Universidade Federal de São João del-Rei - UFSJ

Campus Alto Paraopeba - CAP

José Silvino Dias

O CÓDIGO DA MARINER 9

Dissertação apresentada ao Departamento de Física e Matemática da Universidade Federal de São João del-Rei como parte dos requisitos exigidos para a obtenção do título de Mestre(a) pelo Programa de Mestrado Profissional em rede Nacional, PROFMAT.

Orientador(a)Mariana Garabini Cornelissen Hoyos

Ouro Branco 2017

Dissertação de Mestrado defendida em 03 de fevereiro de 2017 e aprovada pela Banca Examinadora composta pelos Professores.

Profa. Mariana Garabini Cornelissen Hoyos
Universidade Federal de São João del-Rei

Prof. Gil Fidelis de Souza
Universidade Federal de Ouro Preto

Prof. Marcelo Oliveira Veloso
Universidade Federal de São João del-Rei

2017: Trabalho de Conclusão de Curso do Mestrado Profissional em Matemática - PROFMAT
Universidade Federal de São João del-Rei - UFSJ / Campus Alto Paraopeba - CAP
Sociedade Brasileira de Matemática - SBM

O CÓDIGO DA MARINER 9

José Silvino Dias [1]
Mariana Garabini Cornelissen Hoyos[2]

Resumo: Este trabalho apresenta e descreve o código corretor de erros pertencente a uma família de códigos lineares, chamados Códigos 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). Também é apresentada uma proposta de atividade para os professores de matemática do 2º ano do Ensino Médio trabalharem com seus alunos a utilização prática de conteúdos de matrizes na codificação e decodificação de mensagens.

Palavras-chave: Mariner. Códigos. Reed-Muller.

Abstract: This study introduces and describes the error correcting code that belongs to a family of linear codes known as first order Reed-Muller Codes that were used by the space ship Mariner 9 that was sent to space by NASA (National Aeronautics and Space Administration) in 1971, while transmitting pictures of Mars to Earth. There is an activity proposal in this study as well for high school mathematics teachers to work with their students which is to pratice the use of matrices in the coding and decoding of messages.

Key words: Mariner, Codes, Reed- Muller.

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 [3], 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,

[1]Aluno de Mestrado Profissional em Matemática, Turma 2015
Instituição: Universidade Federal de São João del-Rei - UFSJ / Campus Alto Paraopeba - CAP
E-mail: josesilvino@iftm.edu.br
[2]Orientador do Trabalho de Conclusão de Curso
Departamento de Física e Matemática - Defim, UFSJ/CAP
E-mail: mariana@ufsj.edu.br

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 9.920 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 polo 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 trabalho. Esta nave entrou na órbita de Marte em 13 de novembro de 1971, 167 dias após o lançamento.

Nesta missão foi fotografado um majestoso vulcão com 27 km de altura, denominado "Monte Olimpo", que, por curiosidade, 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.

A segunda seção deste artigo apresenta os conceitos e resultados da teoria dos códigos corretores de erros, necessários para o entendimento do código utilizado na Mariner 9. Na terceira seção são apresentados os códigos de Reed-Muller de 1^{a} ordem, uma classe de códigos lineares da qual o código da Mariner 9 faz parte. A quarta seção traz a descrição e propriedades da codificação e decodificação do código utilizado pela Mariner 9. Na quinta seção será apresentada uma proposta de atividade para os professores de matemática trabalharem com os alunos do 2^{o} ano do Ensino Médio sobre aplicação de matrizes na codificação e decodificação de mensagens.

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 [1] e [5].

Entende-se por código, de acordo com a teoria da comunicação, como 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 $\mathcal{F}^n = \underbrace{\mathcal{F} \times \mathcal{F} \times \ldots \times \mathcal{F}}_{n \text{ vezes}}$.*

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 "c" 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 [2]. 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 $\mathbb{F}_2$ definido a seguir.

