, 6] são dos mais utilizados. Os códigos de Reed-Solomon são utilizados pelo CD Audio, com a técnica de *interleave* e como tal designa-se por CIRC (*Cross Interleave Reed-Solomon Code*)⁹. O standard de transmissão digital de vídeo DVB (*Digital Video Broadcasting*) também utiliza códigos RS. Em <http://www.eccpage.com> e http://www.4i2i.com/reed_solomon_codes.htm encontram-se exemplos de outras aplicações.

3.7 Utilização do MATLAB - Communications Toolbox

Apresentam-se exemplos de utilização do MATLAB, nomeadamente algumas funcionalidades da Communications Toolbox. Estabelecem-se polinómios geradores para códigos (n,k) e obtêm-se as matrizes geradoras e de teste de paridade a partir do polinómio gerador.

O troço de código seguinte mostra o cálculo de polinómios geradores para códigos com dimensões (n,k) especificados como parâmetro na chamada à função `cyclpoly`. Note-se que algumas configurações de valores de n e k , não existem polinómios geradores. Para o caso do código (7,4) existem dois polinómios geradores.

```
>> n=10; k=4; pol = cyclpoly (n, k, 'all')
pol = 1      1      0      0      0      1      1

>> n=9; k=4; pol = cyclpoly (n, k, 'all')
No generator polynomial satisfies the given constraints.
pol = []

>> n=8; k=4; pol = cyclpoly (n, k, 'all')
No generator polynomial satisfies the given constraints.
pol = []
```

⁷<http://pauillac.inria.fr/~doligez/zmodem/zmodem.txt>

⁸<http://www.geocities.com/SiliconValley/Pines/8659/crc.htm>

⁹<http://www.cdrinfo.com/Sections/Articles/Specific.asp?ArticleHeadline=Writing+Quality&index=0>

```
>> n=7; k=4; pol = cyclpoly (n, k, 'all')
pol = 1      0      1      1
      1      1      0      1

>> n=6; k=4; pol = cyclpoly (n, k, 'all')
pol = 1      1      1

>> n=5; k=4; pol = cyclpoly (n, k, 'all')
pol = 1      1
```

A função `cyclgen` obtém a matriz geradora e de teste de paridade, especificando a dimensão das palavras código e o polinómio gerador. O vector `pol=[1 1 0 1]` corresponde ao polinómio gerador $g(X) = X^3 + X^2 + 1$.

```
>> n=7; pol=[1 1 0 1]; [H,G] = cyclgen(n, pol )
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
```

Utilizando agora o polinómio gerador $g(X) = X^3 + X + 1$ obtém-se outra matriz geradora e de teste de paridade.

```
>> n=7; pol=[1 0 1 1]; [H,G] = cyclgen(n, pol )
H = 1      0      0      1      1      1      0
     0      1      0      0      1      1      1
     0      0      1      1      1      0      1

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

Referências

- [1] A. Carlson. *Communication Systems*. McGraw-Hill, 1986.
- [2] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- [3] R. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 1950.
- [4] S. Haykin. *Communication Systems*. John Wiley & Sons, 1994.
- [5] D. Welsh. *Codes and Cryptography*. Oxford Science Publications, 1988.
- [6] S. Wicker. *Error Control Systems*. Prentice Hall, 1995.