UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE SÃO JOÃO DA BOA VISTA

ISABELLA SILVA TEIXEIRA

Decodificação de Códigos Matriciais MDS via Matrizes Superregulares com Simplificação por Matrizes de Vandermonde para Aplicação em Sistemas de Armazenamento Distribuído

São João da Boa Vista
2025

Isabella Silva Teixeira

Decodificação de Códigos Matriciais MDS via Matrizes Superregulares com Simplificação por Matrizes de Vandermonde para Aplicação em Sistemas de Armazenamento Distribuído

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Eletrônica e de Telecomunicações do Campus de São João da Boa Vista, Universidade Estatual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Eletrônica e de Telecomunicações .

Orientadora: Profª Drª. Cintya Wink de Oliveira Benedito

São João da Boa Vista
2025

T266d

Teixeira, Isabella Silva

Decodificação de códigos matriciais MDS via matrizes superregulares com simplificação por matrizes de Vandermonde para aplicação em sistemas de armazenamento distribuído / Isabella Silva Teixeira. -- São João da Boa Vista, 2025

87 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Telecomunicações) - Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, São João da Boa Vista

Orientadora: Cintya Wink de Oliveira Benedito

1. Códigos corretores de erros (Teoria da informação). 2. Grupos de matrizes. 3. Teoria de Galois. 4. Codificação. 5. Armazenamento de dados. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA - CÂMPUS DE SÃO JOÃO DA BOA VISTA
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES

TRABALHO DE CONCLUSÃO DE CURSO

DECODIFICAÇÃO DE CÓDIGOS MATRICIAIS MDS VIA MATRIZES SUPERREGULARES COM SIMPLIFICAÇÃO POR MATRIZES DE VANDERMONDE PARA APLICAÇÃO EM SISTEMAS DE ARMAZENAMENTO DISTRIBUÍDO

Aluno: Isabella Silva Teixeira
Orientador: Prof.ª Dr.ª Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientadora)
- Sara Díaz Cardell (Examinadora)
- Débora Beatriz Claro Zanitti (Examinadora)

Os formulários de avaliação e a ata da defesa, na qual consta a aprovação do trabalho, devidamente assinados pela banca encontram-se no prontuário eletrônico do aluno.

São João da Boa Vista, 05 de dezembro de 2025

AGRADECIMENTOS

Agradeço, em primeiro lugar, aos meus pais, Moacir Teixeira e Marina Silva Teixeira, pelo amor, apoio incondicional e por acreditarem em mim em todos os momentos desta jornada acadêmica, tornando possível a realização deste sonho.

Aos meus amigos, que estiveram ao meu lado nos períodos de maior desafio, compartilhando palavras de incentivo, compreensão e momentos de descontração que tornaram esta caminhada mais leve e especial.

Registro também minha profunda gratidão a todos os professores e funcionários da Unesp de São João da Boa Vista que fizeram parte da minha formação, pelos ensinamentos, pela dedicação em sala de aula e pela paciência em guiar meu aprendizado ao longo do curso.

De modo especial, agradeço à minha orientadora, Prof.ª Dra. Cintya Wink de Oliveira Benedito, por sempre ter me incentivado ao longo da graduação, acreditando no meu potencial e apoiando cada passo da minha trajetória acadêmica. Foi graças ao seu incentivo que participei do projeto de extensão Ciência no Feminino, iniciei minha caminhada na iniciação científica e tive a oportunidade de colaborar na publicação de um artigo, experiências que ampliaram minha formação e foram fundamentais para a construção deste trabalho. Sua dedicação, amizade, paciência e disponibilidade em orientar fizeram toda a diferença neste percurso.

À banca examinadora, Profª. Dra. Sara Díaz Cardell e Me. Débora Beatriz Claro Zanitti, pela disponibilidade e cuidado que depositaram ao aceitarem avaliar este trabalho.

Ao projeto de extensão Ciência no Feminino, no qual tive a oportunidade de atuar como bolsista, pelos aprendizados práticos e pela convivência com a equipe.

À empresa Danone, pela confiança ao me conceder a oportunidade de contratação mesmo antes da conclusão deste trabalho, contribuindo de forma decisiva para o meu crescimento profissional.

Por fim, a todos que, direto ou indiretamente, contribuíram para a realização deste Trabalho de Conclusão de Curso, deixo registrado o meu sincero muito obrigada. Sou muito grata à Deus por ter colocado essa oportunidade e pessoas maravilhosas no meu caminho para que esse momento fosse possível.

“Your life is an occasion. Rise to it“
(Suzanne Weyn)

RESUMO

Este Trabalho de Conclusão de Curso apresenta os conceitos algébricos fundamentais para a construção de códigos matriciais MDS (*Maximum Distance Separable*), caracterizados por possuírem distância máxima de separação, oferecendo, assim, maior proteção contra falhas em dispositivos. Esses códigos têm despertado interesse devido à sua capacidade de corrigir erros em rajada, tornando-se altamente eficientes em sistemas sujeitos a falhas correlacionadas. O objetivo deste trabalho é apresentar o algoritmo de decodificação proposto em (CARDELL; CLIMENT; REQUENA, 2013) e (ZANITTI, 2021) para códigos matriciais MDS capazes de corrigir até duas rajadas de erro utilizando matrizes superregulares, bem como propor uma simplificação do procedimento por meio do uso de matrizes superregulares de Vandermonde, reduzindo a complexidade do processo de decodificação sem comprometer sua eficiência. Além disso, propõe-se uma generalização do algoritmo para o caso de três erros, com uma simplificação análoga baseada em matrizes superregulares de Vandermonde. Os códigos estudados são aplicáveis em sistemas de armazenamento distribuído, como os RAIDs (*Redundant Array of Independent Disks*), e são apresentados como alternativa ao RAID 6, cuja codificação está ligada aos códigos Reed-Solomon e permite a correção de até dois discos com falhas quando suas posições são conhecidas. O diferencial dos algoritmos apresentados neste trabalho está no fato de não ser necessário conhecer previamente a localização dos erros para realizar sua correção.

PALAVRAS-CHAVE: codificação; decodificação; códigos matriciais mds; matrizes de vandermonde; erros em rajadas; sistemas de armazenamento distribuído; raid 6.

ABSTRACT

This work presents the fundamental algebraic concepts involved in the construction of MDS (*Maximum Distance Separable*) array codes, which are characterized by having maximum separation distance, thereby offering greater protection against device failures. These codes have attracted interest due to their ability to correct burst errors, making them highly efficient in systems subject to correlated failures. The objective of this work is to present the decoding algorithm proposed in (CARDELL; CLIMENT; REQUENA, 2013) and (ZANITTI, 2021) for MDS array codes capable of correcting up to two burst errors using superregular matrices, as well as to propose a simplification of the procedure through the use of Vandermonde superregular matrices, reducing the complexity of the decoding process without compromising its efficiency. In addition, a generalization of the algorithm is proposed for the case of three errors, with an analogous simplification based on Vandermonde superregular matrices. The codes studied are applicable to distributed storage systems, such as RAID (*Redundant Array of Independent Disks*) architectures, and are presented as an alternative to RAID 6, whose encoding relies on Reed–Solomon codes and enables the correction of up to two disk failures when their positions are known. The distinguishing feature of the algorithms presented in this work lies in the fact that prior knowledge of the error locations is not required for error correction.

KEYWORDS: coding; decoding; mds array codes; vandermonde matrices; burst errors; distributed storage systems; raid 6.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CRC	Cyclic Redundancy Check
ECC	Error Correcting Code
GF	Galois Field
HD	Hard Disk
MDS	Maxium Distance Separable
RAID	Redundant Array of Independent Disks
XOR	Exclusive OR

LISTA DE SÍMBOLOS

$\mathbb{A}$	Conjunto não vazio
A	Matriz superregular
α	Elemento primitivo
B	Unidade de blocos
b	Grau do polinômio primitivo e comprimento da rajada de erro
β	Base
C	Matriz companheira de Frobenius
$\mathcal{C}$	Código
c	Palavra-código recebida
D	Bloco de dados dos RAIDs
d	Distância mínima do código
d_h	Distância de Hamming
e	Vetor erro
$\mathbb{F}_q$	Corpo finito
$\mathbb{G}$	Grupo
G	Matriz geradora de um código
H	Matriz verificação de paridade
I	Matriz identidade
k	Dimensão do código e representação dos dados ou informações
l	Localização na palavra-código
m	Representação da paridade em códigos MDS
N	Unidade de discos
n	Comprimento do código
P	Matriz de dimensão $k \times (n-k)$ e primeiro bloco de paridade do RAID 6
P_{erro}	Probabilidade de erro

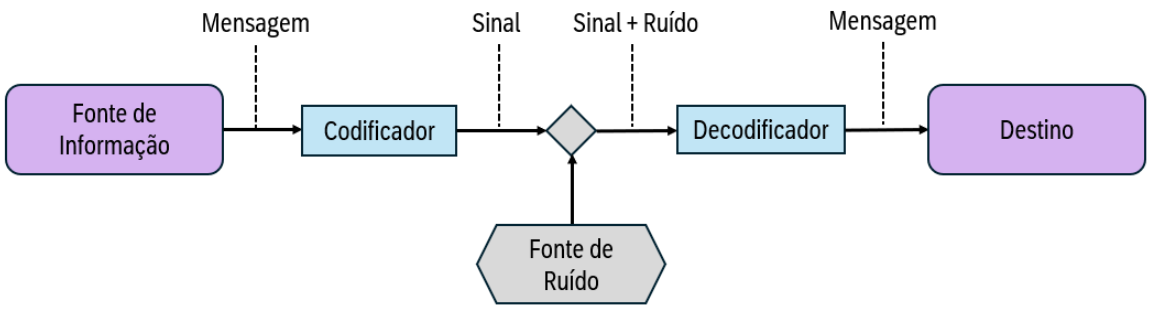
p	Número primo
$p(x)$	Polinômio primitivo
Q	Segundo bloco de paridade do RAID 6
q	Número inteiro primo
r	Expoente da Matriz de Frobenius
$\mathbb{S}$	Subconjunto do espaço vetorial
s	Síndrome do código
$\sigma(i, j)$	Potência do elemento primitivo
u	Mensagem a ser codificada
$\mathbb{V}$	Espaço vetorial
v	Vetor de informação recebida
w	Peso mínimo do código
y	Vetor auxiliar para decodificação de um código matricial MDS
$\mathbb{Z}$	Conjunto dos números inteiros
$Z(n)$	Logaritmo de Zech de n
$\oplus$	Soma exclusiva
$\otimes$	Produto externo
$\oslash$	Divisão conjunto

SUMÁRIO

1 INTRODUÇÃO

A teoria da codificação desempenha um papel crucial quando pretende transmitir ou receber informações, uma vez que a presença de ruídos ou falhas causadas por interferências no canal pode comprometer o conteúdo da mensagem (RAYN; LIN, 2009). Os códigos corretores de erros são um campo de pesquisa altamente dinâmico nos tempos atuais, englobando diversas áreas do conhecimento, incluindo matemática, ciência da computação, engenharias e estatística (MILIES, 2009). Eles também desempenham um papel fundamental em várias atividades cotidianas, como em sistemas de comunicações digitais, exemplificado na Figura 1, e sistemas de armazenamento.

Figura 1 – Sistema de comunicação digital.



Fonte: Elaborado pela autora.

O objetivo essencial desta teoria é desenvolver métodos eficazes para a detecção e correção desses erros, garantindo assim a integridade, confiabilidade e precisão das informações transmitidas. A confiabilidade da transmissão é adquirida por meio da codificação da informação através de bits de redundância, que inevitavelmente trazem consigo um custo. Isso resulta em um dos principais desafios da teoria de códigos: alcançar o nível desejado de confiabilidade com um menor custo possível, permitindo, assim, a detecção e correção de erros (FIRER, 2017).

Ao longo de todo o processo de transferência de informações, erros podem ocorrer tanto na codificação quanto na recepção do sinal. Porém, esses erros ocorrem predominantemente nos canais de transmissão, que incluem canal de micro-ondas, cabos, canal de radiofrequência, circuitos integrados digitais e dispositivos de armazenamento (HEFEZ; VILLELA, 2008). Esses erros podem acontecer de maneira simples, afetando apenas um símbolo, ou de forma sequencial, conhecido como rajada, que são erros que impactam vários símbolos consecutivos. Assim, surge a necessidade de investigar e desenvolver códigos corretores de erros eficazes que permitam a recuperação da informação afetada.

Os códigos corretores de erros em formato matricial fazem parte de uma categoria de códigos lineares bidimensionais que se destacam pela sua capacidade de corrigir erros em formato de rajada (BLAUM; FARRELL; TILBORG, 1998). Os códigos matriciais podem ser gerados a partir de diversos métodos e são conhecidos por sua flexibilidade e facilidade tanto na codificação quanto na decodificação.

Um método para obter códigos corretores de erros de alta qualidade é buscar códigos que possuam a máxima distância mínima possível. Os códigos que apresentam a propriedade de máxima distância de separação são conhecidos como códigos MDS (*Maximum Distance Separable*), os quais proporcionam uma proteção máxima contra falhas de um dispositivo, utilizando uma quantidade específica de redundância para tal propósito (ROMAN, 1992).

Um exemplo de aplicação desses códigos são nos sistemas de armazenamento distribuído, denominados como *Redundant Array of Independent Disks*, conhecidos popularmente como RAIDs, mostrados na Figura 2. A tecnologia do RAID 6 usa esses códigos MDS baseados no corpo de Galois para codificar dados nas unidades para proteger dados de erro ou apagamento (JACOB, 2009).

Figura 2 – Sistema de armazenamento distribuído.

Fonte: Elaborado pela autora.