Definição 2.2 ($\mathbb{F}_2$) *Chamaremos de $\mathbb{F}_2$, o conjunto formado pelos dígitos $\{0, 1\}$ munido das seguintes operações:*

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

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

Observação 2.1 *O conjunto $\mathbb{F}_2$ com essas operações é chamado de Corpo.*

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

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:*

$Para\ Cima \longrightarrow 000$
$Leste \longrightarrow 001$
$Sudeste \longrightarrow 010$
$Sul \longrightarrow 011$
$Oeste \longrightarrow 100$
$Noroeste \longrightarrow 101$
$Norte \longrightarrow 110$
$Para\ Baixo \longrightarrow 111$

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:*

$Para\ Cima \longrightarrow 000 \longrightarrow 0000000$
$Leste \longrightarrow 001 \longrightarrow 0010111$
$Sudeste \longrightarrow 010 \longrightarrow 0101010$
$Sul \longrightarrow 011 \longrightarrow 0111101$
$Oeste \longrightarrow 100 \longrightarrow 1001100$
$Noroeste \longrightarrow 101 \longrightarrow 1011011$
$Norte \longrightarrow 110 \longrightarrow 1100110$
$Para\ Baixo \longrightarrow 111 \longrightarrow 1110001$

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 2.3 (Distância de Hamming) *Dados dois elementos $u = (u_1, u_2, \ldots, u_n)$ e $v = (v_1, v_2, \ldots, v_n)$ 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: $d(u, v) \leq d(u, w) + d(w, v)$.

Abaixo segue o Teorema 2.1 que 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 2.4 (Menor Inteiro) *Dado um código $\mathcal{C} \subset \mathbb{F}_2^n$ com distância mínima d, considere η a parte inteira de $\frac{d-1}{2}$, que será denotada por $\eta = \left[\frac{d-1}{2}\right]$.*

Definição 2.5 (Disco) *Dado um elemento $x \in \mathbb{F}_2^n$ e um número real $\eta > 0$, definimos disco de centro x e raio η 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, $\eta = [\frac{d-1}{2}]$ 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

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

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 $\eta = \left[\frac{d-1}{2}\right]$ 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 $\eta = \left[\frac{d-1}{2}\right] = 1$ erro. Portanto, esse erro será detectado e corrigido pelo código que trocará r por c.*

A partir do Teorema 2.1, segue uma definição importante para a correção de erros de um código.

Definição 2.6 (Capacidade de Correção do Código) *Dado um código $\mathcal{C}$ com distância mínima d, a capacidade de correção do código é dada por:*

$$\eta = \left[\frac{d-1}{2}\right]$$

Assim, como vimos no Teorema 2.1, é possível detectar até $d-1$ erros e corrigir até η erros.

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.

Definição 2.7 (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 2.2 *Todo código linear é por definição um espaço vetorial de dimensão finita. Sejam k a dimensão do código $\mathcal{C}$, $\{v_1, v_2, \ldots, v_k\}$ uma de suas bases e $a_1, a_2, \ldots, a_k$ escalares em $\mathbb{F}_2$. Todo vetor $v \in \mathcal{C}$ se escreve como combinação linear dos vetores $\{v_1, v_2, \ldots, v_k\}$ de forma única, isto é:*

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

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 2.8 (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 $T : \mathbb{F}_2^3 \longrightarrow \mathbb{F}_2^7$

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

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

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

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

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

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

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

(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

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

■

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 $\binom{M}{2} = M(\frac{M-1}{2})$ 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 2.10 (Matriz Geradora de um Código) *Dados um código linear $\mathcal{C} \subset \mathbb{F}_2^n$ de dimensão k sobre $\mathbb{F}_2^n$ e $\beta = \{u_1, u_2, \ldots, u_k\}$ uma base ordenada de $\mathcal{C}$, considere a matriz G, cujas linhas são os vetores $u_i = (u_{i1}, u_{i2}, \ldots, u_{in})$, com $i = 1, 2, \ldots, k$:*

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

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

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 β 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 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 :

$$R(1,m) = \left\{ u\ u, u\ (u+1) \mid u \in R(1,m-1) \text{ e } \bar{1} = \text{vetor } \underbrace{11\ldots1}_{2^{m-1}} \right\}$$

Por exemplo,

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

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.

