

## O Código da Nave Espacial Mariner 9

### The Code of the Spacecraft Mariner 9

**José Silvino Dias**

Instituto Federal de Minas Gerais - IFMG, São João Evangelista, MG  
*jose.silvino@ifmg.edu.br*

**Mariana Garabini Cornelissen**

Universidade Federal de São João del Rei - UFSJ, Ouro Branco, MG  
*mariana@ufsj.edu.br*

**Resumo:** Este trabalho apresenta e descreve o código corretor de erros pertencente a uma família de códigos lineares, chamado Código de Reed-Muller de primeira ordem utilizado pela nave espacial Mariner 9 ao transmitir fotos do planeta Marte à Terra, quando foi enviada ao espaço em 1971 pela NASA (National Aeronautics and Space Administration).

**Palavras-chave:** código; mensagem; erro; Mariner 9.

**Abstract:** This paper presents and describes the error corrector code belonging to a family of linear codes, called the first-order Reed-Muller code used by the Mariner 9 spacecraft when transmitting photos of the planet Mars to Earth, when it was sent into space in 1971 by NASA (National Aeronautics and Space Administration).

**Key words:** code; message; error; Mariner 9.

## 1 Introdução

O presente trabalho descreve o código corretor de erros utilizado pela nave espacial Mariner 9, enviada ao espaço pela NASA em 30 de maio de 1971 com o objetivo de transmitir fotos do planeta Marte à Terra.

O Programa Mariner, segundo [1], teve o seu primeiro lançamento fracassado com a nave Mariner 1. A Mariner 2 foi a primeira missão que obteve sucesso, passou a 35 mil quilômetros do planeta Vênus em 14 de dezembro de 1962 e enviou informações da atmosfera de Vênus. A Mariner 3 lançada em 5 de novembro de 1964, tinha como objetivo alcançar o planeta Marte, porém pouco depois do lançamento surgiram problemas técnicos que inviabilizaram a missão. Após o fracasso da missão Mariner 3, foi enviada em 28 de novembro de 1964 a Mariner 4 que passou a 9920 quilômetros de Marte e transmitiu à Terra as primeiras fotografias da superfície marciana. A Mariner 5 sobrevoou Marte em 19 de outubro de 1967, coletou e transmitiu informações do planeta vermelho. A Mariner 6, passou por Marte em 31 de julho de 1969, tirou fotos e analisou a composição e pressão atmosférica de Marte. A Mariner 7 enviou fotografias do pólo sul de Marte. O lançamento da Mariner 8 não foi bem sucedido, levando a NASA lançar a nave Mariner 9, cujo código corretor de erros será estudado neste artigo. Esta nave entrou na órbita de Marte em 13 de novembro de 1971, 167 dias após o lançamento.

Recebido em 30/08/2017 - Aceito em 18/10/2017

---

RECEN 19(2) p.168-186 jul/dez 2017 DOI:10.5935/RECEN.2017.02.04

Nesta missão foi fotografado um majestoso vulcão com 27 km de altura, denominado “Monte Olimpo”, que já havia sido observado por telescópio, pelo astrônomo italiano Giovanni Schiaparelli (1835-1910), que descreveu como uma região de intenso brilho na superfície de Marte.

A teoria de códigos corretores de erros, foi fundada pelo matemático Claude Shannon (1916-2001), do Laboratório Bell de Nova Jersey, Estados Unidos da América (EUA), num trabalho publicado em 1948. Tal teoria tornou-se muito ativa a partir da década de 70 com a corrida espacial e a popularização dos computadores, sendo, até hoje, amplamente utilizada em diversas áreas do conhecimento: matemática, computação, engenharia elétrica, engenharia espacial, estatística entre outras.

Esta teoria é utilizada sempre que se deseja transmitir ou armazenar dados, garantindo a sua confiabilidade em setores como: comunicação via satélite, comunicações internas de um computador, armazenamento ótico de dados. Contudo, na transmissão ou armazenamento de dados, pode ocorrer interferências eletromagnéticas ou equívocos humanos (erros de digitação) que são chamados de ruídos, possibilitando que a mensagem recebida seja diferente da mensagem transmitida. A referida teoria tem como objetivo corrigir tais erros e fazer com que a mensagem transmitida pelo emissor seja de fato a mesma mensagem recebida pelo usuário.

Atualmente existem diversos trabalhos já publicados e disponíveis sobre a Teoria dos Códigos Corretores de Erros, por exemplo [2], [3], [4] e [5], que citam o código utilizado pela Mariner 9, mas de maneira superficial, apenas como exemplo de uma classe de códigos. O objetivo desse trabalho é explicar de maneira clara e completa o código específico utilizado pela Mariner 9, que pertence à classe dos códigos de Reed Muller ([6], [7], [8] e [9]) começando pela apresentação dos pré requisitos da Teoria de Códigos Corretores de Erros necessários para o total entendimento dessa codificação (Seção 2), passando pela apresentação, descrição e demonstração dos principais parâmetros dos Códigos de Reed Muller de Primeira Ordem (Seção 3) até a explicação em si da codificação e decodificação utilizada por essa nave espacial no envio de fotos de Marte para a Terra (Seção 4). Com isso, espera-se que esse texto sirva de material didático para alunos dos cursos de matemática, computação ou até mesmo engenharia, para profissionais da área de tecnologia da informação e também para professores de matemática e física tanto do Ensino Médio como do Ensino Superior aprenderem uma aplicação importante da matemática.

## 2 Conteúdos Básicos

Nesta seção são apresentadas as principais definições e os principais resultados da teoria dos códigos corretores de erros necessários para que o leitor entenda o código utilizado pela Mariner 9. Todos os resultados desta seção podem ser encontrados em [2] e [3].

Entende-se por código, de acordo com a teoria da comunicação, o conjunto de símbolos que devem ser conhecidos tanto pelo emissor quanto pelo receptor, de modo que a mensagem seja compreendida. Codificar a informação inicial, adicionando informação redundante, de tal forma que, ao receber o sinal modificado pelo “ruído” seja possível, de alguma forma, recuperar a mensagem original, esta é a ideia básica da teoria de códigos corretores de erros.

O ponto de partida para a construção de um código corretor de erros é construir um conjunto de símbolos finito  $\mathcal{F}$ , chamado alfabeto. O número de elementos de  $\mathcal{F}$  será denotado por  $q$ .