A concepção fundamental do RAID consiste em unir N unidades de disco menores e mais econômicas em um conjunto, a fim de atingir objetivos de desempenho ou redundância que seriam impossíveis de alcançar com uma única unidade. Para atingir tal meta, emprega-se técnicas como o *disk striping* (RAID 0), o *disk mirroring* (RAID 1) e o *disk striping with parity* (RAID 5), as quais visam a reduzir a latência, ampliar a banda de passagem, maximizar a capacidade de recuperação de falhas no disco rígido e alcançar a redundância desejada (ARPACI-DUSSEAU, 2008).

Este Trabalho de Conclusão de Curso tem como propósito principal o estudo da abordagem algébrica aplicada aos processos de codificação e decodificação de códigos matriciais MDS para correção de rajadas de erro, fundamentada nas técnicas apresentadas em (CARDELL; CLIMENT; REQUENA, 2013) e (ZANITTI, 2021). Tais códigos têm demonstrado elevado potencial em sistemas de comunicação e armazenamento distribuído, em especial quando construídos a partir de matrizes superregulares, sendo empregados em reparo ótimo de nós defeituosos (YE; BARG, 2017), reparos cooperativos (YE; BARG, 2019) e no projeto de códigos matriciais localmente reparáveis (FANG; WEIJUN; LV, 2024). Neste trabalho, o objetivo é comparar essas construções aos sistemas RAIDs, com ênfase no RAID 6, cujas técnicas de codificação, usualmente baseadas em códigos Reed-Solomon ou EVENODD, permitem a correção de até dois discos defeituosos quando suas posições são previamente conhecidas (BLAUM; HAFNER; HETZLER, 2013; HSIEH; CHEN; LIN, 2004). Uma formulação

algébrica alternativa para um sistema equivalente ao RAID 6, utilizando métodos distintos dos aqui abordados, pode ser vista em (MOUSSA; RYCHLIK, 2018).

Neste trabalho apresenta-se o desenvolvimento de um algoritmo de decodificação capaz de corrigir até duas rajadas de erro em códigos matriciais MDS, sem a necessidade de conhecimento prévio das posições das falhas, tal como apresentado em (CARDELL; CLIMENT; REQUENA, 2013) e (ZANITTI, 2021). A principal contribuição deste Trabalho de Conclusãoo de Curso consiste em uma simplificação do algoritmo baseada na utilização de matrizes de Vandermonde como superregulares. Além disso, também investiga-se a possibilidade de estender o algoritmo para cenários com mais de duas rajadas de erro, superando a capacidade de correção tradicionalmente atribuída ao RAID 6. Para o caso de três rajadas de erro, detalham-se os cálculos e propõe-se novamente uma simplificação fundamentada na matriz de Vandermonde.

O presente trabalho está estruturado da seguinte forma: no Capítulo 2 é abordado os principais conceitos algébricas que serão necessários para o desenvolvimento deste trabalho. No Capítulo 3 são apresentados conceitos e resultados relacionados aos códigos corretores de erros, em especial os códigos de bloco lineares e os códigos matriciais, assim como os códigos de máxima distância de separação (MDS). No Capítulo 4 é apresentado os principais conceitos das Matriz Redundante de Discos Independentes (RAIDs) e sua classificação em níveis. No Capítulo 5 é apresentado a codificação e a decodificação de códigos matriciais MDS, com ênfase nos algoritmos de decodificação para dois e três rajadas de erros bem como suas simplificações. E, por fim, no Capítulo 6 são apresentadas as conclusões obtidas no desenvolvimento desde trabalho.

2 CONCEITOS PRELIMINARES

Neste capítulo apresentamos alguns conceitos fundamentais da álgebra abstrata. Este capítulo possui caráter introdutório e tem como objetivo apresentar e exemplificar os principais tópicos indispensáveis à compreensão da construção dos códigos e ao desenvolvimento do estudo, sem priorizar demonstrações matemáticas rigorosas. Na Seção 2.1, são abordados conceitos relacionados a estruturas algébricas, como corpos e anéis, espaços vetoriais e extensões de Galois. Já na Seção 2.2 é apresentado o conceito de logaritmo de Zech. E por fim, na Seção 2.3, é introduzido o conceito de matriz superregular, com destaque para a matriz de Vandermonde e Cauchy.

2.1 ESTRUTURAS ALGÉBRICAS

Nesta seção serão discutidos os conceitos e definições fundamentais de estruturas algébricas, incluindo grupos, anéis, corpos e espaços vetoriais. Também será explorado as operações e manipulações que podem ser realizadas em um corpo. Os conceitos e resultados apresentados nesta seção podem ser encontrados em (RAYN; LIN, 2009), (DOMINGUES; IEZZI, 1982), (HUBER, 1990), (ZANITTI, 2021), (JACOB, 2009) e (BENEDITO, 2010).

2.1.1 Grupos, Anéis, Corpos e Espaços Vetoriais

Definição 2.1.1 *Um **grupo** é um conjunto não vazio $\mathbb{G}$ com uma operação binária " * ", de modo que os seguintes axiomas sejam satisfeitos:*

1. *Fechamento da Operação: qualquer que sejam a e $b \in \mathbb{G}$, têm-se que $a * b$ também $\in \mathbb{G}$;*
2. *Associatividade: qualquer que sejam a, b e $c \in \mathbb{G}$ tem-se:*

$$(a * b) * c = a\ (b * c);$$

3. *Elemento Identidade: existe um elemento $e \in \mathbb{G}$ tal que para qualquer elemento a de $\mathbb{G}$ tem-se:*

$$a * e = e * a = a;$$

4. *Elemento Inverso: para qualquer elemento $a \in \mathbb{G}$, existe um outro elemento $a' \in \mathbb{G}$ tal que*

$$a * a' = a' * a = e;$$

5. *Comutatividade: O grupo $\mathbb{G}$ com operação "*" é denotado por ($\mathbb{G}$, *) e é dito **abeliano** ou **comutativo** se, para quaisquer elementos a e $b \in \mathbb{G}$, tiver $a * b = b * a$.*

Definição 2.1.2 *Um grupo $\mathbb{G}$ é chamado de finito (ou infinito) se ele contiver um número finito (ou infinito) de elementos. O número de elementos em um grupo finito é chamado de **ordem** do grupo.*

Exemplo 2.1.1 *Considerando um grupo simples com dois elementos, dado por* $\mathbb{Z} = \{0, 1\}$. *Definimos a operação binária, denotada por* $\oplus$*, em* $\mathbb{Z}$*, mostrada na Tabela 1.*

Tabela 1 – Adição em $\mathbb{Z}_2$.

$\oplus$	0	1
0	0	1
1	1	0

Fonte: Produção da Própria Autora.

A partir da tabela, podemos ver claramente que

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0.$$

Também sabemos que o conjunto $\mathbb{Z}_2$ *é fechado sob a operação* $\oplus$ *e é comutativa. Para provar que* $\oplus$ *é associativa, basta determinar* $(a \oplus b) \oplus c$ *e* $a \oplus (b \oplus c)$ *para as oito combinações possíveis de* a*,* b *e* c*, onde* a*,* b *e* $c \in \mathbb{Z}_2$*.*

Além disso, vemos que 0 é o elemento identidade, o inverso de 0 é ele mesmo e o inverso de 1 também é ele mesmo. Assim, $\mathbb{Z}_2$ *com a operação* $\oplus$ *é um grupo comutativo de ordem 2. A operação binária* $\oplus$ *definida no conjunto* $\mathbb{Z}_2$ *é chamada de adição módulo 2.*

Definição 2.1.3 *Um conjunto não vazio* $\mathbb{A}$ *e um par de operações sobre* $\mathbb{A}$*, respectivamente uma adição "+"e uma multiplicação "*$\cdot$*", é chamado* ***anel*** *se:*

1. $(\mathbb{A}, +)$ *é um grupo abeliano:*
   - (a) *Sejam* $a, b, c \in \mathbb{A}$*, então* $a + (b + c) = (a + b) + c$ *(associatividade);*
   - (b) *Se* $a, b \in \mathbb{A}$*, então* $a + b = b + a$ *(comutatividade);*
   - (c) *Existe um elemento* $0_A \in \mathbb{A}$ *tal que, qualquer que seja* $a \in \mathbb{A}$*,* $a + 0_A = a$ *(elemento neutro);*
   - (d) *Qualquer que seja* $a \in \mathbb{A}$*, existe um elemento em* $\mathbb{A}$*, denotado por* $-a$*, tal que* $a + (-a) = 0_A$ *(existência de oposto).*
2. *A multiplicação satisfaz a propriedade associativa* $a \cdot (b \cdot c) = (a \cdot b) \cdot c \; \forall \; a, b, c \in \mathbb{A}$*.*
3. *É distributiva em relação à adição* $a \cdot (b + c) = (a \cdot b + a \cdot c)$ *e* $(a + b) \cdot c = (a \cdot c + b \cdot c)$ $\forall \; a, b, c \in \mathbb{A}$*.*

Um anel $\mathbb{A}$ *com as operações adição "+" e multiplicação "*$\cdot$*" é denotado por* $(\mathbb{A}, +, \cdot)$*.*

A partir da Definição 2.1.3, temos ainda:

- **Anéis Comutativos** quando a multiplicação do anel $\mathbb{A}$ satisfazer a associatividade $\forall \; a, b \in \mathbb{A}$;
- **Anéis com Unidade** quando a multiplicação admitir um elemento neutro, $1 \in \mathbb{A}$, tal que $a \cdot 1 = 1 \cdot a = a$, para todo $a \in \mathbb{A}$.

Definição 2.1.4 *Seja $\mathbb{F}$ um conjunto não vazio para o qual são definidas duas operações binárias, adição "$+$" e multiplicação "$\cdot$". O conjunto $\mathbb{F}$ com estas duas operações é um* ***corpo****, e denota-se por $(\mathbb{F}, +, \cdot)$, se:*

1. *For um anel comutativo com unidade onde todo elemento é não inversível;*
2. *A multiplicação é distributiva em relação à adição, ou seja, dado quaisquer elementos a, b e $c \in \mathbb{F}$, tem-se*

$$a \cdot (b + c) = a \cdot b + a \cdot c.$$

O número de elementos de um corpo pode ser finito ou infinito, representados de acordo com a quantidade de elementos.

Definição 2.1.5 *Um corpo $\mathbb{F}$ com uma quantidade finita de elementos q é chamado de* ***corpo finito*** *e será denotado por $\mathbb{F}_q$. Se q for um número inteiro primo, o corpo finito é chamado de* ***corpo primo****.*

Exemplo 2.1.2 *De acordo com o Exemplo 2.1.1, $\mathbb{Z}_2$ é um grupo comutativo em relação à adição. A partir da Tabela 2 dada abaixo, a multiplicação módulo-2 verifica-se como um corpo finito.*

Tabela 2 – Multiplicação em $\mathbb{Z}_2$.

$\cdot$	0	1
0	0	0
1	0	1

Fonte: Produção da Própria Autora.

Exemplo 2.1.3 *Agora considerando $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$, nota-se que as operações são fechadas no conjunto, como mostrado nas Tabelas 3 e 4.*

Tabela 3 – Adição módulo-5.

$+$	*0*	*1*	*2*	*3*	*4*
0	*0*	*1*	*2*	*3*	*4*
1	*1*	*2*	*3*	*4*	*0*
2	*2*	*3*	*4*	*0*	*1*
3	*3*	*4*	*0*	*1*	*2*
4	*4*	*0*	*1*	*2*	*3*

Fonte: Produção da Própria Autora.

Tabela 4 – Multiplicação módulo-5.

$\cdot$	*0*	*1*	*2*	*3*	*4*
0	*0*	*0*	*0*	*0*	*0*
1	*0*	*1*	*2*	*3*	*4*
2	*0*	*2*	*4*	*1*	*3*
3	*0*	*3*	*1*	*4*	*2*
4	*0*	*4*	*3*	*2*	*1*

Fonte: Produção da Própria Autora.

Baseado na Tabela 3, observa-se que todos os elementos possuem elemento inverso aditivo, multiplicativo e nulos comprovando assim que $\mathbb{Z}_5$ é um corpo, mostrado no exemplo a seguir.

Definição 2.1.6 *Seja $\mathbb{F}$ um corpo e $\mathbb{V}$ um conjunto. O conjunto $\mathbb{V}$, munido de duas operações soma e multiplicação por escalar, é chamado de um* ***espaço vetorial*** *sobre o corpo $\mathbb{F}$ se satisfaz os seguintes axiomas:*

1. *$\mathbb{V}$ é um grupo comutativo sob adição $+$ definida em $\mathbb{V}$.*
2. *Para qualquer elemento a em $\mathbb{F}$ e qualquer elemento v em $\mathbb{V}$, $a \cdot v$ é um elemento em $\mathbb{V}$.*
3. *Para quaisquer elementos u e v de $\mathbb{V}$ e quaisquer elementos a e b de $\mathbb{F}$, verifica-se a propriedade distributiva.*
4. *Para qualquer elemento a em $\mathbb{F}$ e quaisquer elementos u e v em $\mathbb{V}$, verifica-se a propriedade associativa.*
5. *Seja 1 o elemento neutro de $\mathbb{F}$. Então, para qualquer elemento v em $\mathbb{V}$,*

$$1 \cdot v = v.$$

Exemplo 2.1.4 *Seja $\mathbb{V} = \mathbb{Z}_2^4$ o conjunto de todas as 4-uplas sobre $\mathbb{Z}_2 = \{0, 1\}$. $\mathbb{V}$ é um espaço vetorial sobre $\mathbb{Z}_2$ e consiste de $2^4 = 16$ vetores dado por*

$$\mathbb{V} = \{0000, 0001, 0010, 0100, 1000, 0011, 0101, 1001, 0110, 1010, 1100, 0111, 1011, 1101, 1110, 1111\}.$$

Se somarmos quaisquer dois vetores como $[0001]$ e $[1001]$ obtemos outra 4-upla

$$[0001] + [1001] = [1000].$$

Definição 2.1.7 *Seja $\mathbb{S}$ um subconjunto não vazio do espaço vetorial $\mathbb{V}$ sobre o corpo $\mathbb{F}$. O conjunto $\mathbb{S}$ é um **subespaço vetorial** de $\mathbb{V}$ se, e somente se, as seguintes propriedades forem satisfeitas:*

1. *Para quaisquer vetores u e v de $\mathbb{S}$, a soma $u + v$ também é um vetor de $\mathbb{S}$.*
2. *Para qualquer elemento a de $\mathbb{F}$ e qualquer vetor u de $\mathbb{S}$, o produto $a \cdot u$ também pertence a $\mathbb{S}$.*

Definição 2.1.8 *Sejam $v_1, v_2, \ldots, v_k$ k vetores de um espaço vetorial $\mathbb{V}$ sobre um corpo $\mathbb{F}$. Dado um conjunto de escalares $a_1, a_2, \ldots, a_k$ de $\mathbb{F}$, a soma $a_1v_1 + a_2v_2 + \cdots + a_kv_k$ é chamada de **combinação linear** de $v_1, v_2, \ldots, v_k$.*

Definição 2.1.9 *Um conjunto de vetores $v_1, v_2, \ldots, v_k$ pertencente a um espaço vetorial $\mathbb{V}$ sobre um corpo $\mathbb{F}$ é dito **linearmente independente** se, e somente se, existirem k escalares não todos nulos $a_1, a_2, \ldots, a_k$ de $\mathbb{F}$ tais que $a_1v_1 + a_2v_2 + \cdots + a_kv_k = 0$*

Exemplo 2.1.5 *Os vetores $[1000], [0100], [0010]$ e $[0001]$ são linearmente independentes, pois a combinação só é satisfeita quando todos os coeficientes são nulos.*

$$0 \cdot [1000] + 0 \cdot [0100] + 0 \cdot [0010] + 0 \cdot [0001] = [0000]$$

Um conjunto de vetores gera um espaço vetorial $\mathbb{V}$ se qualquer vetor de $\mathbb{V}$ puder ser expresso como uma combinação linear desses vetores.