O código $R(1,m)$ é um subespaço vetorial de $\mathbb{F}_2^{2^m}$. Pode-se mostrar que:
(i) $000\ldots0 \in R(1,m)$;

(ii) $R(1,m)$ é fechado para a adição, ou seja, a soma de quaisquer duas palavras de $R(1,m)$ resulta em uma palavra de $R(1,m)$;

(iii) $R(1,m)$ é fechado com relação à multiplicação por elementos de $\mathbb{F}_2$.

Portanto, o código $R(1,m)$ é um código linear.

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 2.2 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 $\bar{0} = \underbrace{000\ldots0}_{2^m}$ e $\bar{1} = \underbrace{111\ldots1}_{2^m}$ 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 $c \in R(1,m)$, $c \neq \bar{0} = \underbrace{000\ldots0}_{2^m}$ e $c \neq \bar{1} = \underbrace{111\ldots1}_{2^m}$. 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, $c \neq \bar{0} = \underbrace{000\ldots0}_{2^{m-1}}$ e $c \neq \bar{1} = \underbrace{111\ldots1}_{2^{m-1}}$, tem peso $2^{(m-1)-1} = 2^{m-2}$.

Observe que, em $R(1,m)$ dizer que qualquer palavra, $c \neq \bar{0} = \underbrace{000\ldots0}_{2^m}$ e $c \neq \bar{1} = \underbrace{111\ldots1}_{2^m}$, 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 $R(1,m)$, $c \neq \bar{0} = \underbrace{000\ldots0}_{2^m}$ e $c \neq \bar{1} = \underbrace{111\ldots1}_{2^m}$.

Temos duas possibilidades:

(1) $c = u\ u, u \in R(1, m-1)$.
Como $c \neq \underbrace{000\ldots0}_{2^m}$ e $c \neq \underbrace{111\ldots1}_{2^m}$, então, $u \neq \underbrace{000\ldots0}_{2^{m-1}}$ e $u \neq \underbrace{111\ldots1}_{2^{m-1}}$. 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 $2.2^{m-2} = 2^{m-1}$ posições iguais a 1. Portanto, $w(c) = 2^{m-1}$.

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

(2.1) Se $u = \underbrace{000\ldots0}_{2^{m-1}}$, então, $u+1 = \underbrace{111\ldots1}_{2^{m-1}}$. Logo,

$$c = \underbrace{000\ldots0}_{2^{m-1}}\underbrace{111\ldots1}_{2^{m-1}} \Longrightarrow w(c) = 2^{m-1}$$

.

(2.2) Se $u = \underbrace{111\ldots1}_{2^{m-1}}$, então, $u+1 = \underbrace{000\ldots0}_{2^{m-1}}$. Logo,

$$c = \underbrace{111\ldots1}_{2^{m-1}}\underbrace{000\ldots0}_{2^{m-1}} \Longrightarrow w(c) = 2^{m-1}$$

.

(2.3) Caso $u \neq \underbrace{000\ldots0}_{2^{m-1}}$ e $u \neq \underbrace{111\ldots1}_{2^{m-1}}$ temos que, $c = u\ (u+1)$, 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 2.2^{m-2} posições iguais a 1. Portanto,

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

■

A tabela a seguir, permite analisar alguns parâmetros do código Reed-Muller de 1ª ordem para diferentes valores de m.

m	n	M	d	k	η
1	2	4	1	2	0
2	4	8	2	3	0
3	8	16	4	4	1
4	16	32	8	5	3
5	32	64	16	6	7

Tabela 4: Parâmetros do código Reed-Muller de 1ª ordem

3.2 Matriz Geradora 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 é a matriz geradora para $R(1, m-1)$, então, a matriz geradora para $R(1, m)$ é

$$G(1,m) = \begin{bmatrix} G(1,m-1) & G(1,m-1) \\ 0\ldots0 & 1\ldots1 \end{bmatrix}$$

Conforme visto na seção 3.1, a dimensão de $R(1, m)$ sobre $\mathbb{F}_2$ é igual a $m+1$, 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.