**Definição 2.1 (Códigos Corretores de Erros)** *Um código corretor de erros é um subconjunto próprio qualquer de  $\mathcal{F}^n$ , para algum  $n$  natural, onde  $\mathcal{F}^n = \underbrace{\mathcal{F} \times \mathcal{F} \times \dots \times \mathcal{F}}_{n \text{ vezes}}$ .*

O exemplo a seguir ilustra a definição acima.

**Exemplo 2.1** *O idioma português é um exemplo de um código corretor de erros. Dado o alfabeto  $\mathcal{F}$  da língua portuguesa, formado por 26 letras, bem como o espaço em branco, também considerado como uma letra, o "ç" cedilha e as vogais acentuadas: à, á, â, ã, é, ê, í, ó, ô, õ e ú (neste caso, o número de elementos de  $\mathcal{F}$  é  $q = 39$ ), uma palavra desta língua pode ser considerada como um elemento de  $\mathcal{F}^{46}$ , já que 46 é o comprimento da palavra mais longa da mesma, "pneumoultramicroscopicossilicovulcanoconiótico", segundo [10]. As outras palavras que não possuem 46 letras são completadas com espaços em branco do lado direito ao término da palavra, omitindo-os na escrita. Assim, o conjunto  $\mathcal{C}$  de todas as palavras da língua portuguesa é um subconjunto próprio de  $\mathcal{F}^{46}$  e, portanto, um código corretor de erros. Suponha que, ao escrever uma palavra, produza a sequência de letras "espatial". Como esta palavra não é um elemento de  $\mathcal{C}$ , percebe-se imediatamente que houve erro e, nesse caso, a correção é possível pois, a palavra de  $\mathcal{C}$  que mais se assemelha a "espatial" é "espacial". Percebe-se, porém, que este código não é muito eficiente, uma vez que, se a palavra "nave" for erroneamente escrita como "neve", ou ainda, como "nove", não se conseguiria detectar e muito menos corrigir o erro.*

Neste artigo trabalharemos apenas com códigos binários, isto é, códigos definidos sobre o alfabeto  $\mathcal{F}$  igual ao corpo  $\mathbb{F}_2 = \{0, 1\}$  cujas operações de soma e produto são dadas pela tabela abaixo.

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

Tabela 1: Adição e multiplicação em  $\mathbb{F}_2$

Segue abaixo outro exemplo de código, desta vez sobre o corpo  $\mathbb{F}_2$ .

**Exemplo 2.2 (Código da Nave)** *Supõe-se que um protótipo de uma nave espacial se mova a 20 metros de altura (acima do solo), de modo que, ao dar um dos comandos: Para Cima, Leste, Sudeste, Sul, Oeste, Noroeste, Norte ou Para Baixo, ele se desloca em uma destas direções. Estes oito comandos podem ser codificados como elementos de  $\mathbb{F}_2^3$ , como abaixo:*

<i>Para Cima</i> → 000
<i>Leste</i> → 001
<i>Sudeste</i> → 010
<i>Sul</i> → 011
<i>Oeste</i> → 100
<i>Noroeste</i> → 101
<i>Norte</i> → 110
<i>Para Baixo</i> → 111

Tabela 2: Codificação da fonte do código da nave

O código acima é chamado de “código da fonte”. Suponha que estes ternos ordenados devam ser transmitidos via rádio e que o sinal no caminho sofra interferências. Imagine que a mensagem 111(*Para Baixo*) possa, na chegada, ser recebida como 011(*Sul*), o que faria com que o protótipo, em vez de ir *Para Baixo*, fosse para o *Sul*. Numa tentativa de corrigir tal erro, pode-se fazer uma recodificação das palavras, de modo que permita detectar e corrigir os erros ocorridos na transmissão, acrescentando redundâncias nos códigos da fonte. Como na tabela abaixo:

<i>Para Cima</i> → 000 → 0000000
<i>Leste</i> → 001 → 0010111
<i>Sudeste</i> → 010 → 0101010
<i>Sul</i> → 011 → 0111101
<i>Oeste</i> → 100 → 1001100
<i>Noroeste</i> → 101 → 1011011
<i>Norte</i> → 110 → 1100110
<i>Para Baixo</i> → 111 → 1110001

Tabela 3: Codificação de canal do código da nave

Recodificando desta maneira, observe que os três primeiros símbolos reproduzem o código da fonte, enquanto os quatro restantes são redundâncias inseridas. O novo código inserido na recodificação é um código detector e corretor de erros, chamado de “código de canal”.

Suponha que seja inserido um erro ao transmitir uma das palavras, por exemplo, a palavra 1110001(*Para Baixo*), de modo que a mensagem recebida seja 1110000. Comparando essa mensagem com as palavras do código de canal, nota-se que ela não faz parte do mesmo e, portanto, detectam-se erros. A palavra deste código mais próxima da referida mensagem (a que tem menor número de elementos diferentes) é 1110001, que é, portanto, a palavra transmitida.

A teoria dos códigos corretores de erros consiste em transformar o código da fonte em código de canal, em detectar e corrigir erros na recepção das palavras e em decodificar o código de canal em código da fonte.

Consideram-se, neste trabalho, apenas canais simétricos, isto é, todos os símbolos transmitidos do código têm a mesma probabilidade de serem recebidos de forma errada.

Será apresentada a seguir uma forma de medir a distância entre palavras de um código em  $\mathbb{F}_2^n$ .

**Definição 2.2 (Distância de Hamming)** Dados dois elementos  $u = (u_1, u_2, \dots, u_n)$  e  $v = (v_1, v_2, \dots, v_n)$  com  $u, v \in \mathbb{F}_2^n$ , chama-se distância de Hamming entre  $u$  e  $v$  ao número de posições em que estes dois elementos diferem, isto é:

$$d(u, v) = |\{i : u_i \neq v_i, 1 \leq i \leq n\}|$$

Dado um código  $\mathcal{C} \subset \mathbb{F}_2^n$  chama-se de “distância mínima” do código  $\mathcal{C}$  o número:

$$d = \min \{d(u, v) : u, v \in \mathcal{C}, u \neq v\}$$

**Exemplo 2.3** No código da nave temos que:

$$d(0101010, 1110001) = 5$$

$$d(0111101, 1011011) = 4$$

$$d(0000000, 1001100) = 3$$

Observe que, a distância mínima do código da nave é  $d = 3$ .