Definição 2.1.10 *Em um espaço vetorial, o conjunto de vetores linearmente independentes que o gera é denominado **base** do espaço. A quantidade de vetores que compõem essa base define a **dimensão** do espaço vetorial.*

2.1.2 Corpos Finitos e suas Extensões

Corpos finitos, também conhecidos como corpos de Galois, desepenham papel central na teoria de códigos e comunicações digitais (RAYN; LIN, 2009).

É definido como um corpo finito da forma $\mathbb{F}_p = \mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$, onde p é um número primo. A partir de $\mathbb{F}_p$, pode-se construir uma extensão de corpos da seguinte forma. Seja b um inteiro positivo, define-se $\mathbb{F}_{p^b}$ como uma **extensão de Galois** que contém $\mathbb{F}_p$ como subcorpo. Com o objetivo de simplificar a notação, considera-se $q = p^b$, de modo que $\mathbb{F}_q = \mathbb{F}_{p^b}$.

Definição 2.1.11 *Seja a um elemento não nulo pertencente ao corpo finito $\mathbb{F}_q$. O menor inteiro positivo n tal que $a^n = 1$ é denominado **ordem do elemento**.*

Definição 2.1.12 *Um elemento α de um corpo finito $\mathbb{F}_q$ é denominado **elemento primitivo** se*

$$\mathbb{F}_q = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}, \tag{2.1}$$

ou seja, se a ordem de α for $q-1$.

Um polinômio $p(x) \in \mathbb{F}_q[x]$ pode ser representado da forma

$$p(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n, \tag{2.2}$$

onde $p_0, p_1, \ldots, p_n \in \mathbb{F}_q$.

Definição 2.1.13 *O grau do polinômio é a maior potência de x cujo coeficiente é diferente de zero e é dito **mônico** quando o coeficiente do termo de maior grau é igual a 1.*

Definição 2.1.14 *Um polinômio $p(x)$ de grau b sobre $\mathbb{F}_q$ é denominado **irredutível** se não puder ser fatorado como produto de polinômios não constantes em $\mathbb{F}_q$, ou seja, se não admitir divisores de grau estritamente menor que b, exceto unidades.*

Teorema 2.1.1 *Todo polinômio irredutível $p(x)$ sobre $\mathbb{F}_q$, de grau b, divide $x^{q^b-1} - 1$.*

Definição 2.1.15 *Um **polinômio primitivo** é um polinômio irredutível cujo elemento raiz gera todos os elementos não nulos de $\mathbb{F}_q$.*

Para um dado inteiro positivo b pode haver mais do que um polinômio irredutível de grau b. Esses polinômios primitivos podem ser encontrados por ferramentas computacionais ou por tabelas disponíveis em livros, como listado em (RAYN; LIN, 2009) e mostrado na Tabela 5.

Para construir uma extensão de um corpo de Galois $\mathbb{F}_p = \{0, 1, 2, \ldots, p-1\}$, sendo p um número primo, o primeiro passo é selecionar um polinômio primitivo de grau b sobre $\mathbb{F}_p$, ou seja,

Tabela 5 – Polinômios primitivos selecionados até grau 7.

b	p(x)
2	$x^2 + x + 1$
3	$x^3 + x^2 + 1$ $x^3 + x + 1$
4	$x^4 + x^3 + 1$ $x^4 + x + 1$
5	$x^5 + x^4 + x^3 + x^2 + 1$ $x^5 + x^4 + x^3 + x + 1$ $x^5 + x^4 + x^2 + x + 1$ $x^5 + x^3 + x^2 + x + 1$ $x^5 + x^3 + 1$ $x^5 + x^2 + 1$
6	$x^6 + x^5 + x^4 + x + 1$ $x^6 + x^5 + x^3 + x^2 + 1$ $x^6 + x^5 + 1$ $x^6 + x^4 + x^3 + x + 1$ $x^6 + x + 1$
7	$x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$ $x^7 + x^4 + x^3 + x^2 + 1$ $x^7 + x^3 + x^2 + x + 1$ $x^7 + x^3 + 1$ $x^7 + x + 1$

Fonte: Baseado em (RAYN; LIN, 2009).

$$p(x) = p_0 + p_1 x + p_2 x^2 + \ldots + p_{b-1} x^{b-1} + x^b \in \mathbb{F}_p[x]. \tag{2.3}$$

Desde que o grau de $p(x)$ seja b, o polinômio terá b raízes, as quais não pertencem a $\mathbb{F}_p$ e sim à sua extensão $\mathbb{F}_{p^b}$. Dado α uma raiz de $p(x)$, pode-se substituir x por α na Equação 2.3 e obter:

$$p(\alpha) = p_0 + p_1 \alpha + p_2 \alpha^2 + \ldots + p_{b-1} \alpha^{b-1} + \alpha^b = 0. \tag{2.4}$$

Através do Teorema 2.1.1, sabe-se que $p(x)$ divide $x^{p^b-1} - 1$, então:

$$x^{p^b-1} - 1 = p(x) \cdot q(x). \tag{2.5}$$

Substituindo-se x por α,

$$\alpha^{p^b-1} - 1 = p(\alpha) \cdot q(\alpha) = 0 \cdot q(\alpha) = 0, \tag{2.6}$$

resultando em

$$\alpha^{p^b-1} = 1. \tag{2.7}$$

Assim, a partir de $\alpha^{i(p^b-1)}$, sendo $i = 1, 2, \ldots$, as potências passam a se repetirem. Portanto, a sequência contém p^b **elementos distintos**, incluindo o elemento neutro 0, ou seja,

$$\mathbb{F}_q = \mathbb{F}_{p^b} = \{0, 1, \alpha, \ldots, \alpha^{b-2}\}. \tag{2.8}$$

Essas raízes podem ser expressas tanto na forma polinomial quanto por uma representação vetorial, onde os coeficientes são alocados em um vetor conforme o grau de suas potências.

Exemplo 2.1.6 *Seja $\mathbb{F}_2 = \{0, 1\}$ um corpo finito. Deseja-se construir a extensão de Galois $\mathbb{F}_{2^4}$. Dado que $b = 4$ e considerando $p(x) = x^4 + x + 1$ um polinômio primitivo de grau 4 em $\mathbb{F}_2[x]$, mostrado na Tabela 5. Se α é uma raiz primitiva de $p(x)$, tem-se que*

$$p(\alpha) = \alpha^4 + \alpha + 1 = 0 \implies \alpha^4 = \alpha + 1.$$

Sabendo então que $\alpha^4 = \alpha + 1$ e usando (2.7) tem-se

$$\alpha^{2^4-1} = 1 \implies \alpha^{15} = 1.$$

Agora, é possível construir $\mathbb{F}_{2^4}$. Por exemplo,

$$\alpha^5 = \alpha \cdot \alpha^4 = \alpha \cdot (\alpha + 1) = \alpha^2 + \alpha,$$

$$\alpha^6 = \alpha \cdot \alpha^5 = \alpha \cdot (\alpha^2 + \alpha) = \alpha^3 + \alpha^2,$$

$$\alpha^7 = \alpha \cdot \alpha^6 = \alpha \cdot (\alpha^3 + \alpha^2) = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1$$

O restante dos elementos de $\mathbb{F}_{2^4}$ são mostrados na Tabela 6.
Para multiplicar dois elementos α^j e α^i, basta conservar a base e somar seus expoentes, usando do fato de que $\alpha^{2^4-1} = \alpha^{15} = 1$. Por exemplo,

$$\alpha^2 \cdot \alpha^3 = \alpha^5 \quad \textit{e} \quad \alpha^8 \cdot \alpha^9 = \alpha^{17} = \alpha^{15} \cdot \alpha^2 = 1 \cdot \alpha^2 = \alpha^2.$$

Para dividir α^j por α^i, basta multiplicar α^j pelo inverso multiplicativo de α^i, ou seja, α^{15-i}, mostrado abaixo.

$$\frac{\alpha^4}{\alpha^{12}} = \alpha^4 \cdot \alpha^{15-12} = \alpha^4 \cdot \alpha^3 = \alpha^7.$$

Para realizar somas, pode-se usar suas representações polinomiais ou representações vetoriais. Por exemplo,

$$\alpha^4 + \alpha^{10} = (\alpha + 1) + (\alpha^2 + \alpha + 1) = \alpha^2,$$

$$1 + \alpha^5 + \alpha^{10} = 1 + (\alpha^2 + \alpha) + (\alpha^2 + \alpha + 1) = 0.$$

Por fim, usando a representação vetorial tem-se,

$$\alpha^4 + \alpha^{10} = 0011 + 0111 = 0100 = \alpha^2,$$

Tabela 6 – Representação dos elementos de $\mathbb{F}_{2^4}$ gerados por $p(x) = x^4 + x + 1$.

Representação por Potência	**Representação Polinomial**	**Representação Vetorial**
0	0	0 0 0 0
1	1	0 0 0 1
α	α	0 0 1 0
α^2	α^2	0 1 0 0
α^3	α^3	1 0 0 0
α^4	$\alpha + 1$	0 0 1 1
α^5	$\alpha^2 + \alpha$	0 1 1 0
α^6	$\alpha^3 + \alpha^2$	1 1 0 0
α^7	$\alpha^3 + \alpha + 1$	1 0 1 1
α^8	$\alpha^2 + 1$	0 1 0 1
α^9	$\alpha^3 + \alpha$	1 0 1 0
α^{10}	$\alpha^2 + \alpha + 1$	0 1 1 1
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1 1 1 0
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1 1 1 1
α^{13}	$\alpha^3 + \alpha^2 + 1$	1 1 0 1
α^{14}	$\alpha^3 + 1$	1 0 0 1

Fonte: Produção da Própria Autora.

$$1 + \alpha^5 + \alpha^{10} = 0001 + 0110 + 0111 = 0000 = 0.$$

A seguir apresenta-se a relação entre a operação lógica XOR e a adição no corpo finito $\mathbb{F}_2$.

Definição 2.1.16 *No corpo finito $\mathbb{F}_2 = \{0, 1\}$, a operação XOR é equivalente à* ***soma módulo 2*** *definida por*

$$a \oplus b = (a + b) \quad (\bmod 2), \tag{2.9}$$

sendo

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 0.$$

Dessa forma, o XOR pode ser interpretado como uma operação linear sobre vetores binários em $\mathbb{F}_2$.

Visto que todo corpo finito tem um polinômio primitivo $p(x)$ que é usado para gerar o corpo, a multiplicação neste corpo pode ser representada por

$$A \otimes B \equiv (A \cdot B) \quad (\bmod p(x)). \tag{2.10}$$

Exemplo 2.1.7 *Sejam dois polinômios* $A(x) = x^3 + x + 1$ *e* $B(x) = x^2 + 1 \in \mathbb{F}_{2^4}$. *Realizando a multiplicação polinomial, obtem-se*

$$A(x) \cdot B(x) = (x^3 + x + 1) \cdot (x^2 + 1) = x^5 + x^2 + x + 1.$$

Considerando o polinômio irredutível $p(x) = x^4 + x + 1$, *realiza-se a redução módulo* $p(x)$*:*

$$x^5 + x^2 + x + 1 \quad (\bmod\ x^4 + x + 1\ ).$$

Substituindo $x^4 = x + 1$, *tem-se*

$$x^5 = x \cdot (x^4) = x \cdot (x + 1) = x^2 + x.$$

Logo,

$$A \otimes B = (x^2 + x) + x^2 + x + 1 = 1.$$

Por fim, a divisão com corpos de Galois é a multiplicação pelo inverso multiplicativo do divisor

$$A \oslash B = A \otimes B^{-1}. \tag{2.11}$$

Entretanto, encontrar o inverso multiplicativo em um corpo finito não é trivial. Como alternativa, ela pode ser calculada usando logaritmos e exponenciais a partir das Tabelas 7 e 8, respectivamente, e dada por (2.12).

$$A \oslash B = \exp\left[(\log(A) - \log(B)) \quad (\bmod\ 0xFF\ )\right], \tag{2.12}$$

onde $0xFF$ é a representação em hexadecimal da sequência binária.