$$G(1,m) = \begin{bmatrix} G(1,m-1) & G(1,m-1) \\ \underbrace{0\ldots0}_{2^{m-1}} & \underbrace{1\ldots1}_{2^{m-1}} \end{bmatrix}_{(m+1)\times 2^m}$$

Desta forma, temos, por exemplo:

$$G(1,2) = \begin{bmatrix} G(1,1) & G(1,1) \\ 0\ldots0 & 1\ldots1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$G(1,3) = \begin{bmatrix} G(1,2) & G(1,2) \\ 0\ldots0 & 1\ldots1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

e assim sucessivamente.

3.3 Codificação do Código Reed-Muller de 1ª Ordem

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, $b = (b_0, b_1, \ldots, b_m)$ 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 4×8, a mensagem, código da fonte, deverá ter tamanho 1×4. A mensagem é codificada para a palavra do código do seguinte modo,*

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

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

3.4 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,

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

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

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

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

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

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

$$r - (a_1 v_1 + a_2 v_2 + a_3 v_3)\,.$$

Assim a palavra transmitida será recuperada por:

$$c = a_0 v_0 + a_1 v_1 + a_2 v_2 + a_3 v_3.$$

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:*

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

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

$$\begin{gathered} a_1 = 0 + 1 = 0 + 1 = 0 + 1 = 0 + 0 \\ a_1 = 1 = 1 = 1 = 0 \Longrightarrow a_1 = 1 \end{gathered}$$

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*.*

$$\begin{gathered} a_2 = 0 + 0 = 1 + 1 = 0 + 0 = 1 + 0 \\ a_2 = 0 = 0 = 0 = 1 \Longrightarrow a_2 = 0 \end{gathered}$$

$$\begin{gathered} a_3 = 0 + 0 = 1 + 1 = 0 + 0 = 1 + 0 \\ a_3 = 0 = 0 = 0 = 1 \Longrightarrow a_3 = 0 \end{gathered}$$

Para encontrar o valor de a_0*, calculamos:*

$$\begin{aligned} r - (a_1 v_1 + a_2 v_2 + a_3 v_3) &= r - (1.v_1 + 0.v_2 + 0.v_3) \\ &= 01010100 - 01010101 \\ &= 00000001 \end{aligned}$$

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

$$\begin{gathered} c = a_0 v_0 + a_1 v_1 + a_2 v_2 + a_3 v_3 \\ c = 0.v_0 + 1.v_1 + 0.v_2 + 0.v_3 \end{gathered}$$

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

4 O Código da Mariner 9

A nave espacial Mariner 9 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.

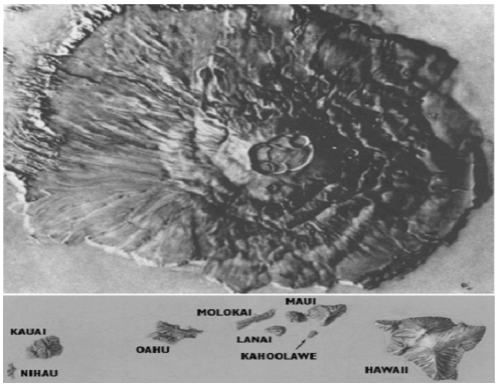

Figura 1 - Fonte Mariner 9 - NASA: Monte Olimpo em comparação com o arquipélago do Havai.

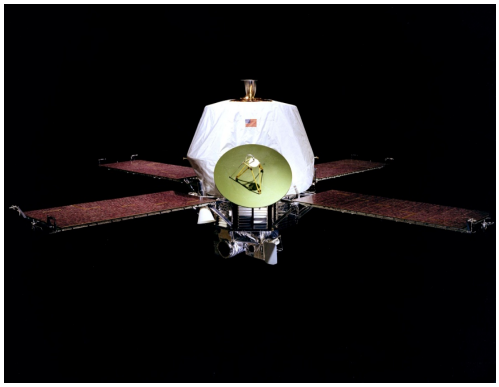

Figura 2 - Fonte NASA: Nave Espacial Mariner 9

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 $R(1, 5)$.