A distância de Hamming, conforme definida acima, é uma métrica. Portanto, para todo  $u, v, w \in \mathbb{F}_2^n$ , temos as seguintes propriedades:

(i) Positividade:  $d(u, v) \geq 0$ , a igualdade acontece, se e somente se,  $u = v$ ;

(ii) Simetria:  $d(u, v) = d(v, u)$ ;

(iii) Desigualdade Triangular:  $d(u, v) \leq d(u, w) + d(w, v)$ .

A seguir, o Teorema 2.1 apresenta um dos principais resultados da teoria de códigos corretores de erros. Para a demonstração desse teorema necessitaremos, anteriormente, de duas definições e de um lema.

**Definição 2.3 (Menor Inteiro)** Dado um código  $\mathcal{C} \subset \mathbb{F}_2^n$  com distância mínima  $d$ , considere  $\eta$  a parte inteira de  $\frac{d-1}{2}$ , que será denotada por  $\eta = \lfloor \frac{d-1}{2} \rfloor$ .

**Definição 2.4 (Disco)** Dado um elemento  $x \in \mathbb{F}_2^n$  e um número real  $\eta > 0$ , definimos disco de centro  $x$  e raio  $\eta$  como sendo o conjunto:

$$D(x, \eta) = \{u \in \mathbb{F}_2^n : d(u, x) \leq \eta\}$$

**Lema 2.1** Sejam  $\mathcal{C} \subset \mathbb{F}_2^n$  um código com distância mínima  $d$ ,  $\eta = \lfloor \frac{d-1}{2} \rfloor$  e  $c$  e  $c'$  duas palavras de  $\mathcal{C}$ . Então  $D(c, \eta) \cap D(c', \eta) = \emptyset$ .

**Demonstração:** Suponha que exista  $x \in D(c, \eta) \cap D(c', \eta)$ . Assim  $d(c, x) \leq \eta$  e  $d(c', x) \leq \eta$ . Como  $d(c', x) = d(x, c')$  e, pela desigualdade triangular temos que

$$\begin{aligned} d(c, c') &\leq d(c, x) + d(x, c') \\ &\leq \eta + \eta \\ &\leq 2\eta \\ &\leq d - 1 \\ &< d \end{aligned}$$

Agora, isto é um absurdo, pois as palavras  $c$  e  $c' \in \mathcal{C}$  tem distância maior ou igual a  $d$ , já que  $d$  é a distância mínima de  $\mathcal{C}$ . Portanto  $D(c, \eta) \cap D(c', \eta) = \emptyset$ . ■

**Teorema 2.1** *Seja  $\mathcal{C} \subset \mathbb{F}_2^n$  um código com distância mínima  $d$ . Então:*

- (i)  $\mathcal{C}$  detecta até  $d - 1$  erros;
- (ii)  $\mathcal{C}$  corrige até  $\eta = \lfloor \frac{d-1}{2} \rfloor$  erros.

**Demonstração:**

- (i) Se  $d$  é a distância mínima do código  $\mathcal{C}$  então qualquer palavra que tenha até  $d - 1$  erros não pertence a  $\mathcal{C}$  e, portanto, seu erro será detectado;
- (ii) Seja  $c$  a palavra do código  $\mathcal{C}$  a ser transmitida e  $r$  a palavra recebida sendo cometidos  $t$  erros, com  $t \leq \eta$ , então  $d(r, c) = t \leq \eta$ , assim  $r \in D(c, \eta)$ . Logo, basta trocar  $r$  por  $c$  já que, pelo Lema 2.1, não há outra palavra de  $\mathcal{C}$  em  $D(c, \eta)$  que não seja  $c$ .

□

Observe que se  $c$  é a palavra a ser transmitida e foi recebida a palavra  $r$  com  $t$  erros, sendo  $t \leq \eta$ , como  $c$  é a palavra mais próxima do código  $\mathcal{C}$  então troca-se  $r$  por  $c$ . Mas não se tem garantia total de que a palavra transmitida foi  $c$ , pois poderia ter sido cometido mais que  $t$  erros o que levaria a outra palavra do código  $\mathcal{C}$  diferente de  $c$ .

**Exemplo 2.4** *Suponha que se queira mover a nave do exemplo 2.2 para cima. Neste caso, a mensagem a ser transmitida é  $c = 0000000$  (Para Cima). Mas, a mensagem recebida pelo receptor foi  $r = 1000000$ , ocorrendo 1 erro na transmissão. Como a distância mínima do código da nave é  $d = 3$  então este código detecta até 2 erros e corrige até  $\eta = \lfloor \frac{d-1}{2} \rfloor = 1$  erro. Portanto, esse erro será detectado e corrigido pelo código que trocará  $r$  por  $c$ .*

Pelo item (ii) do Teorema 2.1, segue uma definição importante para a correção de erros de um código.

**Definição 2.5 (Capacidade de Correção do Código)** *Dado um código  $\mathcal{C}$  com distância mínima  $d$ , a capacidade de correção do código é dada por:*

$$\eta = \left\lfloor \frac{d-1}{2} \right\rfloor$$

Interessa-nos códigos que tenham um número  $M$  de palavras relativamente grande, para que se possa transmitir muita informação e que tenha uma distância mínima  $d$  também grande, para se ter uma boa capacidade de correção de erros.

A seguir, vamos definir uma classe de códigos muito importante que será utilizado neste trabalho. Lembre-se que  $\mathbb{F}_2^n$  é um espaço vetorial sobre  $\mathbb{F}_2$  com soma e multiplicação por escalar usuais.

**Definição 2.6 (Códigos Lineares)** Um código  $\mathcal{C} \subset \mathbb{F}_2^n$  é chamado de código linear se for um subespaço vetorial de  $\mathbb{F}_2^n$ .

**Observação 2.1** Todo código linear é por definição um espaço vetorial de dimensão finita. Sejam  $k$  a dimensão do código  $\mathcal{C}$ ,  $\{v_1, v_2, \dots, v_k\}$  uma de suas bases e  $a_1, a_2, \dots, a_k$  escalares em  $\mathbb{F}_2$ . Todo vetor  $v \in \mathcal{C}$  se escreve como combinação linear dos vetores  $\{v_1, v_2, \dots, v_k\}$  de forma única, isto é:

$$v = a_1v_1 + a_2v_2 + a_3v_3 + \dots + a_kv_k$$

Logo, um código linear  $\mathcal{C} \subset \mathbb{F}_2^n$  de dimensão  $k$  possui  $2^k$  elementos.