Tabela 7 – Logaritmo de GF(2^8).

log	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	$X^{(1)}$	0	1	19	2	32	1A	C6	3	DF	33	EE	1B	68	C7	4B
1	4	64	E0	0E	34	8D	EF	81	1C	C1	69	F8	C8	8	4C	71
2	5	8A	65	2F	E1	24	0F	21	35	93	8E	DA	F0	12	82	45
3	1D	B5	C2	7D	6A	27	F9	B9	C9	9A	9	78	4D	E4	72	A6
4	6	BF	8B	2	66	DD	30	FD	E2	98	25	B3	10	91	22	88
5	36	D0	94	CE	8F	96	DB	BD	F1	D2	13	5C	83	38	46	40
6	1E	42	B6	A3	C3	48	7E	6E	6B	3A	28	54	FA	85	BA	3D
7	CA	5E	9B	9F	0A	15	79	2B	4E	D4	E5	AC	73	F3	A7	57
8	7	70	C0	F7	8C	80	63	0D	67	4A	DE	ED	31	C5	FE	18
9	E3	A5	99	77	26	B8	B4	7C	11	44	92	D9	23	20	89	2E
A	37	3F	D1	5B	95	BC	CF	CD	90	87	97	B2	DC	FC	BE	61
B	F2	56	D3	AB	14	2A	5D	9E	84	3C	39	53	47	6D	41	A2
C	1F	2D	43	D8	B7	7B	A4	76	C4	17	49	EC	7F	0C	6F	F6
D	6C	A1	3B	52	29	9D	55	AA	FB	60	86	B1	BB	CC	3E	5A
E	CB	59	5F	B0	9C	A9	A0	51	0B	F5	16	EB	7A	75	2C	D7
F	4F	AE	D5	E9	E6	E7	AD	E8	74	D6	F4	EA	A8	50	58	AF

Fonte: Retirado de (JACOB, 2009).

Tabela 8 – Exponencial de GF(2^8).

exp	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	4	8	10	20	40	80	1D	3A	74	E8	CD	87	13	26
1	4C	98	2D	5A	B4	75	EA	C9	8F	3	6	0C	18	30	60	C0
2	9D	27	4E	9C	25	4A	94	35	6A	D4	B5	77	EE	C1	9F	23
3	46	8C	5	0A	14	28	50	A0	5D	BA	69	D2	B9	6F	DE	A1
4	5F	BE	61	C2	99	2F	5E	BC	65	CA	89	0F	1E	3C	78	F0
5	FD	E7	D3	BB	6B	D6	B1	7F	FE	E1	DF	A3	5B	B6	71	E2
6	D9	AF	43	86	11	22	44	88	0D	1A	34	68	D0	BD	67	CE
7	81	1F	3E	7C	F8	ED	C7	93	3B	76	EC	C5	97	33	66	CC
8	85	17	2E	5C	B8	6D	DA	A9	4F	9E	21	42	84	15	2A	54
9	A8	4D	9A	29	52	A4	55	AA	49	92	39	72	E4	D5	B7	73
A	E6	D1	BF	63	C6	91	3F	7E	FC	E5	D7	B3	7B	F6	F1	FF
B	E3	DB	AB	4B	96	31	62	C4	95	37	6E	DC	A5	57	AE	41
C	82	19	32	64	C8	8D	4	0E	1C	38	70	E0	DD	A7	53	A6
D	51	A2	59	B2	79	F2	F9	EF	C3	9B	2B	56	AC	45	8A	9
E	12	24	48	90	3D	7A	F4	F5	F7	F3	FB	EB	CB	8B	0B	16
F	2C	58	B0	7D	FA	E9	CF	83	1B	36	6C	D8	AD	47	8E	$X^{(1)}$

Fonte: Retirado de (JACOB, 2009).

2.2 LOGARITMO DE ZECH

Outro conceito que será utilizado neste trabalho é o logaritmo de Zech, cuja definição será apresentada a seguir. Apresentaremos alguns conceitos e propriedades que serão necessários no Capítulo 5. Uma análise mais detalhada sobre o tema pode ser encontrada em (HUBER, 1990).

Definição 2.2.1 *Seja α um elemento primitivo de um corpo finito $\mathbb{F}_q$. Para $\mathbb{F}_q = \{0, 1, \ldots, q-2\} \cup \{-\infty\}$, o **logaritmo de Zech** de n em relação à base α é definido pela equação*

$$Z(n) = \log_\alpha(1 + \alpha^n) \Longleftrightarrow \alpha^{Z(n)} = 1 + \alpha^n. \tag{2.13}$$

A adição de potências de um elemento primitivo α pode ser realizada da mesma forma apresentada na Subseção 2.1.2, ou pode-se utilizar o logaritmo de Zech, como forma alternativa. Sejam α^x e α^y elementos primitivos de $\mathbb{F}_q$, se z é o expoente da soma $\alpha^x + \alpha^y$, isto é, $\alpha^x + \alpha^y = \alpha^z$, então

$$z = x + Z(y - x) = y + Z(x - y). \tag{2.14}$$

A seguir, são apresentadas algumas propriedades do logaritmo de Zech em um corpo finito $\mathbb{F}_q$ e p um número primo.

1. $Z(0) = -\infty$, para $p = 2$;
2. $Z\left(\frac{q-1}{2}\right) = -\infty$, para $p \neq 2$;
3. $Z(px) = p \cdot Z(x) \pmod{q-1}$;
4. $Z(q - 1 - x) = Z(x) - x \pmod{q-1}, \quad x \neq -\infty$;
5. $Z(-\infty) = 0$, para todos os primos.

Exemplo 2.2.1 *Seja $\alpha \in \mathbb{F}_{2^3}$ uma raiz do polinômio primitivo $p(x) = x^3 + x + 1$. Tem-se que $\alpha^3 + \alpha + 1 = 0$ e seus elementos são apresentados na Tabela 9.*

Tabela 9 – Representação dos elementos de $\mathbb{F}_{2^3}$ gerados por $p(x) = x^3 + x + 1$.

Representação por Potência	**Representação Polinomial**
0	0
1	1
α	α
α^2	α^2
α^3	$\alpha + 1$
α^4	$\alpha^2 + \alpha$
α^5	$\alpha^2 + \alpha + 1$
α^6	$\alpha^2 + 1$
α^7	1

Fonte: Produção da Própria Autora.

A partir da Equação 2.13, pode-se calcular os logaritmos de Zech, cujos alguns valores são apresentados na Tabela 10.

Tabela 10 – Logaritmos de Zech para $p(x) = x^3 + x + 1$.

Logaritmo de Zech
$Z(0) = -\infty$
$Z(1) = 3$
$Z(2) = 6$
$Z(3) = 1$
$Z(4) = 5$
$Z(5) = 4$
$Z(6) = 2$

Fonte: Produção da Própria Autora.

Exemplificando os cálculos para $n = 5$ e $n = -4$, tem-se

$$\alpha^{Z(5)} = 1 + \alpha^5 = 1 + \alpha^2 + \alpha + 1 = \alpha^2 + \alpha = \alpha^4 \implies Z(5) = 4$$

e

$$\alpha^{Z(-4)} = 1 + \alpha^{(7-4)} = 1 + \alpha^3 = 1 + \alpha + 1 = \alpha \implies Z(-4) = 1.$$

2.3 MATRIZES SUPERREGULARES

Nesta seção é apresentado o conceito de matrizes superregulares, em especial das matrizes de Vandermonde e Cauchy, as quais desempenham grande importância na construção dos códigos matriciais MDS que serão apresentados neste trabalho. Os conceitos apresentados nesta seção podem ser encontrados em (CARDELL; CLIMENT; REQUENA, 2013), (HUFFMAN; PLESS, 2003) e (FIEDLER, 2010).

Definição 2.3.1 *Uma matriz A com elementos em um corpo finito $\mathbb{F}_q$ é chamada de matriz superregular se toda submatriz de A não for singular, ou seja, possuir todos os subdeterminantes diferentes de zero.*

Podemos também definir um tipo especial de matrizes superregulares chamado matrizes superregulares por blocos b.

Definição 2.3.2 *Uma matriz $A \in M_{mb\times nb}(\mathbb{F}_q)$ é dita uma **matriz superregular por blocos** b se toda submatriz de A e também a submatriz A composta por blocos de tamanho $b \times b$ forem não singulares sobre $\mathbb{F}_q$.*

Exemplo 2.3.1 *Seja o polinômio primitivo* $p(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$ *e seja* α *uma raiz de* $p(x)$ *gerando* $\mathbb{F}_{2^3}$. *Ao considerar a matriz*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 \end{bmatrix},$$

pode-se verificar a superregularidade calculando os determinantes 2×2

$$\begin{vmatrix} 1 & 1 \\ 1 & \alpha \end{vmatrix} = \alpha - 1 = \alpha^3; \qquad \begin{vmatrix} 1 & 1 \\ \alpha & \alpha^2 \end{vmatrix} = \alpha^2 - \alpha = \alpha^4;$$

$$\begin{vmatrix} 1 & \alpha \\ 1 & \alpha^2 \end{vmatrix} = \alpha^2 - \alpha = \alpha^4; \qquad \begin{vmatrix} \alpha & \alpha^2 \\ \alpha^2 & \alpha^4 \end{vmatrix} = \alpha^5 - \alpha^4 = 1;$$

$$\begin{vmatrix} 1 & \alpha^2 \\ 1 & \alpha^3 \end{vmatrix} = \alpha^3 - \alpha^2 = \alpha^5; \qquad \begin{vmatrix} \alpha^2 & \alpha^4 \\ \alpha^3 & \alpha^6 \end{vmatrix} = \alpha^8 - \alpha^7 = \alpha^3;$$

$$\begin{vmatrix} 1 & 1 \\ \alpha^2 & \alpha^3 \end{vmatrix} = \alpha^3 - \alpha^2 = \alpha^5; \qquad \begin{vmatrix} \alpha^2 & \alpha^3 \\ \alpha^4 & \alpha^6 \end{vmatrix} = \alpha^8 - \alpha^7 = \alpha^3;$$

$$\begin{vmatrix} \alpha^4 & \alpha^6 \\ \alpha^6 & \alpha^2 \end{vmatrix} = \alpha^6 - \alpha^{12} = \alpha.$$

Por fim, observa-se que os determinantes das submatrizes de ordem 3 e 4 são não nulos, completando a verificação da superregularidade.

Através do Exemplo 2.3.1 observa-se que quando aumenta a dimensão, aumenta o grau de dificuldade de verificação da superregularidade, pois a quantidade de subdeterminantes a serem calculados aumentam. A seguir é apresentado a construção de duas famílias de matrizes superregulares.

Definição 2.3.3 *Uma matriz A é dita matriz de Vandermonde se todas as suas linhas estiverem em progressão geométrica.*

Sejam $\alpha_1, ..., \alpha_{n-k}$ elementos não nulos de um corpo e sendo A uma matriz de dimensões $k \times (n-k)$, a forma geral de uma matriz de Vandermonde é definida por

$$A = \begin{bmatrix} 1 & 1 & ... & 1 \\ \alpha_1 & \alpha_2 & ... & \alpha_{n-k} \\ \alpha_1^2 & \alpha_2^2 & ... & \alpha_{n-k}^2 \\ \alpha_1^3 & \alpha_2^3 & ... & \alpha_{n-k}^3 \\ \vdots & \vdots & & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & ... & \alpha_{n-k}^{k-1} \end{bmatrix}, \tag{2.15}$$

A transposta de uma matriz de Vandermonde também é uma matriz de Vandermonde.

Lema 2.3.1 *Se A é uma matriz de Vandermonde, sua determinante é dada por*

$$|A| = \prod_{1 \leq i < j \leq n-k} (\alpha_j - \alpha_i), \tag{2.16}$$

com i e j representando as linhas e as colunas da matriz, respectivamente. Em particular, A será não singular se os elementos $\alpha_1, ..., \alpha_{n-k}$ forem distintos.

Exemplo 2.3.2 *Sabendo que α, α^2, α^3, α^4 e α^5 são elementos pertencentes ao corpo $\mathbb{F}_{2^5}$ e considerando tais elementos em progressão geométrica, a matriz de Vandermonde é dada como*

$$A = \begin{bmatrix} (\alpha)^0 & (\alpha^2)^0 & (\alpha^3)^0 & (\alpha^4)^0 & (\alpha^5)^0 \\ (\alpha)^1 & (\alpha^2)^1 & (\alpha^3)^1 & (\alpha^4)^1 & (\alpha^5)^1 \\ (\alpha)^2 & (\alpha^2)^2 & (\alpha^3)^2 & (\alpha^4)^2 & (\alpha^5)^2 \\ (\alpha)^3 & (\alpha^2)^3 & (\alpha^3)^3 & (\alpha^4)^3 & (\alpha^5)^3 \\ (\alpha)^4 & (\alpha^2)^4 & (\alpha^3)^4 & (\alpha^4)^4 & (\alpha^5)^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} \\ \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} \end{bmatrix}.$$

Definição 2.3.4 *Uma matriz A de dimensões $m \times n$ é chamada **matriz de Cauchy** se, e somente se,*

$$A_{ij} = \frac{1}{x_i + y_j}, \quad \textit{para } 1 \leq i \leq m \textit{ e } 1 \leq j \leq n, \tag{2.17}$$

sendo $x_i + y_j \neq 0$ e x_i e y_j elementos pertencentes ao corpo $\mathbb{F}_q$ e distintos.