Conforme visto na seção 3.1, os parâmetros desse código são:

$$n = 32$$

$$M = 64$$

$$d = 16$$

Portanto,

- cada palavra deste código contém um comprimento igual a 32, ou seja, é uma sequência de 32 dígitos $0's$ e $1's$: isto significa que a codificação de canal, dada pelo $R(1, 5)$, transformou sequências binárias de 6 dígitos em sequências binárias de 32 dígitos, acrescentando 26 dígitos à codificação da fonte, através da multiplicação do código da fonte pela matriz geradora do código $R(1, 5)$.

- o código utilizado pela Mariner 9 possui 64 palavras: isto consiste em atribuir, pela codificação da fonte, a 64 tons de cinza pré-estabelecidos, sequências binárias de comprimento 6, sendo o branco denotado por 000000 e o preto por 111111. Já pela codificação de canal do $R(1, 5)$ essas sequências binárias de comprimento 6 são transformadas em sequências binárias de comprimento 32, as quais representam os mesmos 64 tons de cinza, sendo o branco denotado por $\underbrace{000\ldots0}_{32}$ e o preto por $\underbrace{111\ldots1}_{32}$.

- esse código detecta até 15 erros e corrige até 7 erros.

Exemplo 4.1 *Como exemplo, se* $b = 000011$ *representa uma determinada tonalidade de cinza, ao multiplicarmos* b *pela matriz* $G(1,5)$ *geradora do código* $R(1,5)$*,*

$$\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix}_{6\times 32}$$

obtemos

$c = 00000000111111111111111100000000$ *que é a mesma tonalidade de cinza, porém, representada agora com* 32 *dígitos, para possibilitar a detecção e correção de possíveis erros que podem ocorrer durante a transmissão.*

4.1 Decodificação da Mariner 9

Um dos motivos da utilização do código Reed-Muller $R(1,5)$ pela nave Mariner 9, é por causa do seu algoritmo de decodificação, visto na seção 3.4. Seja a matriz $G(1,5)$

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

As linhas dessa matriz formam uma base para o código $R(1,5)$ que identificaremos como $\{v_0, v_1, v_2, v_3, v_4, v_5\}$, nesta sequência, da primeira até a sexta linha. Qualquer vetor c do código é uma combinação linear destes vetores, isto é,

$$c = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4 + a_5v_5, \text{ onde } a_i \in \mathbb{F}_2.$$

Assim, se nenhum erro ocorrer, qualquer vetor c do código $R(1,5)$ é da forma:

$c = (c_0, c_1, \ldots, c_{31}) = (a_0, a_1, a_2, a_3, a_4, a_5).G(1,5) = (a_0, a_0+a_1, a_0+a_2, a_0+a_1+a_2, a_0+a_3,$
$a_0+a_1+a_3, a_0+a_2+a_3, a_0+a_1+a_2+a_3, a_0+a_4, a_0+a_1+a_4, a_0+a_2+a_4, a_0+a_1+a_2+a_4,$
$a_0+a_3+a_4, a_0+a_1+a_3+a_4, a_0+a_2+a_3+a_4, a_0+a_1+a_2+a_3+a_4, a_0+a_5, a_0+a_1+a_5,$
$a_0+a_2+a_5, a_0+a_1+a_2+a_5, a_0+a_3+a_5, a_0+a_1+a_3+a_5, a_0+a_2+a_3+a_5, a_0+a_1+a_2+a_3+a_5,$
$a_0+a_4+a_5, a_0+a_1+a_4+a_5, a_0+a_2+a_4+a_5, a_0+a_1+a_2+a_4+a_5, a_0+a_3+a_4+a_5,$
$a_0+a_1+a_3+a_4+a_5, a_0+a_2+a_3+a_4+a_5, a_0+a_1+a_2+a_3+a_4+a_5)$.