**Exemplo 2.5** O código da nave

$$\mathcal{C} = \{0000000, 0010111, 0101010, 0111101, 1001100, 1011011, 1100110, 1110001\}$$

é um código linear pois o conjunto  $\mathcal{C}$  acima é fechado com relação à adição, ou seja, a soma de quaisquer duas palavras desse conjunto resulta em uma palavra de  $\mathcal{C}$ , fechado com relação à multiplicação por elementos de  $\mathbb{F}_2$  e também contém o elemento nulo. Logo,  $\mathcal{C}$  é um subespaço vetorial de  $\mathbb{F}_2^7$ .

**Definição 2.7 (Parâmetros de um Código)** Um código  $\mathcal{C} \subset \mathbb{F}_2^n$  possui três parâmetros fundamentais  $[n, M, d]$ , que são, respectivamente, o seu comprimento (o número  $n$  corresponde ao espaço ambiente  $\mathbb{F}_2^n$  onde  $\mathcal{C}$  se encontra), o seu número de elementos  $M$  e a sua distância mínima  $d$ .

**Exemplo 2.6** Vimos no exemplo 2.5 que o código da nave

$$\mathcal{C} = \{0000000, 0010111, 0101010, 0111101, 1001100, 1011011, 1100110, 1110001\}$$

é um código linear. Seus parâmetros são:  $n = 7$ ,  $M = 8$  e  $d = 3$ . Tal código também pode ser visto como a imagem da seguinte aplicação linear  $T: \mathbb{F}_2^3 \rightarrow \mathbb{F}_2^7$

$$(x_1, x_2, x_3) \mapsto (x_1, x_2, x_3, x_1 + x_2, x_1 + x_3, x_2 + x_3, x_3)$$

Por exemplo, a codificação  $(0, 1, 1)$  que denotaremos por 011 (código da fonte) é  $T(011) = 0111101$  (código de canal).

Veremos a seguir que a distância mínima pode ser calculada utilizando o peso de um código linear.

**Definição 2.8 (Peso de um Código Linear)** O peso de um código linear  $\mathcal{C}$ , que denotaremos por  $w(\mathcal{C})$ , é o peso mínimo de todas as palavras não nulas de  $\mathcal{C}$ , isto é,

$$w(\mathcal{C}) = \min \{w(u) : u \in \mathcal{C} \setminus \{0\}\}$$

onde  $w(u) = |\{i : u_i \neq 0\}|$  representa o número de caracteres não nulos da palavra  $u$ .

Observe que  $w(u) = d(u, 0)$ .

**Proposição 2.1** *Seja  $\mathcal{C} \subset \mathbb{F}_2^n$  um código linear com distância mínima  $d$ . Então:*

$$(i) \ d(u, v) = w(u - v), \quad \forall u, v \in \mathbb{F}_2^n;$$

$$(ii) \ d = w(\mathcal{C}).$$

**Demonstração:**

(i) Segue  $\forall u, v \in \mathbb{F}_2^n$ , com  $u = (u_1, u_2, \dots, u_n)$  e  $v = (v_1, v_2, \dots, v_n)$  que

$$\begin{aligned} w(u - v) &= |\{i : u_i - v_i \neq 0, 1 \leq i \leq n\}| \\ &= |\{i : u_i \neq v_i, 1 \leq i \leq n\}| \\ &= d(u, v). \end{aligned}$$

(ii) Para todo par de elementos  $u, v \in \mathcal{C}$ , com  $u \neq v$ , tem-se  $z = u - v \in \mathcal{C} \setminus \{0\}$ . Assim, temos

$$\begin{aligned} d &= \min \{i : u_i \neq v_i, 1 \leq i \leq n\} \\ &= \min \{i : u_i - v_i \neq 0, 1 \leq i \leq n\} \\ &= \min \{i : z_i \neq 0, 1 \leq i \leq n\} \\ &= \min \{w(z) : z \in \mathcal{C} \setminus \{0\}\} \\ &= w(\mathcal{C}) \end{aligned}$$

□

Observe que, como demonstrado na proposição 2.1, nos códigos lineares o peso coincide com a distância mínima do código, isto é,  $w(\mathcal{C}) = d$ . Em um código linear com  $M$  elementos, podemos calcular a distância mínima  $d$ , deste código, a partir do seu peso com  $M - 1$  cálculos de distâncias, em vez dos  $\binom{M}{2} = \frac{M(M-1)}{2}$  cálculos que deveriam ser feitos em um código qualquer, não linear, para o cálculo de  $d$ .

Veremos, na definição a seguir, que é usual colocar os elementos da base de um código linear  $\mathcal{C}$  numa matriz.

**Definição 2.9 (Matriz Geradora de um Código)** *Dados um código linear  $\mathcal{C} \subset \mathbb{F}_2^n$  de dimensão  $k$  sobre  $\mathbb{F}_2$  e  $\beta = \{u_1, u_2, \dots, u_k\}$  uma base ordenada de  $\mathcal{C}$ , considere a matriz  $G$ , cujas linhas são os vetores  $u_i = (u_{i1}, u_{i2}, \dots, u_{in})$ , com  $i = 1, 2, \dots, k$ :*

$$G = \begin{pmatrix} u_1 \\ \vdots \\ u_k \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ u_{k1} & u_{k2} & \cdots & u_{kn} \end{pmatrix}_{k \times n}$$

Tal matriz  $G$  é denominada “matriz geradora” do código  $\mathcal{C}$ , a qual não é única, dependendo da escolha da base  $\beta$ .

Dada a matriz  $G$ , matriz geradora de um código  $\mathcal{C}$ , para se codificar uma mensagem  $x$  utilizando tal código, basta fazermos  $x.G$ .

**Exemplo 2.7** O conjunto  $\beta = \{1001100, 0101010, 0010111\}$  é uma base do código  $\mathcal{C}$  da nave, já que os vetores de  $\beta$  são linearmente independentes e geram o conjunto  $\mathcal{C}$ . Disso temos a matriz geradora de  $\mathcal{C}$  como abaixo

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}_{3 \times 7}$$

De acordo com o exemplo 2.6, observe que a codificação de 011 é  $T(011) = 0111101$ , que nada mais é do que  $(011).G$ .

Agora, para decodificar a palavra 0111101 do código  $\mathcal{C}$ , isto é, achar a palavra  $x \in \mathbb{F}_2^3$ , tal que,  $T(x) = 0111101$ , basta resolver a equação  $(x_1, x_2, x_3).G = 0111101$ , o que implica:  $x_1 = 0, x_2 = 1, e x_3 = 1$ .