Propriedade 2.3.1 *Seja A uma matriz de dimensões $m \times n$ com entradas com elementos diferentes de zero de um corpo $\mathbb{F}_q$. São equivalentes:*

1. *A é uma matriz de Cauchy.*
2. *Toda submatriz 2×2 de A é não singular.*

Exemplo 2.3.3 *Seja $\alpha \in \mathbb{F}_{2^3}$ um elemento primitivo obtido de $p(x) = x^3 + x + 1$, apresentado na Tabela 5, tal que $\alpha^3 + \alpha + 1 = 0$, e se $x_1 = 0$, $x_2 = 1$, $x_3 = \alpha$, $y_1 = \alpha^2$, $y_2 = \alpha^3$ e $y_3 = \alpha^4$ é possível verificar que*

$$A = \begin{bmatrix} \alpha^5 & \alpha^4 & \alpha^3 \\ \alpha & \alpha^6 & \alpha^2 \\ \alpha^3 & 1 & \alpha^5 \end{bmatrix}$$

é uma matriz superregular de Cauchy sobre $\mathbb{F}_{2^3}$. Observa-se que para este caso, a realização do cálculo das potências do elemento primitivo α foram baseados nos apresentados na Seção 2.1.2 juntamente com o auxílio da Tabela 9:

$$a_{11} = \frac{1}{x_1 + y_1} = \frac{1}{0 + \alpha^2} = \frac{1}{\alpha^2} = 1 \cdot \alpha^{7-2} = \alpha^5;$$

$$a_{12} = \frac{1}{x_1 + y_2} = \frac{1}{0 + \alpha^3} = \frac{1}{\alpha^3} = 1 \cdot \alpha^{7-3} = \alpha^4;$$

$$a_{13} = \frac{1}{x_1 + y_3} = \frac{1}{0 + \alpha^4} = \frac{1}{\alpha^4} = 1 \cdot \alpha^{7-4} = \alpha^3;$$

$$a_{21} = \frac{1}{x_2 + y_1} = \frac{1}{1 + \alpha^2} = \frac{1}{\alpha^6} = 1 \cdot \alpha^{7-6} = \alpha;$$

$$a_{22} = \frac{1}{x_2 + y_2} = \frac{1}{1 + \alpha^3} = \frac{1}{\alpha} = 1 \cdot \alpha^{7-1} = \alpha^6;$$

$$a_{23} = \frac{1}{x_2 + y_3} = \frac{1}{1 + \alpha^4} = \frac{1}{\alpha^5} = 1 \cdot \alpha^{7-5} = \alpha^2;$$

$$a_{31} = \frac{1}{x_3 + y_1} = \frac{1}{\alpha + \alpha^2} = \frac{1}{\alpha^4} = 1 \cdot \alpha^{7-4} = \alpha^3;$$

$$a_{32} = \frac{1}{x_3 + y_2} = \frac{1}{\alpha + \alpha^3} = \frac{1}{1} = 1;$$

$$a_{33} = \frac{1}{x_3 + y_3} = \frac{1}{\alpha + \alpha^4} = \frac{1}{\alpha^2} = 1 \cdot \alpha^{7-2} = \alpha^5.$$

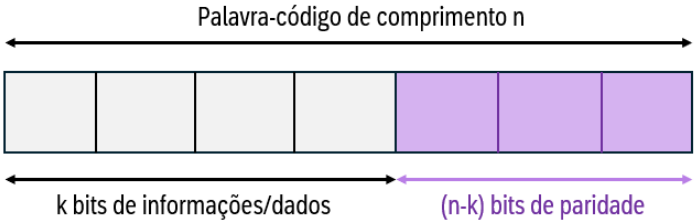
3 CÓDIGOS CORRETORES DE ERROS

Neste capítulo é realizado um estudo sobre códigos corretores de erros. Um código corretor de erros consiste em um método sistemático de adicionar informações, conhecidas como redundância ou paridade, aos dados transmitidos ou armazenados, possibilitando a detecção e correção de falhas durante a transmissão. Neste trabalho iremos considerar que a saída de uma fonte de informação seja uma sequência de símbolos sobre o corpo finito $\mathbb{F}_2 = \{0, 1\}$, denominada como sequência de informação. Os símbolos binários dessa sequência são comumente chamados de bits de informação, onde o bit representa um dígito binário "0"ou "1". Na Seção 3.1, são introduzidos os códigos de bloco lineares e, na Seção 3.3, são abordados os códigos de máxima distância de separação (MDS). Em seguida, a Seção 3.4 apresenta os códigos matriciais lineares, que servirão de base para a construção dos códigos matriciais MDS no Capítulo 5. As principais referências utilizadas neste capítulo foram (HEFEZ; VILLELA, 2008), (ROTH, 2006), (RAYN; LIN, 2009) e (GUIMARÃES, 2003).

3.1 CÓDIGOS DE BLOCOS LINEARES

Na codificação por blocos, cada mensagem de entrada $u = [u_1, u_2, \ldots, u_k]$, composta por k bits de informação, é transformada em uma sequência $v = [v_1, \ldots, v_n]$, de n bits, com $n > k$, de acordo com determinadas regras de codificação. Essa sequência expandida v é denominada como palavra-código correspondente à mensagem u, exemplificada na Figura 3.

Figura 3 – Estrutura de uma palavra-código.



Fonte: Elaborado pela autora.

Os $n-k$ bits adicionais inseridos pelo codificador são chamados de bits redundantes ou de paridade. Esses bits não carregam nova informação, mas têm a função essencial de permitir que o código detecte e corrija erros de transmissão provocados por ruído ou interferência no canal. Um dos principais desafios no projeto de um codificador é determinar como gerar esses bits redundantes de forma que o código apresente uma boa capacidade de correção de erros. Uma das estruturas mais desejáveis para isso é a linearidade.

Definição 3.1.1 *Um código em bloco binário de comprimento n com q^k palavras-código é chamado de código de bloco linear $\mathcal{C}$ $[n, k]$, com n sendo seu comprimento e k a sua dimensão, se, e somente se, suas q^k palavras-código formarem um subespaço vetorial k-dimensional sobre o espaço vetorial $\mathbb{F}_q^n$.*

Como um código linear binário $\mathcal{C}$ $[n, k]$ é um subespaço k-dimensional do espaço vetorial das n-uplas sobre $\mathbb{F}_q$, existem k palavras-código linearmente independentes $g_1, g_2, \ldots, g_k$ em $\mathcal{C}$, tais que qualquer palavra-código $c \in \mathcal{C}$ pode ser expressa como uma combinação linear delas. Essas palavras formam uma base β de $\mathcal{C}$.

Seja $u = [u_1, u_2, \ldots, u_k]$ a mensagem a ser codificada. A palavra-código correspondente é obtida pela combinação linear:

$$c = u_1 g_1 + u_2 g_2 + \cdots + u_k g_k. \tag{3.1}$$

Podemos organizar as k palavras-código linearmente independentes como as linhas de uma matriz $k \times n$ sobre $\mathbb{F}_q$:

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}. \tag{3.2}$$

A matriz G é chamada de **matriz geradora** do código linear $\mathcal{C}$ $[n, k]$. A escolha de G não é única, pois depende da base escolhida para $\mathcal{C}$, mas sua dimensão é sempre igual a k. Assim, a codificação da mensagem u pode ser expressa como

$$c = u \cdot G. \tag{3.3}$$

Exemplo 3.1.1 *Seja $\mathcal{C}$ um código de bloco linear em $\mathbb{F}_2^3$ sobre $\mathbb{F}_2$ com parâmetros $[3, 2]$. Este código possui comprimento $n = 3$ e dimensão $k = 2$, pois existem $2^2 = 4$ palavras-código. Escolhendo como base $\beta = \{011,\ 101\}$, temos que*

$$G = \begin{bmatrix} 011 \\ 101 \end{bmatrix}$$

é a matriz geradora de $\mathcal{C}$. Assim,

$$\mathcal{C} = \{a_1 \cdot [011] + a_2 \cdot [101] \mid a_1, a_2 \in \mathbb{F}_2\} = \{000,\ 011,\ 101,\ 110\}.$$

A métrica que será utilizada neste trabalho é a distância de Hamming que será definida a seguir.

Definição 3.1.2 *Dados $u = [u_1, \ldots, u_n]$ e $c = [c_1, \ldots, c_n] \in \mathbb{F}_q^n$, a distância de Hamming é definida $d(u, c)$ entre u e c como o número de símbolos em que se diferem.*

Definição 3.1.3 *Seja $\mathcal{C}$ um código linear. A **distância mínima do código** é definida como*

$$d = \min\{d(u, c);\ u, c \in \mathcal{C}, u \neq c\}. \tag{3.4}$$

Exemplo 3.1.2 *Seja o código de bloco linear* $C = \{00000,\ 01101,\ 10110,\ 11011\}$ *sobre* $\mathbb{F}_2^5$. *Assim,*

$$\begin{aligned} d(00000, 01101) &= 3, \\ d(00000, 10110) &= 3, \\ d(00000, 11011) &= 4, \\ d(01101, 10110) &= 4, \\ d(01101, 11011) &= 3, \\ d(10110, 11011) &= 3. \end{aligned}$$

Logo, a distância mínima deste código é $d = 3$.

Propriedade 3.1.1 *Dados três vetores* $u, c, w \in \mathcal{C}$, *valem as seguintes propriedades:*

1. *Positividade:* $d(u, c) \geq 0$, *e a igualdade vale se, e somente se,* $u = c$.
2. *Simetria:* $d(u, c) = d(c, u)$.
3. *Desigualdade Triangular:* $d(u, c) \leq d(u, w) + d(w, c)$.

Definição 3.1.4 ***O peso*** $w(v)$ *de um vetor* v *é o número de coordenadas não nulas de* v. ***O peso mínimo*** $w(\mathcal{C})$ *de um código* $\mathcal{C}$ *é o menor peso entre suas palavras-código não nulas.*

Exemplo 3.1.3 *No código do exemplo anterior* $\mathcal{C} = \{00000,\ 01101,\ 10110,\ 11011\}$, *tem-se*

$$\begin{aligned} w(01101) &= 3, \\ w(10110) &= 3, \\ w(11011) &= 4. \end{aligned}$$

Logo, o peso mínimo deste código é $w(\mathcal{C}) = 3$.

Teorema 3.1.1 *Seja* $\mathcal{C}$ *um código linear e* $u,\ c \in \mathcal{C}$, *pode-se afirmar que*

1. $d(u, c) = w(u - c)$;
2. $d(\mathcal{C}) = w(\mathcal{C})$.

Dada a distância mínima d de um código linear, os parâmetros de $\mathcal{C}$ são $[n, k, d]$, onde n é o comprimento do código e k é a dimensão.

Teorema 3.1.2 *Um código de bloco linear* $\mathcal{C}$ *com parâmetros* $[n, k, d]$ *possui capacidade de detectar* $(d - 1)$ *erros e corrigir até* $\left\lfloor \dfrac{d-1}{2} \right\rfloor$ *erros.*

Exemplo 3.1.4 *No código* $C = \{00000,\ 01101,\ 10110,\ 11011\}$, *tem-se* $d(\mathcal{C}) = w(\mathcal{C}) = 3$. *Logo, os parâmetros deste código são* $[5, 2, 3]$ *e ele é capaz de detectar* $d - 1 = 2$ *erros e corrigir* $\left\lfloor \dfrac{d-1}{2} \right\rfloor = 1$ *erro.*

Definição 3.1.5 *Uma matriz geradora G de um código linear $\mathcal{C}$ está na forma padrão, ou sistemática, se ela estiver escrita como*

$$G = [I_k \mid P], \tag{3.5}$$

em que I_k é a matriz identidade de ordem k e P é uma matriz de dimensão $k \times (n-k)$.

Teorema 3.1.3 *Para todo código linear $\mathcal{C}$, existe um código $\mathcal{C}'$ equivalente cuja matriz geradora está escrita na forma padrão, possuindo os mesmos parâmetros de $\mathcal{C}$.*

Quando a codificação é realizada utilizando uma matriz geradora na forma sistemática (3.5), torna-se possível identificar diretamente a mensagem original, uma vez que ela aparece nas primeiras posições da palavra-código, enquanto as últimas posições representam os bits de paridade adicionados durante o processo de codificação.

Exemplo 3.1.5 *Considere uma matriz geradora G de um código linear $\mathcal{C}$ sobre o corpo finito $\mathbb{F}_2^5$, dada por*

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

A matriz

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

é a matriz geradora de um código linear equivalente a $\mathcal{C}$, com os mesmos parâmetros, mas agora na forma padrão. Se, por exemplo, for recebida a palavra-código $[10101]$, observa-se que a mensagem original corresponde aos três primeiros bits $[101]$, enquanto os dois últimos bits $[01]$ representam a redundância adicionada pelo processo de codificação.

A partir de uma matriz geradora na forma padrão, é possível determinar a matriz verificação de paridade, a qual desempenha papel essencial no processo de decodificação e detecção de erros.

Definição 3.1.6 *Seja $\mathcal{C}$ um código linear com parâmetros $[n, k]$. Se $G = [I_k \mid P]$ é a matriz geradora de $\mathcal{C}$ escrita na forma sistemática, então a matriz de verificação de paridade H é dada por*

$$H = [-P^t \mid I_{n-k}], \tag{3.6}$$

onde P^t representa a transposta da matriz P.

A matriz H é utilizada para verificar se uma palavra pertence ou não ao código, a partir de

$$G \cdot H^t = [I_k \mid P] \cdot [-P^t \mid I_{n-k}] = -P + P = \mathbf{0}, \tag{3.7}$$

segue que

$$c \cdot H^t = \mathbf{0}, \tag{3.8}$$

para todo $c \in \mathcal{C}$.

As linhas da matriz de verificação de paridade H são linearmente independentes, assim como as linhas da matriz geradora G. Portanto, H também é uma matriz geradora de outro código linear, denominado **código de** $\mathcal{C}$, indicado por $\mathcal{C}^{\perp}$.