Logo, temos:

$a_1 = c_0 + c_1 = c_2 + c_3 = c_4 + c_5 = c_6 + c_7 = c_8 + c_9 = c_{10} + c_{11} = c_{12} + c_{13} = c_{14} + c_{15} =$
$c_{16} + c_{17} = c_{18} + c_{19} = c_{20} + c_{21} = c_{22} + c_{23} = c_{24} + c_{25} = c_{26} + c_{27} = c_{28} + c_{29} = c_{30} + c_{31}$

$a_2 = c_0 + c_2 = c_1 + c_3 = c_4 + c_6 = c_5 + c_7 = c_8 + c_{10} = c_9 + c_{11} = c_{12} + c_{14} = c_{13} + c_{15} =$
$c_{16} + c_{18} = c_{17} + c_{19} = c_{20} + c_{22} = c_{21} + c_{23} = c_{24} + c_{26} = c_{25} + c_{27} = c_{28} + c_{30} = c_{29} + c_{31}$

$a_3 = c_0 + c_4 = c_1 + c_5 = c_2 + c_6 = c_3 + c_7 = c_8 + c_{12} = c_9 + c_{13} = c_{10} + c_{14} = c_{11} + c_{15} =$
$c_{16} + c_{20} = c_{17} + c_{21} = c_{18} + c_{22} = c_{19} + c_{23} = c_{24} + c_{28} = c_{25} + c_{29} = c_{26} + c_{30} = c_{27} + c_{31}$

$a_4 = c_0 + c_8 = c_1 + c_9 = c_2 + c_{10} = c_3 + c_{11} = c_4 + c_{12} = c_5 + c_{13} = c_6 + c_{14} = c_7 + c_{15} = c_{16} + c_{24} = c_{17} + c_{25} = c_{18} + c_{26} = c_{19} + c_{27} = c_{20} + c_{28} = c_{21} + c_{29} = c_{22} + c_{30} = c_{23} + c_{31}$

$a_5 = c_0 + c_{16} = c_1 + c_{17} = c_2 + c_{18} = c_3 + c_{19} = c_4 + c_{20} = c_5 + c_{21} = c_6 + c_{22} = c_7 + c_{23} = c_8 + c_{24} = c_9 + c_{25} = c_{10} + c_{26} = c_{11} + c_{27} = c_{12} + c_{28} = c_{13} + c_{29} = c_{14} + c_{30} = c_{15} + c_{31}$

Se não ocorrer nenhum erro na transmissão da mensagem c, cada uma das 16 equações em cada linha acima resultará no valor de $a_i = 1, 2, 3, 4, 5$ correspondente. Caso ocorra erro na transmissão da palavra c, a palavra recebida será $r = (r_0, r_1, \cdots, r_{31})$ e, neste caso, os valores dos $a_i's$ serão dados por:

$a_1 = r_0 + r_1 = r_2 + r_3 = r_4 + r_5 = r_6 + r_7 = r_8 + r_9 = r_{10} + r_{11} = r_{12} + r_{13} = r_{14} + r_{15} = r_{16} + r_{17} = r_{18} + r_{19} = r_{20} + r_{21} = r_{22} + r_{23} = r_{24} + r_{25} = r_{26} + r_{27} = r_{28} + r_{29} = r_{30} + r_{31}$

$a_2 = r_0 + r_2 = r_1 + r_3 = r_4 + r_6 = r_5 + r_7 = r_8 + r_{10} = r_9 + r_{11} = r_{12} + r_{14} = r_{13} + r_{15} = r_{16} + r_{18} = r_{17} + r_{19} = r_{20} + r_{22} = r_{21} + r_{23} = r_{24} + r_{26} = r_{25} + r_{27} = r_{28} + r_{30} = r_{29} + r_{31}$

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

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

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

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* $c = 0101\mathbf{01}0101010\mathbf{1}0101010\mathbf{01}0101010\mathbf{01}$ *e, recebida a mensagem* $r = 0101\mathbf{10}0101011\mathbf{1}0101010\mathbf{10}0101010\mathbf{10}$ *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:*