### 3 Códigos de Reed-Muller de 1ª Ordem

Os códigos de Reed-Muller foram criados em 1954, por David Eugene Muller (1924 – 2008). Neste mesmo ano, Irving Stoy Reed (1923 – 2012) descobriu o algoritmo de decodificação destes códigos. Estes códigos formam uma classe de códigos lineares sobre  $\mathbb{F}_2$  que possuem várias maneiras de serem definidos. Vamos, a seguir, dar uma definição recursiva para estes códigos.

Os códigos Reed-Muller de 1ª ordem -  $R(1, m)$  são códigos binários lineares definidos, recursivamente, por:

- $R(1, 0) = \{0, 1\} = \mathbb{F}_2$ .
- $R(1, 1) = \mathbb{F}_2 \times \mathbb{F}_2 = \{00, 01, 10, 11\} = \mathbb{F}_2^2$ .
- Para  $m > 1$ , defina :

$$R(1, m) = \left\{ u \ u, u \ (u + \bar{1}) \mid u \in R(1, m - 1) \text{ e } \bar{1} = \text{vetor } \underbrace{11 \dots 1}_{2^{m-1}} \right\}$$

Por exemplo,

$$R(1, 2) = \{u \ u, u \ (u + \bar{1}) \mid u \in R(1, 1)\} = \{0000, 0101, 1010, 1111, 0011, 0110, 1001, 1100\}$$

Utilizando  $R(1, 2)$ , obtemos:

$$R(1, 3) = \left\{ \begin{array}{cccc} 00000000 & 01010101 & 10101010 & 11111111 \\ 00110011 & 01100110 & 10011001 & 11001100 \\ 00001111 & 01011010 & 10100101 & 11110000 \\ 00111100 & 01101001 & 10010110 & 11000011 \end{array} \right\}$$

Através do  $R(1, 3)$ , obtemos o  $R(1, 4)$  e assim sucessivamente.

### 3.1 Parâmetros do Código de Reed-Muller de 1ª Ordem

De acordo com a definição 2.7, os parâmetros de um código são:  $[n, M, d]$ , onde  $n$  é o comprimento do código,  $M$  é a cardinalidade desse código e  $d$  é sua distância mínima.

Pela definição dos códigos de Reed-Muller de primeira ordem, temos que

$$R(1, 0) \subset \mathbb{F}_2^1 = \mathbb{F}_2^{2^0}$$

$$R(1, 1) \subset \mathbb{F}_2^2 = \mathbb{F}_2^{2^1}$$

$$R(1, 2) \subset \mathbb{F}_2^4 = \mathbb{F}_2^{2^2}$$

$$R(1, 3) \subset \mathbb{F}_2^8 = \mathbb{F}_2^{2^3}$$

Continuando esse raciocínio, teremos  $R(1, 4) \subset \mathbb{F}_2^{16} = \mathbb{F}_2^{2^4}$  e assim sucessivamente, obtendo:

$$R(1, m) \subset \mathbb{F}_2^{2^m}$$

Logo, o comprimento dos Códigos de Reed-Muller de Primeira Ordem, ou seja, o comprimento de  $R(1, m)$  é:

$$n = 2^m.$$

Agora, observe que a cardinalidade de  $R(1, 0)$  que será denotada aqui por  $|R(1, 0)|$ , é igual a 2,

$$|R(1, 1)| = 4 = 2^2, \quad |R(1, 2)| = 8 = 2^3, \quad |R(1, 3)| = 16 = 2^4$$

obtendo por indução que

$$|R(1, m)| = 2^{m+1}$$

Assim, o número de palavras de  $R(1, m)$  é

$$M = 2^{m+1}$$

Segue pela observação 2.1 que a dimensão do espaço vetorial  $R(1, m)$  sobre  $\mathbb{F}_2$  é  $k = m+1$ .

Vamos mostrar, agora, que a distância mínima do código Reed-Muller de 1ª ordem é  $d = 2^{m-1}$ .

Para isso, temos que mostrar que o peso de qualquer palavra de  $R(1, m)$ , exceto as palavras  $\bar{0} = \underbrace{000 \dots 0}_{2^m}$  e  $\bar{1} = \underbrace{111 \dots 1}_{2^m}$  é igual a  $2^{m-1}$ , que tem  $w(\bar{0}) = 0$  e  $w(\bar{1}) = 2^m$ . Com isso, segue pela Proposição 2.1 que  $d = w(R(1, m)) = 2^{m-1}$ .

**Teorema 3.1** *Seja  $c \in R(1, m)$ ,  $c \neq \bar{0} = \underbrace{000 \dots 0}_{2^m}$  e  $c \neq \bar{1} = \underbrace{111 \dots 1}_{2^m}$ . Então,  $w(c) = 2^{m-1}$ .*

**Demonstração:** (Vamos verificar a afirmação por indução em  $m$ )

Para  $m = 1$ , temos que  $R(1, 1) = \{00, 01, 10, 11\}$ , donde qualquer palavra,  $c \neq \bar{0} = 00$  e  $c \neq \bar{1} = 11$ , tem peso  $2^{1-1} = 1$ . Observe que, 01 e 10, ambas tem peso 1. Logo, o resultado é verdadeiro para  $m = 1$ .

Hipótese de Indução: Em  $R(1, m - 1)$  qualquer palavra,  $c \neq \bar{0} = \underbrace{000 \dots 0}_{2^{m-1}}$  e  $c \neq \bar{1} = \underbrace{111 \dots 1}_{2^{m-1}}$ , tem peso  $2^{(m-1)-1} = 2^{m-2}$ .

Observe que, em  $R(1, m)$  dizer que qualquer palavra,  $c \neq \bar{0} = \underbrace{000 \dots 0}_{2^m}$  e  $c \neq \bar{1} = \underbrace{111 \dots 1}_{2^m}$ , tem peso  $2^{m-1}$  equivale a dizer que ela é composta por metade 0's e metade 1's já que seu comprimento é  $2^m$  e  $2^{m-1} = \frac{2^m}{2}$ .

Seja  $c$  uma palavra de  $R(1, m)$ ,  $c \neq \bar{0} = \underbrace{000 \dots 0}_{2^m}$  e  $c \neq \bar{1} = \underbrace{111 \dots 1}_{2^m}$ .

Temos duas possibilidades:

(1)  $c = u u, u \in R(1, m - 1)$ .