Exemplo 3.1.6 *Considerando o Exemplo 3.1.5, a matriz geradora G é dada por*

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

Observando que a matriz geradora deste código já está na forma padrão, como definido em (3.5), segue que a matriz de verificação de paridade é dada por

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

Supondo que os vetores $c_1 = [10101]$ e $c_2 = [10111]$ foram recebidos, para verificar se a mensagem é uma palavra código de $\mathcal{C}$, tem-se

$$c_1 \cdot H^t = [10101] \cdot \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = [00],$$

$$c_2 \cdot H^t = [10111] \cdot \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = [10].$$

Logo, conclui-se que $c_1 \in \mathcal{C}$, pois o resultado é o vetor nulo, enquanto $c_2 \notin \mathcal{C}$.

A decodificação está diretamente relacionada aos processos de detecção e correção de erros em sistemas de comunicação digital. Durante a transmissão de dados, ruídos ou interferências podem alterar bits da mensagem original, resultando em erros no sinal recebido (RAYN; LIN, 2009).

Definição 3.1.7 *Seja $\mathcal{C}$ um código linear com parâmetros $[n, k, d]$ e matriz de verificação de paridade H. Considere um vetor $c \in \mathcal{C}$ transmitido através de um canal ruidoso, resultando no vetor recebido v. O vetor erro, que representa as alterações ocorridas durante a transmissão, é definido por*

$$e = v - c. \tag{3.9}$$

Exemplo 3.1.7 *Considere um código linear $\mathcal{C}$ sobre $\mathbb{F}_2$. Se for transmitido o vetor $c = [011001]$ e recebido $v = [101010]$, o erro ocorrido é dado por*

$$e = v - c = [101010] - [011001] = [110011].$$

A quantidade de posições não nulas, ou seja, o peso do vetor erro, indica o número de bits que foram alterados na transmissão. Seja H a matriz de verificação de paridade associada ao código $\mathcal{C}$, tem-se

$$e \cdot H^t = (v - c) \cdot H^t = v \cdot H^t - c \cdot H^t = v \cdot H^t, \tag{3.10}$$

pois de acordo com (3.8), $c \cdot H^t = 0$.

Definição 3.1.8 *É possível verificar se tal informação recebida pertence ao código através do produto entre v e H^t, o qual denomina-se síndrome*

$$s = v \cdot H^t. \tag{3.11}$$

Caso $s = \mathbf{0}$, conclui-se que não houve erro, e portanto $v \in \mathcal{C}$. Se $s \neq \mathbf{0}$, ocorreu erro na transmissão. Para corrigir o vetor recebido, é necessário identificar a posição do erro.

Seja $\mathcal{C}$ um código linear de distância mínima $d \geq 3$, para corrigir no mínimo 1 erro, e o vetor e introduzido entre a palavra transmitida c e recedida v. Se

$$e \cdot H^t = \mathbf{0}, \tag{3.12}$$

então a mensagem recebida pertence ao código, ou seja, $c \in \mathcal{C}$. Se

$$e \cdot H^t \neq \mathbf{0}, \tag{3.13}$$

e considerando que apenas um erro tenha ocorrido, o vetor erro será

$$e = [0, \ldots, 1, \ldots, 0], \tag{3.14}$$

onde 1 está na i-ésima posição. Assim,

$$e \cdot H^t = h_i, \tag{3.15}$$

sendo h_i a i-ésima coluna da matriz H. Para o vetor recebido v, obtém-se

$$v \cdot H^t = h_i. \tag{3.16}$$

Portanto, a síndrome resultante indica a posição do erro, permitindo sua correção e garantindo que o vetor decodificado corresponda à palavra original transmitida.

Exemplo 3.1.8 *Considere um código linear binário com parâmetros $[6, 3]$ e a matriz de verificação*

de paridade dado por

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

Supondo que o vetor recebido foi $v = [000110]$*, tem-se*

$$s = v \cdot H^T = [000110] \cdot \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = [011] = h_2.$$

Portanto, $e = [011000]$ *e a correção é feita por*

$$c = v - e = [000110] - [011000] = [011110].$$

Outra abordagem possível para a decodificação é o método baseado em **classes laterais** ou **arranjo padrão**. Essa técnica aplica o princípio da máxima verossimilhança, ou seja, busca-se corrigir o vetor recebido v para a palavra-código c cuja distância é a menor possível. Assim, as palavras consideradas mais próximas entre si pertencem a uma mesma classe lateral.

Definição 3.1.9 *Seja* $\mathcal{C}$ *um código linear* $[n, k, d]$*. Para qualquer vetor* $v \subset F_q^n$*, define-se o conjunto*

$$v + \mathcal{C} = \{v + c \mid c \in \mathcal{C}\},$$

como sendo a ***classe lateral*** *de* v*.*

Exemplo 3.1.9 *Considere o subespaço linear* $\mathcal{C}$ *de* $\mathbb{F}_2^7$ *gerado pela matriz*

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

O código $\mathcal{C}$ *consiste em todas as combinações lineares das linhas de* G*. Assim, o código é dado por*

$$C = \{0000000, 1001001, 0101011, 0011101, 1100010, 1010100, 0110110, 1111111\}.$$

Tomando o vetor recebido $v = [1110000] \in \mathbb{F}_2^7$*. A classe lateral* $v + C$ *é obtida somando* v *a cada palavra de* C*, obtendo*

$$v + C = \{1110000, 0111001, 1011011, 1101101, 0010010, 0100100, 1000110, 0001111\}.$$

O interesse nas classes laterais está no fato de que todos os vetores pertencentes a uma mesma classe possuem a mesma síndrome. Assim, o vetor v é denominado líder da classe lateral.

Exemplo 3.1.10 *Seja a matriz G geradora e a matriz H controle de paridade de um código de bloco linear $\mathcal{C} \subset \mathbb{F}_2^7$*

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

As síndromes para $v + \mathcal{C}$ são

$$[1110000] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [1111], \qquad [0111001] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [1111],$$

$$[1011011] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [1111], \qquad [1101101] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [1111].$$

$$[0010010] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [1111], \qquad [0100100] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [1111].$$

$$[1000110]\cdot\begin{bmatrix}1&0&0&1\\1&0&1&1\\1&1&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}=[1111],\qquad [0001111]\cdot\begin{bmatrix}1&0&0&1\\1&0&1&1\\1&1&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}=[1111].$$

A partir dessa propriedade, o processo de decodificação consiste em calcular as síndromes de todos os líderes das classes laterais. Assim, quando uma palavra é recebida, basta determinar sua síndrome e compará-la com uma tabela previamente formada pelas síndromes dos líderes. O número de síndromes e de líderes das classes laterais, são dados por q^{n-k}, para um código $\mathcal{C}\in\mathbb{F}_q^n$ com parâmetros $[n,k,d]$.

Exemplo 3.1.11 *Seja a matriz verificação de paridade de um código de bloco linear* $\mathcal{C}\subset\mathbb{F}_2^7$

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

Para obter a tabela basta calcular a síndrome para cada líder, ou seja

$$[0000001]\cdot\begin{bmatrix}1&0&1\\1&1&0\\1&1&1\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=[011],\qquad [0000010]\cdot\begin{bmatrix}1&0&1\\1&1&0\\1&1&1\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=[010],$$

$$[0000100]\cdot\begin{bmatrix}1&0&1\\1&1&0\\1&1&1\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=[100],\qquad [0001000]\cdot\begin{bmatrix}1&0&1\\1&1&0\\1&1&1\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=[011],$$

$$[0010000]\cdot\begin{bmatrix}1&0&1\\1&1&0\\1&1&1\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=[111],\qquad [0100000]\cdot\begin{bmatrix}1&0&1\\1&1&0\\1&1&1\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=[110].$$

$$[1000000] \cdot \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [101].$$

Juntando todas as síndromes, tem-se a Tabela 11.

Tabela 11 – Síndrome de erros da classe lateral do Exemplo 3.1.11.

Líder	**Síndrome**
0000000	000
0000001	001
0000010	010
0000100	100
0001000	011
0010000	111
0100000	110
1000000	101

Fonte: Produção da Própria Autora.

Supondo que foi recebido $v = [1001100]$*, então*

$$s = v \cdot H^T = [1001100] \cdot \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [010].$$

Tomando o erro como sendo o líder da classe com esta síndrome, obtemos $e = [0000010]$*. Portanto, a palavra transmitida corretamente foi*

$$c = v - e = [1001100] - [0000010] = [1001110].$$

3.2 CÓDIGO DE HAMMING

Os códigos de Hamming são amplamente reconhecidos como um exemplo clássico de código linear. Um código de Hamming é descrito pelos parâmetros

$$[n, k, d] = [2^r - 1,\ 2^r - r - 1,\ 3], \tag{3.17}$$

sendo r o número de bits de paridade, n o comprimento total da palavra-código, k o número de bits de informação e $d = 3$ a distância mínima que permite corrigir 1 erro e detectar 2 erros.

Exemplo 3.2.1 *O código de Hamming [7, 4,3] codifica 4 bits em 7 bits através da adição de 3 bits de paridade. Considerando um codificador sistemático, se* $u = [u_1, u_2, u_3, u_4]$ *é uma mensagem a ser codificada por este código então uma palavra-código é dada por*

$$c = [u_1, u_2, u_3, u_4, r_1, r_2, r_3]. \tag{3.18}$$

Os bits de paridade são calculados de acordo com as relações:

- $r_1 = u_1 \oplus u_2 \oplus u_3$*;*
- $r_2 = u_2 \oplus u_3 \oplus u_4$*;*
- $r_3 = u_1 \oplus u_3 \oplus u_4$*.*

Assim, para determinar a palavra-código correspondente à mensagem u = [1010] usando o código de Hamming [7, 4,3], tem-se

$$r_1 = 1 \oplus 0 \oplus 1 = 0;\ \ r_2 = 0 \oplus 1 \oplus 0 = 1;\ \ r_3 = 1 \oplus 1 \oplus 0 = 0.$$

Portanto, a palavra-código é c = [1010010] e a matriz G para o código de Hamming [7,4,3] é dada por

$$G = \begin{array}{c} \begin{matrix} u_1 & u_2 & u_3 & u_4 & r_1 & r_2 & r_3 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}.$$

3.3 CÓDIGOS DE MÁXIMA DISTÂNCIA DE SEPARAÇÃO (MDS)

Nesta seção, são apresentados os códigos de máxima distância de separação (MDS — *Maximum Distance Separable*), os quais possuem a maior distância mínima possível. Essa característica é de grande importância na teoria da codificação, pois a distância mínima está diretamente relacionada à capacidade de correção de erros do código. Assim, códigos com essa propriedade oferecem máxima proteção contra falhas de um dispositivo, para uma determinada quantidade de redundância. Os

conceitos e definições abordados nesta seção podem ser encontrados em (ROMAN, 1992), (ROTH, 2006) e (HUFFMAN; PLESS, 2003).

Seja $\mathcal{C}$ um código linear com parâmetros $[n, k, d]$. Sabe-se que q^k representa o número de palavras-código de $\mathcal{C}$ sobre $\mathbb{F}_q^n$.

Teorema 3.3.1 ***Limitante de Singleton:*** *Seja $\mathcal{C}$ um código linear com parâmetros $[n, k, d]$ sobre um corpo finito $\mathbb{F}_q$, tem-se*

$$q^k \leq q^{n-d+1}. \tag{3.19}$$

A partir da Equação 3.19, obtém-se

$$k \leq n - d + 1, \tag{3.20}$$

que é equivalente,

$$d \leq n - k + 1. \tag{3.21}$$

Definição 3.3.1 *Um código linear $\mathcal{C}$ para o qual a igualdade é satisfeita no Limitante de Singleton é chamado de* ***código de distância máxima separável****, MDS.*

Teorema 3.3.2 *Seja $\mathcal{C}$ um código linear $[n, k]$ sobre $\mathbb{F}_q$. Então, são equivalentes:*

1. *$\mathcal{C}$ é um código MDS.*
2. *Cada conjunto de coordenadas k é um conjunto de informações para $\mathcal{C}$.*
3. *Se uma matriz geradora para $\mathcal{C}$ estiver na forma padrão $G = [I|A]$, então A é uma matriz superregular.*
4. *$\mathcal{C}^{\perp}$ é um código MDS.*
5. *Cada conjunto de coordenadas $n - k$ é um conjunto de informações para $\mathcal{C}^{\perp}$.*

De acordo com a Definição 3.3.1 e o Teorema 3.3.2, nenhum código linear com comprimento $[n, k]$ tem distância mínima maior que um código MDS com os mesmos parâmetros.

A seguir, é mostrado alguns casos em que C é um código MDS trivial sobre $\mathbb{F}_q$:

1. $\mathcal{C} = \mathbb{F}_q^n$ é um código MDS trivial com distância mínima $d = 1$.

   Exemplo 3.3.1 *Considere o espaço vetorial $\mathbb{F}_2^3$. Para $\mathcal{C} = \mathbb{F}_2^3$, os elementos desse código são $\mathcal{C} = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Nesse caso, a distância mínima desse código é $d = 1$.*

2. São códigos MDS triviais e têm a distância mínima $d = n$, os códigos que possuem apenas duas palavras-código, sendo elas a palavra toda nula e a palavra completa com 1's.

Exemplo 3.3.2 *Seja o código de bloco linear $\mathcal{C} \subset \mathbb{F}_2^4$ de parâmetros $[4, 1]$, conhecido como código de repetição. As palavras-código de $\mathcal{C}$ são $\{0000, 1111\}$. Nesse caso, $\mathcal{C}$ é um código MDS trivial, cuja distância mínima é $d = 4$.*

Exemplo 3.3.3 *Os códigos de Hamming formam uma classe de códigos lineares binários, para os quais, dado um inteiro $r \geq 2$, existe um código de comprimento $n = 2^r - 1$ e dimensão $k = 2^r - r - 1$, com distância mínima $d = 3$. Para o caso onde $r = 2$, tem-se um código MDS pois um código com parâmetros $[3, 1, 3]$ as palavras-código de C são $\{000, 111\}$, com distância mínima $d = 3$.*

3. Códigos com um único símbolo de paridade são códigos MDS triviais com distância mínima $d = 2$.

   Exemplo 3.3.4 *Seja $\mathbb{F}_2^4$ com código de fonte dado por $u = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Se*

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

   é uma matriz geradora, então o código $\mathcal{C} = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$ é um código MDS trivial. E, neste caso, tem-se $d = 2$.