$a_1 = 1 = 1 = 1 = 1 = 1 = 1 = 0 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1$, *assim* $a_1 = 1$.
$a_2 = 0 = 0 = 1 = 1 = 0 = 0 = 1 = 0 = 0 = 0 = 1 = 1 = 0 = 0 = 1 = 1$, *assim* $a_2 = 0$.
$a_3 = 1 = 1 = 0 = 0 = 1 = 0 = 0 = 0 = 1 = 1 = 0 = 0 = 0 = 0 = 1 = 1$, *assim* $a_3 = 0$.
$a_4 = 0 = 0 = 0 = 0 = 0 = 1 = 0 = 0 = 0 = 0 = 0 = 0 = 1 = 1 = 1 = 1$, *assim* $a_4 = 0$.
$a_5 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 1 = 0 = 1 = 1$, *assim* $a_5 = 0$.

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

$r-(a_1v_1+a_2v_2+a_3v_3+a_4v_4+a_5v_5) = 0101100101011101010110010101011 0-$
$[1.(0101010101010101010101010101010101)+0.(0011001100110011001100110011001 1)+$
$0.(0000111100001111000011110000111 1)+0.(0000000011111111000000001111111 1)+$
$0.(0000000000000000111111111111111 1)] = 0000110000001000000011000000001 1.$

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, $c = v_1 = 0101010101010101010101010101010 1$, 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.

5 Atividade de Aprendizagem para o 2º Ano do Ensino Médio

Nesta seção será apresentada uma atividade para os professores de matemática do 2º ano do Ensino Médio mostrarem aos seus alunos uma aplicação do conteúdo de matrizes na codificação e decodificação de mensagens de códigos binários.

Para que seja realizada esta atividade, o professor já deverá ter ensinado aos alunos a teoria de matrizes, incluindo a multiplicação de matrizes e resolução de sistemas lineares.

Para despertar o interesse dos alunos, o professor de matemática pode utilizar de um documentário sobre a viagem da nave espacial Mariner 9 ao planeta Marte. Para isso, basta acessar o site em [4] e, em seguida, falar da importância da aplicação de matrizes no sucesso da missão.

A linguagem utilizada pelo computador é o sistema binário que é empregado para representar os números e as letras do nosso alfabeto, neste sistema a soma e a multiplicação dos elementos do conjunto $\mathbb{F}_2 = \{0,1\}$ é feita da seguinte forma: a soma $0 + 0 = 0$, $0 + 1 = 1 + 0 = 1$ e $1 + 1 = 0$ e a multiplicação $0 \,.\, 0 = 0$, $0 \,.\, 1 = 1 \,.\, 0 = 0$, finalmente, $1 \,.\, 1 = 1$.

Primeiramente precisamos definir uma codificação inicial que transforma as letras e os outros caracteres do nosso alfabeto em sequências de $0's$ e $1's$. A título de exemplo, vamos codificar somente as letras maiúsculas e o espaço em branco, como abaixo:

espaço = 00000

A = 10000	B = 01000	C = 00100	D = 00010	E = 00001	F = 11000
G = 10100	H = 10010	I = 10001	J = 01100	K = 01010	L = 01001
M = 00110	N = 00101	O = 00011	P = 11100	Q = 10110	R = 10101
S = 11010	T = 11001	U = 01110	V = 00111	X = 11110	Z = 11111

Essa primeira codificação é chamada de código da fonte.

O processo de transmissão de mensagens de um código pode sofrer interferências que modificam a informação transmitida. Sendo assim, a informação inicial (código da fonte) é codificada novamente, adicionando informação redundante (código de canal), de tal modo que ao receber o sinal modificado seja possível recuperar a informação original.

Podemos fazer essa outra codificação utilizando a multiplicação de matrizes. Por exemplo, considere a matriz G dada abaixo:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{5\times 9}$$

Para acharmos a outra codificação (chamada codificação de canal) da letra A, por exemplo, fazemos a seguinte multiplicação de matrizes:

$$10000.G = ... = 100000110$$

Assim a letra A que era inicialmente denotada por 10000, com essa nova codificação obtida através da multiplicação de matrizes, passou a ser denotada por 100000110. Como a codificação inicial da letra A era uma matriz de tamanho 1×5 e a matriz G tem tamanho 5×9, observe que o produto é possível, gerando uma nova matriz que representa a letra A de tamanho bem maior igual a 1×9. Isso permite que, caso ocorra algum erro na transmissão dessa letra, esse erro possa ser detectado e até mesmo corrigido!