Como  $c \neq \underbrace{000 \dots 0}_{2^m}$  e  $c \neq \underbrace{111 \dots 1}_{2^m}$ , então,  $u \neq \underbrace{000 \dots 0}_{2^{m-1}}$  e  $u \neq \underbrace{111 \dots 1}_{2^{m-1}}$ . Por hipótese de indução,  $w(u) = 2^{m-2}$ , ou seja,  $u$  tem  $2^{m-2}$  posições iguais a 1. Logo,  $c = u u$  terá  $2 \cdot 2^{m-2} = 2^{m-1}$  posições iguais a 1. Portanto,  $w(c) = 2^{m-1}$ .

(2)  $c = u (u + \bar{1}), u \in R(1, m - 1)$ .

(2.1) Se  $u = \underbrace{000 \dots 0}_{2^{m-1}}$ , então,  $u + 1 = \underbrace{111 \dots 1}_{2^{m-1}}$ . Logo,

$$c = \underbrace{000 \dots 0}_{2^{m-1}} \underbrace{111 \dots 1}_{2^{m-1}} \implies w(c) = 2^{m-1}$$

(2.2) Se  $u = \underbrace{111 \dots 1}_{2^{m-1}}$ , então,  $u + 1 = \underbrace{000 \dots 0}_{2^{m-1}}$ . Logo,

$$c = \underbrace{111 \dots 1}_{2^{m-1}} \underbrace{000 \dots 0}_{2^{m-1}} \implies w(c) = 2^{m-1}$$

(2.3) Caso  $u \neq \underbrace{000 \dots 0}_{2^{m-1}}$  e  $u \neq \underbrace{111 \dots 1}_{2^{m-1}}$  temos que,  $c = u (u + \bar{1})$ , onde  $u \in R(1, m - 1)$ .

Pela hipótese de indução,  $w(u) = 2^{m-2} = \frac{2^{m-1}}{2}$ , ou seja, metade das coordenadas de  $u$  são iguais a zero e metade das coordenadas de  $u$  são iguais a 1. Observe que, 0 em  $u$ , vira 1 em  $u + 1$  e, 1 em  $u$ , vira 0 em  $u + 1$ . Logo, a palavra  $u (u + 1)$  terá  $2 \cdot 2^{m-2}$  posições iguais a 1. Portanto,

$$w(c) = 2^{m-1}$$

□

### 3.2 Codificação do Código Reed-Muller de 1ª Ordem

A seguir, vai ser apresentada uma construção recorrente para a matriz geradora do código  $R(1, m)$ , que será denotada por  $G(1, m)$ .

Considere a matriz geradora de  $R(1, 1)$  por

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

Se  $G(1, m - 1)$  é a matriz geradora para  $R(1, m - 1)$ , então, a matriz geradora para  $R(1, m)$  é

$$G(1, m) = \begin{bmatrix} G(1, m - 1) & G(1, m - 1) \\ \underbrace{0 \dots 0}_{2^{m-1}} & \underbrace{1 \dots 1}_{2^{m-1}} \end{bmatrix}_{(m+1) \times 2^m}$$

Conforme visto anteriormente, a dimensão de  $R(1, m)$  sobre  $\mathbb{F}_2$  é igual a  $m + 1$ , o que implica que a matriz  $G(1, m)$  possui  $m + 1$  linhas. E como o comprimento de  $R(1, m)$  é  $2^m$ , a matriz  $G(1, m)$  possui  $2^m$  colunas.

Desta forma, temos, por exemplo:

$$G(1, 2) = \begin{bmatrix} G(1, 1) & G(1, 1) \\ 0 \dots 0 & 1 \dots 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$G(1, 3) = \begin{bmatrix} G(1, 2) & G(1, 2) \\ 0 \dots 0 & 1 \dots 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

e assim sucessivamente.

Para codificar uma mensagem  $b$  utilizando o código Reed-Muller de 1ª ordem, basta efetuar a operação:

$$b.G(1, m)$$

Como a matriz  $G(1, m)$  é uma matriz de tamanho  $(m + 1) \times 2^m$ , a mensagem  $b$ , ou o código da fonte, deverá ter comprimento  $m + 1$ , ou seja,  $b = (b_0, b_1, \dots, b_m)$  e o código de canal ou a mensagem codificada terá comprimento  $2^m$ .

**Exemplo 3.1** Para codificar uma mensagem usando a matriz geradora do código  $R(1, 3)$  de tamanho  $4 \times 8$ , a mensagem, código da fonte, deverá ter tamanho  $1 \times 4$ . A mensagem é codificada para a palavra do código do seguinte modo,

$$(b_0, b_1, b_2, b_3) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}_{4 \times 8} =$$

$$= (b_0, b_0 + b_1, b_0 + b_2, b_0 + b_1 + b_2, b_0 + b_3, b_0 + b_1 + b_3, b_0 + b_2 + b_3, b_0 + b_1 + b_2 + b_3)$$

### 3.3 Decodificação dos Códigos Reed-Muller - Reed Decoding

A decodificação dos códigos Reed-Muller, denominada “Reed Decoding”, é relativamente simples e será explicada neste trabalho através de um exemplo. Vamos considerar inicialmente o caso  $m = 3$ . Já sabemos que o código  $R(1, 3) \subset \mathbb{F}_2^8$  possui 16 palavras, tem dimensão 4 e distância mínima também igual a 4. Por isso, esse código detecta até 3 erros e corrige até 1 erro. Considere a matriz geradora do código  $R(1, 3)$  dada abaixo:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

As linhas desta matriz são os vetores de uma base do código  $R(1, 3)$  identificadas como  $\{v_0, v_1, v_2, v_3\}$ , nesta sequência, da primeira até a quarta linha. Qualquer palavra  $c$  deste código é uma combinação linear destes vetores, isto é,

$$c = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3, \text{ onde } a_i \in \mathbb{F}_2.$$

Assim qualquer vetor  $c$  do código  $R(1, 3)$  é da forma:

$$c = (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7) = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3).$$

Agora, note que: (lembre-se que em  $\mathbb{F}_2$  a soma de dois elementos iguais é zero)

$$a_1 = c_0 + c_1 = c_2 + c_3 = c_4 + c_5 = c_6 + c_7$$

$$a_2 = c_0 + c_2 = c_1 + c_3 = c_4 + c_6 = c_5 + c_7$$

$$a_3 = c_0 + c_4 = c_1 + c_5 = c_2 + c_6 = c_3 + c_7$$

Se não ocorrer nenhum erro na transmissão da palavra  $c$ , cada uma das 4 equações em cada linha acima resultará no valor de  $a_i$ ,  $i = 1, 2, 3$  correspondente. Caso ocorra erro na transmissão da palavra  $c$ , a palavra recebida será  $r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7)$  e, neste caso, os valores dos  $a_i$ 's serão dados por:

$$a_1 = r_0 + r_1 = r_2 + r_3 = r_4 + r_5 = r_6 + r_7$$

$$a_2 = r_0 + r_2 = r_1 + r_3 = r_4 + r_6 = r_5 + r_7$$

$$a_3 = r_0 + r_4 = r_1 + r_5 = r_2 + r_6 = r_3 + r_7$$

Observe, agora, que nem todas as 4 equações em cada linha vão coincidir (pois houve erro) e, neste caso, o valor de  $a_i$  será igual ao dígito que mais aparece nas 4 equações acima que determinam o respectivo  $a_i$ .