Ao analisar a matriz geradora na forma padrão, conforme apresentado no item 3 do Teorema 3.3.2, é possível verificar o seguinte resultado a respeito dos códigos binários:

Teorema 3.3.3 *Seja $\mathcal{C}$ um código binário com parâmetros $[n, k, d]$.*

1. *Se $\mathcal{C}$ é um código MDS, então $\mathcal{C}$ é trivial.*
2. *Se $3 \leq d$ e $5 \leq k$, então $k \leq n - d - 1$.*

3.4 CÓDIGOS MATRICIAIS LINEARES

Nesta seção é apresentado um estudo sobre uma subclasse dos códigos de bloco lineares, denominada **códigos matriciais lineares**. Esses códigos podem ser empregados tanto na correção de erros aleatórios quanto na correção de erros em rajada (*burst of errors*) (FRITSCH, 2025).

Os códigos matriciais destacam-se principalmente por sua simplicidade e flexibilidade. Além disso, por apresentarem decodificação rápida e de fácil implementação, são amplamente aplicados em sistemas de comunicação e sistemas de armazenamento (GUIMARÃES, 2003).

Define-se a seguir o conceito de rajada de erros de comprimento b.

Definição 3.4.1 *Um vetor erro e, dado por (3.22), de comprimento n contém **erro em rajada de comprimento** b se suas coordenadas não nulas estiverem confinadas em b posições consecutivas em e.*

$$e = [0, 0, \ldots, 0, \underbrace{1, *, \ldots, *, 1}_{\textit{rajada } b}, 0, \ldots, 0], \tag{3.22}$$

onde o trecho central representa uma rajada de comprimento b. Se a primeira coordenada diferente de zero da rajada estiver na posição i, diz-se que a rajada b começa na coordenada i e possui padrão de surto $(b_i = 1,\ b_{i+1}, \ldots, b_{i+b-2},\ b_{i+b-1} = 1)$.

Observação 3.4.1 *Nem todas as posições em uma rajada são necessariamente erros.*

Exemplo 3.4.1 *Seja o vetor erro* $e = [000010110000]$*, de comprimento* $n = 12$*. Este vetor contém erro em rajada de comprimento* $b = 4$*.*

Um código é denominado **código corretor em rajada** de comprimento máximo b se ele for capaz de corrigir qualquer erro que ocorra em uma rajada de comprimento não superior a b (BLAUM; FARRELL; TILBORG, 1998).

Considere um código de bloco linear C definido sobre o corpo finito $\mathbb{F}_q$, caracterizado pelos parâmetros $[n, k, d]$, onde n representa o comprimento do código, k a dimensão e d a distância mínima de Hamming, todos expressos em termos de símbolos q-ários.

Seja $b > 0$ um número inteiro positivo tal que b divida k. Nessas condições, pode-se construir um código matricial linear sobre $\mathbb{F}_q^b$, com parâmetros $[n', k', d']$, onde $k' = \frac{k}{b}$, $n' = \frac{n}{b}$ e d' representa a distância mínima considerando o código sobre $\mathbb{F}_q^b$. Assim, esse código é parametrizado em função de blocos de símbolos q-ários de comprimento b.

Tais códigos são normalmente descritos por meio de uma matriz de verificação de paridade H de dimensão $(n' - k')b \times n'b$, a qual gera palavras-código de comprimento $n'b$, sendo b o comprimento da rajada de erro.

Observa-se que o limitante de Singleton, apresentado no Teorema 3.3.1, permanece válido para os códigos matriciais lineares. Quando a igualdade da Equação 3.21 é satisfeita, o código é dito como um **código matricial linear MDS**.

Por fim, apresenta-se a seguir um exemplo geral de códigos matriciais lineares, com o intuito de exemplificar sua relação com os códigos lineares. O Capítulo 5 será dedicado à abordagem dos métodos de codificação e decodificação desses códigos, os quais possuem a propriedade MDS e são capazes de corrigir erros em rajada.

Exemplo 3.4.2 *Seja um código de bloco linear C sobre $\mathbb{F}_2$ com parâmetros $[9, 3, 3]$, formado pelas palavras-código* $C = \{000000000,\ 100110100,\ 010101010,\ 001011101,\ 110011110,\ 011110111,\ 101101001,\ 111000011\}$ *geradas a partir da matriz geradora G, dada por*

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

De acordo com a Equação 3.6, a matriz verificação de paridade associada a este código é dada por

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

Tomando $b = 3$, podemos olhar C como um código matricial linear sobre $\mathbb{F}_2^3$ com parâmetros $[3, 1, 1]$. Neste caso, a matriz de verificação de paridade pode ser organizada em blocos da forma

$$H = \left[\begin{array}{ccc:c:ccc:c:ccc} 1 & 1 & 0 & \vdots & 1 & 0 & 0 & \vdots & 0 & 0 & 0 \\ 1 & 0 & 1 & \vdots & 0 & 1 & 0 & \vdots & 0 & 0 & 0 \\ 0 & 1 & 1 & \vdots & 0 & 0 & 1 & \vdots & 0 & 0 & 0 \\ \hdots & \hdots & \hdots & \hdots & \hdots & \hdots & \hdots & \hdots & \hdots & \hdots & \hdots \\ 1 & 0 & 1 & \vdots & 0 & 0 & 0 & \vdots & 1 & 0 & 0 \\ 0 & 1 & 0 & \vdots & 0 & 0 & 0 & \vdots & 0 & 1 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 & 0 & \vdots & 0 & 0 & 1 \end{array}\right]. \tag{3.23}$$

As palavras-código poderiam ser representadas como $C = \{000\ 000\ 000,\ 100\ 110\ 100,\ 010\ 101\ 010,\ 001\ 011\ 101,\ 110\ 011\ 110,\ 011\ 110\ 111,\ 101\ 101\ 001,\ 111\ 000\ 011\}$, com erros analisados em rajadas de $b = 3$.

Como a distância mínima $d_{min} = 1$, o código não é capaz de detectar ou corrigir rajadas de comprimento $b = 3$.

4 MATRIZ REDUNDANTE DE DISCOS INDEPENDENTES (RAIDS)

Neste capítulo são apresentados os conceitos fundamentais relacionados a um tipo de Sistema de Armazenamento Distribuído (*Distributed Storage System*), conhecido como Matriz Redundante de Discos Independentes (RAID, *Redundant Array of Independent Disks*), bem como suas principais classificações e métodos de operação. Na Seção 4.1, são abordados os conceitos básicos e as técnicas empregadas bem como são apresentados os níveis de RAIDs junto com suas vantagens e desvantagens. Na Seção 4.2 é explicada a codificação e decodificação no RAID 6, e exemplificado o processo de detecção de erros. Os conceitos e definições abordados neste capítulo podem ser encontrados em (JACOB, 2009), (CORPORATION, 2005), (PRESTES, 2023), (ARPACI-DUSSEAU, 2008) e (MOUSSA; RYCHLIK, 2018).

4.1 CLASSIFICAÇÃO DOS RAIDS

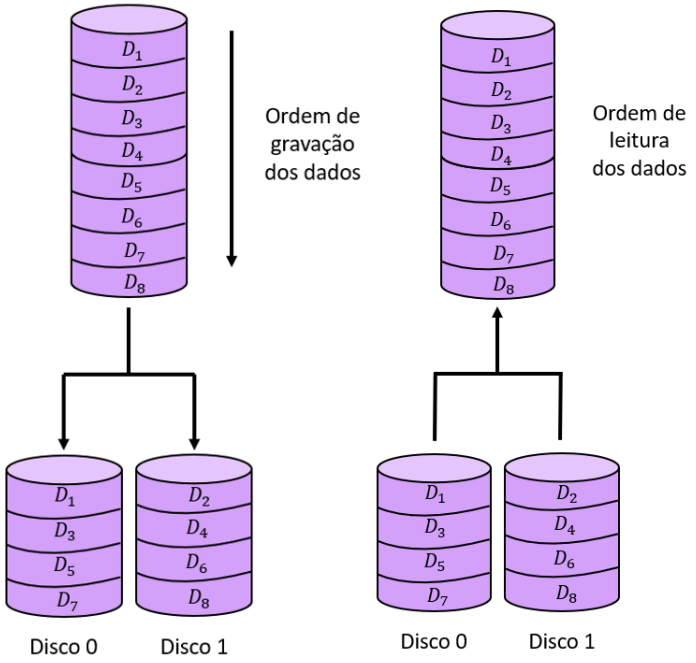
A concepção fundamental da Matriz Redundante de Discos Independentes, RAID, consiste em unir diversas unidades de disco menores e mais econômicos em um conjunto, a fim de atingir objetivos de desempenho ou redundância que seriam impossíveis de alcançar com uma única unidade. Para atingir tal meta, emprega-se técnicas como o *disk striping* (RAID 0), o *disk mirroring* (RAID 1) e o *disk striping with parity* (RAID 5), as quais visam reduzir a latência, ampliar a banda de passagem, maximizar a capacidade de recuperação de falhas no disco rígido e alcançar a redundância desejada.

Em um sistema RAID, os dados são distribuídos por cada unidade do conjunto, sendo em seguida divididos nos blocos de tamanho uniforme. Cada bloco é então gravado em um disco rígido dentro do conjunto RAID, de acordo com o nível empregado. Quando os dados são lidos, esse processo é revertido, criando a impressão de que as diversas unidades dentro do conjunto são, de fato, uma única unidade, como demostrado na Figura 4.

A técnica *striping* é usada para distribuir os dados uniformemente entre os discos de um conjunto. Isso significa que os dados são divididos em blocos (*stripes*), e cada bloco é armazenado em um dos discos, seguindo uma ordem sequencial. Assim, cada disco armazena partes diferentes dos dados, aumentando o desempenho de leitura e escrita paralela. A técnica *disk mirroring* ocorre quando os dados são duplicados integralmente em dois, ou mais discos. E, a técnica *disk striping with parity*, é a divisão de dados em blocos distribuídos entre discos considerando algum tipo de paridade, com a informação de redundância para reconstrução dos dados em caso de falha de disco.

Existem diferentes níveis de RAID, cada um com características específicas de desempenho e redundância, que variam conforme as necessidades do sistema. Eles são classificados de acordo com sua capacidade e sua confiabilidade baseado no número de falhas de disco que o sistema pode tolerar. A seguir, são apresentados os principais níveis de RAID.

Figura 4 – Processo de gravação e leitura dos dados.



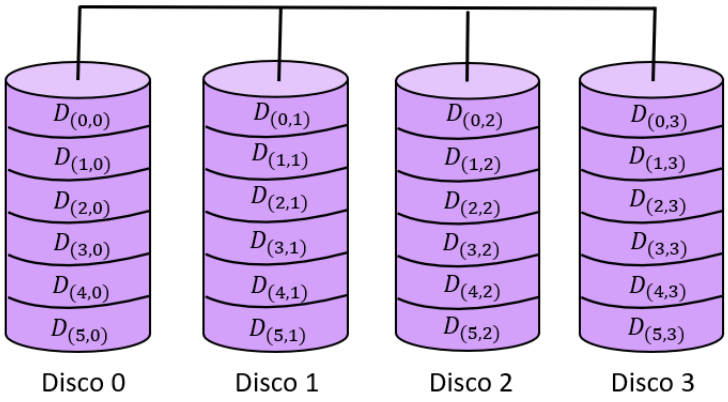
Fonte: Elaborado pela autora.

4.1.1 RAID 0

No RAID 0 a técnica *striping* é empregada, ou seja, os dados são divididos em pequenos segmentos e distribuídos uniformemente entre as unidades de armazenamento. Os discos são mesclados em um único grande volume. A configuração dos discos distribuídos como uma única unidade aumenta o desempenho, pois vários discos executam operações de leitura e gravação simultaneamente.

Trata-se de um nível que não oferece proteção contra falhas, pois não existe redundância nele. Com capacidade de N (discos) x B (blocos), sendo todo o espaço usado para armazenar dados. Devido a essas características, ele é muito usado em aplicações que lidam com grandes volumes de dados e que não podem apresentar lentidão, como tratamento de imagens e edição de vídeos. Um exemplo do RAID 0 é mostrado na Figura 5, assim como na Tabela 12, com suas vantagens e desvantagens.

Figura 5 – Exemplo RAID 0.



Fonte: Elaborado pela autora.

Tabela 12 – Vantagens e desvantagens do RAID 0.

Vantagens	Desvantagem
- Custo-benefício e simples de implementar; - Maior desempenho de leitura e gravação; - Sem sobrecarga (uso da capacidade total); - Simples de configurar e fácil de usar;	- Não fornece tolerância à falhas;

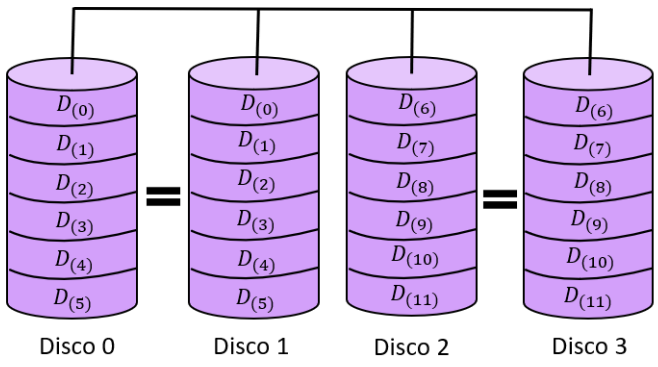
Fonte: Produção da Própria Autora.

4.1.2 RAID 1

Fornece redundância gravando os dados idênticos em cada disco da matriz, como mostrado na Figura 6. Esse nível opera com dois ou mais discos que podem usar acesso paralelo para altas taxas de transferência de dados durante a leitura, mas geralmente operam independentemente para fornecer altas taxas de dados. Ele fornece confiabilidade de dados e alto desempenho para aplicativos de leitura, porém seu custo é alto, como exemplificado na Tabela 13.