Utilizando a matriz G acima, pede-se:

a) Codifique a palavra "MARTE".

b) Decodifique a mensagem abaixo, recebida da nave espacial Mariner 9, admitindo que não houve erro na transmissão da mensagem.

001101101 000110011 001011110 110010011 000011001

Para resolverem a letra b), os alunos devem considerar, por exemplo, que o código de canal seja a palavra 110010011 e para decodificá-la, ou seja, encontrar o código da fonte correspondente temos que achar uma matriz X tal que $X.G = 110010011$. Essa matriz X deve ter tamanho 1×5, já que a matriz G tem tamanho 5×9 e a matriz 110010011 tem tamanho 1×9. Então, basta fazermos $(x_1, x_2, x_3, x_4, x_5).G = 110010011$, o que implica resolver o sistema:

$$\begin{cases} x_1 = 1 \\ x_2 = 1 \\ x_3 = 0 \\ x_4 = 0 \\ x_5 = 1 \\ x_2 + x_4 + x_5 = 0 \\ x_1 + x_2 + x_3 = 0 \\ x_1 + x_3 + x_4 = 1 \\ x_3 + x_5 = 1 \end{cases}$$

A seguir, o professor pode observar com seus alunos que a matriz X é exatamente as cinco primeiras posições da palavra a ser decodificada. Logo, a palavra 110010011 é facilmente decodificada como 11001.

Após esse exercício, pode-se falar também sobre o código da nave, apresentado no exemplo 2.2 desse texto, mostrar sua codificação de fonte e codificação de canal e exibir com este exemplo a importância da codificação de canal na detecção e correção de erros num possível erro no envio de um comando.

Considerações Finais

Foi apresentado neste artigo um estudo sobre o código de Reed-Muller de 1ª ordem $R(1, 5)$, um código corretor de erros utilizado pela nave espacial Mariner 9 para o envio de imagens do planeta Marte para a Terra em 1971. Para isso, foram utilizados conceitos da Àlgebra e Aritmética, conceitos esses que podem ser aplicados por professores de matemática do 2º ano do Ensino Médio para ensinar sistema binário e matrizes na codificação e decodificação de códigos corretores de erros. A atividade de aprendizagem proposta neste artigo não foi aplicada em sala de aula. A sugestão é que o professor de matemática possa trabalhar esta atividade adequando-a no seu plano de aula.

Agradecimentos

Agradeço a Deus, por ter me dado a graça de realizar este sonho, à minha esposa Penha, aos meus filhos Ana Luíza e Fellippe, pela compreensão e incentivo neste período de estudo. Agradeço a minha orientadora, Profª. Dra. Mariana Garabini Cornelissen Hoyos, pelo empenho, disponibilidade e contribuições com este artigo. Agradeço à banca composta pelos professores Dr. Marcelo Oliveira Veloso (UFSJ) e Dr. Gil Fidelis de Souza (UFOP) pelas contribuições neste trabalho. À CAPES e SBM por tornarem possível a realização deste mestrado.

Referências

[1] HEFEZ, A.; VILLELA, M.L. *Códigos corretores de erros*. 1ª ed. Rio de Janeiro: IMPA, 2002.

[2] HOUAISS, A. *Dicionário Houaiss da Língua Portuguesa*. 1.ed. Rio de Janeiro: Objetiva, 2009.

[3] LABORATORY, J. P. *California Institute of Technology*. Disponível em http://www.jpl.nasa.gov. Acesso em 12/01/2016.

[4] NASA JPL. Mariner 9 Mars Exploration. Disponível em http://www.youltube.com/watch?v=JTNyoUj4mBI. Acesso em 20/01/2016.

[5] POLCINO, C. M. *Breve introdução à teoria dos códigos corretores de erros*. São Paulo: IME-USP, 2009.