Já para encontrar o valor de  $a_0$ , vamos lembrar que

$$r = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3$$

$$a_0v_0 = r - (a_1v_1 + a_2v_2 + a_3v_3).$$

Como  $v_0 = \bar{1} = 11111111$ , então  $a_0v_0 = a_0$ ,

logo o valor de  $a_0$  será determinado pela maioria dos elementos que aparecem em

$$r - (a_1v_1 + a_2v_2 + a_3v_3).$$

Assim a palavra transmitida será recuperada por:

$$c = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3.$$

**Exemplo 3.2** *Suponha que seja transmitida a palavra  $c = 01010101 = v_1$  e recebida a palavra  $r = 01010100$  (observe que houve 1 erro no último dígito). Lembre-se que o código  $R(1,3)$  detecta até 3 erros e corrige até 1 erro. Portanto, neste caso, o erro será detectado e corrigido. Utilizando a decodificação Reed, temos que:*

$$\begin{aligned} a_1 &= r_0 + r_1 = r_2 + r_3 = r_4 + r_5 = r_6 + r_7 \\ a_2 &= r_0 + r_2 = r_1 + r_3 = r_4 + r_6 = r_5 + r_7 \\ a_3 &= r_0 + r_4 = r_1 + r_5 = r_2 + r_6 = r_3 + r_7 \end{aligned}$$

Os valores de  $a_1, a_2$  e  $a_3$  são obtidos da seguinte forma:

$$\begin{aligned} a_1 &= 0 + 1 = 0 + 1 = 0 + 1 = 0 + 0 \\ a_1 &= 1 = 1 = 1 = 0 \implies a_1 = 1 \end{aligned}$$

Observe que, conforme explicado anteriormente, como houve erro na transmissão, nem todas as equações foram iguais. Neste caso, consideramos como o valor de  $a_1$ , o dígito que mais aparece como resultado das 4 equações que, neste caso, foi  $a_1 = 1$ . Temos, a seguir, o mesmo raciocínio para  $a_2$  e  $a_3$ .

$$\begin{aligned} a_2 &= 0 + 0 = 1 + 1 = 0 + 0 = 1 + 0 \\ a_2 &= 0 = 0 = 0 = 1 \implies a_2 = 0 \end{aligned}$$

$$\begin{aligned} a_3 &= 0 + 0 = 1 + 1 = 0 + 0 = 1 + 0 \\ a_3 &= 0 = 0 = 0 = 1 \implies a_3 = 0 \end{aligned}$$

Para encontrar o valor de  $a_0$ , calculamos:

$$\begin{aligned} r - (a_1v_1 + a_2v_2 + a_3v_3) &= r - (1.v_1 + 0.v_2 + 0.v_3) \\ &= 01010100 - 01010101 \\ &= 00000001 \end{aligned}$$

Como a maioria dos dígitos da palavra encontrada são iguais a zero, então,  $a_0 = 0$ . Deste modo, encontramos a palavra transmitida calculando:

$$\begin{aligned} c &= a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3 \\ c &= 0.v_0 + 1.v_1 + 0.v_2 + 0.v_3 \end{aligned}$$

$c = v_1 = 01010101$ , que é a palavra transmitida corrigida de um erro.

## 4 O Código da Mariner 9

A nave espacial Mariner 9 ([11]) transmitiu para a Terra 7.329 fotografias, em preto e branco, que cobriram mais de 80% da superfície do planeta Marte. Estas fotografias revelaram leitos de rios, crateras, vulcões extintos, e um sistema de canyons com mais de 4.000 km de extensão, denominados “Valles Mariners”, em homenagem a nave espacial Mariner 9. Foram encontradas evidências de erosão eólica e hídrica, frentes meteorológicas, nevoeiros, e ainda, registradas as primeiras imagens das luas de Marte; Phobos e Deimos. Também foi obtida uma revelação surpreendente, a grande cratera encontrada em Marte, era um vulcão extinto, hoje chamado de “Monte Olimpo” (Figura 1), que possui mais de 20 km de altitude.

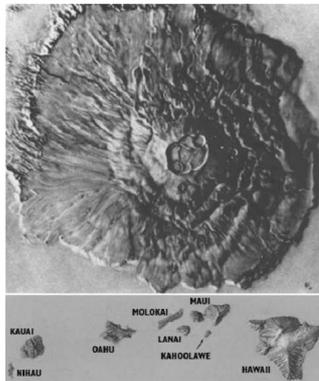


Figura 1: Monte Olimpo em comparação com o arquipélago do Havai. Fonte NASA

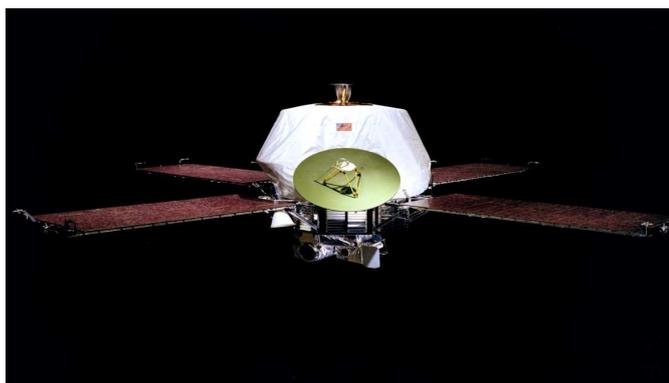


Figura 2: Nave Espacial Mariner 9. Fonte NASA

O código utilizado para a detecção e correção de erros dos dados enviados pela nave espacial Mariner 9 (Figura 2), à Terra, pertence à família de Códigos de Reed-Muller de Primeira Ordem,  $R(1, m)$ , para  $m = 5$ , ou seja, o código da Mariner 9 é o  $R(1, 5)$ .





$$a_3 = r_0 + r_4 = r_1 + r_5 = r_2 + r_6 = r_3 + r_7 = r_8 + r_{12} = r_9 + r_{13} = r_{10} + r_{14} = r_{11} + r_{15} = r_{16} + r_{20} = r_{17} + r_{21} = r_{18} + r_{22} = r_{19} + r_{23} = r_{24} + r_{28} = r_{25} + r_{29} = r_{26} + r_{30} = r_{27} + r_{31}$$

$$a_4 = r_0 + r_8 = r_1 + r_9 = r_2 + r_{10} = r_3 + r_{11} = r_4 + r_{12} = r_5 + r_{13} = r_6 + r_{14} = r_7 + r_{15} = r_{16} + r_{24} = r_{17} + r_{25} = r_{18} + r_{26} = r_{19} + r_{27} = r_{20} + r_{28} = r_{21} + r_{29} = r_{22} + r_{30} = r_{23} + r_{31}$$

$$a_5 = r_0 + r_{16} = r_1 + r_{17} = r_2 + r_{18} = r_3 + r_{19} = r_4 + r_{20} = r_5 + r_{21} = r_6 + r_{22} = r_7 + r_{23} = r_8 + r_{24} = r_9 + r_{25} = r_{10} + r_{26} = r_{11} + r_{27} = r_{12} + r_{28} = r_{13} + r_{29} = r_{14} + r_{30} = r_{15} + r_{31}$$

Depois de serem feitos todos estes cálculos, vamos obter pelo menos 9 dos 16 valores correspondentes para cada  $a_i$ , sendo assim, o valor correto será obtido pela maioria dos dígitos de cada  $a_i$ , isto é, o dígito que mais aparece na igualdade é o que será tomado como  $a_i$ . Finalmente,  $a_0$  pode ser determinado pela maioria dos dígitos de:

$$r - (a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4 + a_5v_5).$$

Assim, a mensagem transmitida corrigida de até 7 erros pode ser recuperada fazendo:

$$c = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4 + a_5v_5$$

Vamos mostrar como funciona o algoritmo com o exemplo a seguir.

**Exemplo 4.2** *Seja a mensagem transmitida  $c = 0101010101010101010101010101010101$  e, recebida a mensagem  $r = 01011001010111010101100101010110$  com 7 erros. Usando o algoritmo de Decodificação Reed desenvolvido por Irving Stoy Reed para decodificar os códigos Reed-Muller e, em especial, para decodificar o código  $R(1, 5)$ , temos:*

$$\begin{aligned} a_1 &= 1 = 1 = 1 = 1 = 1 = 1 = 0 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1, \text{ assim } a_1 = 1. \\ a_2 &= 0 = 0 = 1 = 1 = 0 = 0 = 1 = 0 = 0 = 0 = 1 = 1 = 0 = 0 = 1 = 1, \text{ assim } a_2 = 0. \\ a_3 &= 1 = 1 = 0 = 0 = 1 = 0 = 0 = 0 = 1 = 1 = 0 = 0 = 0 = 0 = 1 = 1, \text{ assim } a_3 = 0. \\ a_4 &= 0 = 0 = 0 = 0 = 0 = 1 = 0 = 0 = 0 = 0 = 0 = 0 = 1 = 1 = 1 = 1, \text{ assim } a_4 = 0. \\ a_5 &= 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 1 = 0 = 1 = 1, \text{ assim } a_5 = 0. \end{aligned}$$

Segue, então que, para encontrar o valor de  $a_0$ , fazemos:

$$\begin{aligned} r - (a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4 + a_5v_5) &= 01011001010111010101100101010110 - \\ &[1.(01010101010101010101010101010101) + 0.(0011001100110011001100110011) + \\ &0.(00001111000011110000111100001111) + 0.(00000000111111110000000011111111) + \\ &0.(00000000000000001111111111111111)] = 00001100000010000000110000000011. \end{aligned}$$

Como a maioria dos dígitos do resultado é zero, então, este é o valor de  $a_0$ , ou seja,  $a_0 = 0$ .

Desta forma, a mensagem transmitida  $c$ , corrigida dos 7 erros, é obtida por:

$$c = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4 + a_5v_5 = 0.v_0 + 1.v_1 + 0.v_2 + 0.v_3 + 0.v_4 + 0.v_5$$

Portanto,  $c = v_1 = 01010101010101010101010101010101$ , é a mensagem transmitida do código  $R(1, 5)$ , código de canal, corrigida dos 7 erros, que representa a tonalidade de cinza  $b = 010000$ , código da fonte.

## Referências

- [1] LABORATORY, J. P. California Institute of Technology. Disponível em <http://www.jpl.nasa.gov>. Acesso em 12/01/2016.
- [2] HEFEZ, A.; VILLELA, M. L. Códigos corretores de erros. 1ª ed. Rio de Janeiro: IMPA, 2002.
- [3] POLCINO, C. M. Breve introdução à teoria dos códigos corretores de erros. São Paulo: IME-USP, 2009.
- [4] MENEGHESSO, C. Códigos Corretores de Erros. Disponível em <https://www.dm.ufscar.br/dm/index.php/component/attachments/download/40> Acesso em 17/10/2017.
- [5] KOCH, H. C. R. Códigos Corretores de Erros e Teoria de Galois. Disponível em <http://www.mtm.ufsc.br/ebatista/2016-1/Helena.pdf> Acesso em 17/10/2017.
- [6] VICENTE, A. G. Um estudo dos Códigos de Reed Muller. Disponível em <https://repositorio.ufsc.br/bitstream/handle/123456789/157503/201961.pdf> Acesso em 17/10/2017.
- [7] RAAPHORST, S. Reed-Muller Codes. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.3214&rep=rep1&type=pdf>. Acesso em 17/10/2017.
- [8] MALEK, M. Coding Theory. Disponível em <http://www.mcs.csueastbay.edu/~malek/Class/Reed-Muller.pdf>. Acesso em 17/10/2017.
- [9] CHEROWITZO, B. Reed-Muller Codes. Disponível em <http://www-math.ucdenver.edu/wcherowi/courses/m7823/reedmuller.pdf>. Acesso em 17/10/2017.
- [10] HOUAISS, A. Dicionário Houaiss da Língua Portuguesa. 1.ed. Rio de Janeiro: Objetiva, 2009.
- [11] NASA JPL. Mariner 9 Mars Exploration. Disponível em <http://www.youtube.com/watch?v=JTNyoUj4mBI>. Acesso em 20/01/2016.