O RAID 1 é focado na proteção dos dados, porém ele não dispensa soluções de *backup*. Como a duplicação dos dados é feita em tempo real, se uma informação indevida for gravada na primeira unidade ou se um arquivo importante for apagado, o mesmo acontecerá no segundo disco. Portanto, seu uso costuma ser mais indicado para proteger o sistema de falhas físicas nas unidades de armazenamento.

Figura 6 – Exemplo RAID 1.



Fonte: Elaborado pela autora.

Tabela 13 – Vantagens e desvantagens do RAID 1.

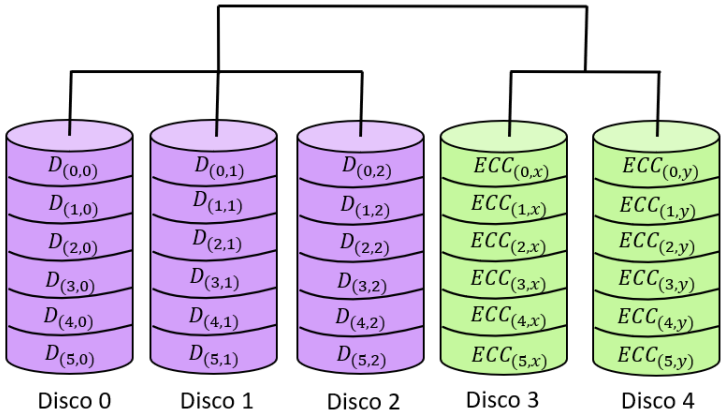
Vantagens	Desvantagens
- Maior desempenho de leitura; - Fornece redundância; - Tolerância à falhas; - Simples de configurar e fácil de usar;	- Usa metade da capacidade de armazenamento; - Custo relativamente alto; - Perda no desempenho;

Fonte: Produção da Própria Autora.

4.1.3 RAID 2

Tipo de solução de armazenamento que surgiu no final de 1980, o RAID 2, demonstrado na Figura 7, foi criado como solução para a limitação dos HDs que não tinham um padrão de confiabilidade. Esse nível é parecido com o RAID 0, porém é o primeiro nível de RAID que conta com um mecanismo de detecção de falhas como um código corretor de erros (ECC, *Error Correcting Code)*, o que é visto como uma vantagem, como mostrado na Tabela 14. Ao gravar os dados, ele grava também o código no disco de redundância. Posteriormente, ao ler dados do disco, ele também lê do disco de redundância para verificar os dados e fazer correções, se necessário.

Figura 7 – Exemplo RAID 2.



Fonte: Elaborado pela autora.

Tabela 14 – Vantagens e desvantagens do RAID 2.

Vantagens	**Desvantagens**
- Capacidade de corrigir informações armazenadas;	- Alto custo;
- Confiabilidade;	- Difícil implementação;
- Taxas de transferência de dados altas;	- Requer discos inteiros para ECC;

Fonte: Produção da Própria Autora.

Um código corretor de erros muito utilizado no RAID 2 é o código de Hamming apresentado na Subseção 3.2.

4.1.4 RAID 3

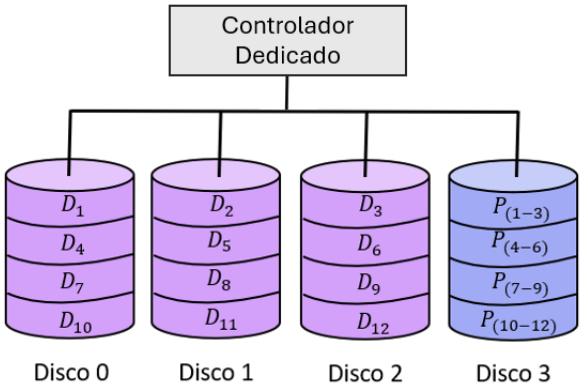
O RAID 3, apresentado na Figura 8, utiliza distribuição em nível de *bytes* (sequência de 8 bits) e um disco de paridade dedicado, ou seja, cada bloco de dados é dividido em *bytes* e distribuído nos discos de dados juntamente com os *bytes* de paridade correspondentes. Por causa disso, requer pelo menos três unidades, onde duas são usadas para armazenar faixas de dados e uma é usada para paridade.

Para a realização da leitura, os dados são lidos simultaneamente a partir de cada unidade de disco e a paridade é usada para recuperar os dados perdidos em caso de falha em uma das unidades de

disco. Durante a leitura, os dados de cada disco são combinados com a paridade correspondente para reconstruir os dados originais.

Além disso, o RAID 3 requer um controlador especializado que é responsável por gerenciar as operações de gravação e leitura em todas as unidades de disco. Ele calcula e escreve a paridade nos discos dedicados de paridade, atingindo melhores taxas de desempenho com operações sequenciais do que com operações aleatórias de leitura/gravação, como mostra a Tabela 15.

Figura 8 – Exemplo RAID 3.



Fonte: Elaborado pela autora.

Tabela 15 – Vantagens e desvantagens do RAID 3.

Vantagens	**Desvantagens**
- Boa taxa de transferência para grande quantidades de dados; - Alta eficiência com operações sequenciais; - Resiliência de falha de disco;	- Não é adequado para transferir arquivos pequenos; - Complexo de implementar; - Difícil configuração como *software*;

Fonte: Produção da Própria Autora.

A seguir apresentamos um exemplo do funcionamento do RAID 3.

Exemplo 4.1.1 *Dada uma mensagem* $u = [001000011001001000110110]$*, tem-se a distribuição de dados em nível de bytes mostrada abaixo.*

0	**1**	**2**	**P**
00100001	10010010	00110110	

Para ser calculada a paridade P*, é preciso fazer XOR simples com os discos de dados. Assim,* $P = 00100001 \oplus 10010010 \oplus 00110110 = 10000101$.

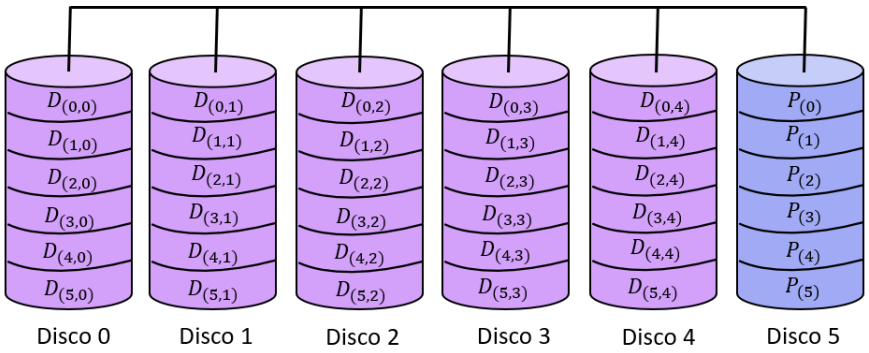
0	**1**	**2**	**P**
00100001	10010010	00110110	10000101

4.1.5 RAID 4

O RAID 4, mostrado na Figura 9, também tem uma técnica de paridade calculada usando uma função XOR simples, além de apresentar funcionamento similar ao RAID 3. O seu diferencial está em dividir os dados em blocos e oferecer acesso individual a cada unidade de armazenamento do sistema. Portanto, não há necessidade de um controlador especializado.

Além disso, esse nível pode apresentar algum comprometimento de desempenho, citado na Tabela 16, pois toda e qualquer operação de gravação exige atualização na unidade de paridade. Por esse motivo, seu uso é mais indicado para sistemas que priorizam leitura de dados, ou seja, que realizam muito mais consultas do que operações de gravação. E com relação a sua confiabilidade, permite a recuperação de no máximo uma falha de disco e sua paridade está centralizada em um único disco.

Figura 9 – Exemplo RAID 4.



Fonte: Elaborado pela autora.

Tabela 16 – Vantagens e desvantagens do RAID 4.

Vantagens	**Desvantagens**
- Operações de leitura rápida e independente; - Baixa sobrecarga de armazenamento;	- Operações de gravação lentas; - A redundância é perdida se o disco de paridade falhar;

Fonte: Produção da Própria Autora.

A seguir vemos um exemplo da utilização do RAID 4.

Exemplo 4.1.2 *Dada uma mensagem* $u = [0010000110010010001101100101010111001010001011100]$*, tem-se a distribuição de dados em nível de blocos mostrada abaixo.*

0	**1**	**2**	**P**
00100001 10010010	00110110 01010101	11001010 00101100	

Para ser calculada a Paridade, também é possível fazer XOR simples com os discos de dados.

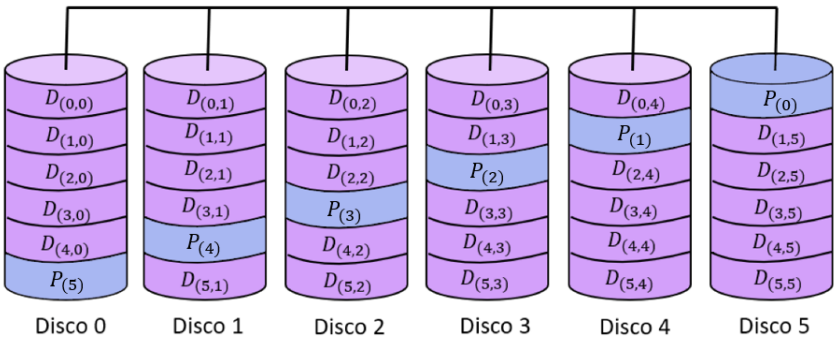
0	**1**	**2**	**P**
00100001 10010010	00110110 01010101	11001010 00101100	11011101 1110011

4.1.6 RAID 5

O RAID 5 é a implementação de RAID mais utilizada na atualidade. Ele combina a técnica *striping* e paridade para fornecer uma configuração rápida e confiável, oferecendo ao usuário usabilidade de armazenamento como no RAID 1 e a eficiência de desempenho do RAID 0.

No RAID 5 os dados são divididos em faixas e distribuídos em diferentes discos na matriz, mostrado na Figura 10. Isso permite altas taxas de desempenho, citado na Tabela 17, devido à transações de dados de leitura rápida que podem ser feitas simultaneamente por diferentes unidades na matriz, e sua paridade é calculada igual ao RAID 4 e distribuída pelos discos.

Figura 10 – Exemplo RAID 5.



Fonte: Elaborado pela autora.

Tabela 17 – Vantagens e desvantagens do RAID 5.

Vantagens	**Desvantagens**
- Alta performance e capacidade; - Velocidade de leitura rápida e confiável; - Tolera falha de unidade única;	- Maior tempo de reconstrução; - Se mais de um disco falhar, os dados serão perdidos;

Fonte: Produção da Própria Autora.

A seguir apresentamos um exemplo do funcionamento do RAID 5.

Exemplo 4.1.3 *Dada a mesma mensagem do Exemplo 4.1.2, tem-se a distribuição de dados em nível de blocos mostrada abaixo.*

0	**1**	**P**	**2**
00100001 10010010	00110110 01010101		11001010 00101100

Para ser calculada a Paridade, assim como no RAID 4, fazemos um XOR simples com os discos de dados.

0	**1**	**P**	**2**
00100001 10010010	00110110 01010101	11011101 1110011	11001010 00101100

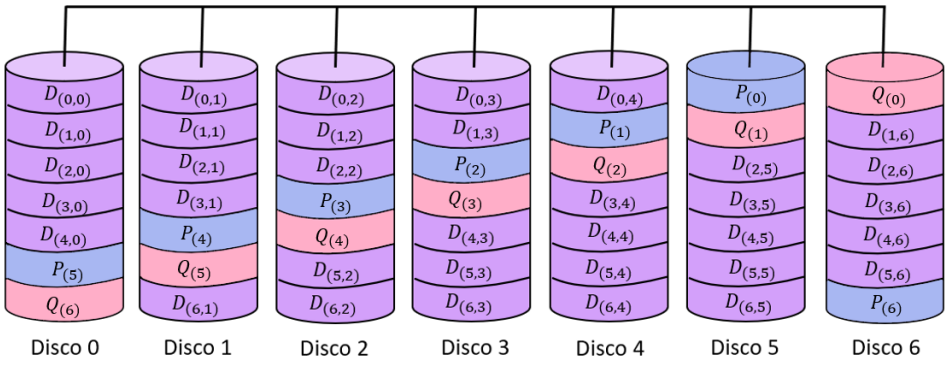
4.1.7 RAID 6

Conhecido como “RAID de paridade dupla”, é semelhante ao RAID 5, mas incluindo um segundo esquema de paridade distribuído diagonalmente pelos discos na matriz. Esta configuração requer um mínimo de quatro unidades. Utiliza *striping* em nível de blocos para distribuir os dados pela matriz e reserva dois blocos de paridade para os dados da linha, exemplificado na Figura 11. O uso de paridade adicional permite que a matriz continue a funcionar mesmo se duas unidades falharem ao mesmo tempo. Este tipo de configuração oferece boa tolerância a falhas, sendo utilizado em ambientes onde é priorizada a segurança dos dados armazenados.

Seu desempenho, citado na Tabela 18, depende de como a matriz é implementada, bem como do número total de unidades. As operações de gravação são mais lentas em comparação com outras configurações devido ao seu recurso de dupla paridade. Sua tecnologia usa códigos MDS baseados no GF para codificar dados nas unidades para proteger dados de erro ou apagamento e está intimamente relacionada aos códigos Reed-Solomon, com um sistema que corrige até dois erros com dois discos de paridade quando a posição da falha é conhecida.

Sua implementação consiste em formar uma palavra código tomando um único *byte* de cada um dos k discos de dados na matriz e calcular os dois símbolos de paridade (P e Q) armazenados nos blocos de paridade. Como o comprimento máximo da palavra-código em $\mathbb{F}_{2^8}$ é 255, o número máximo de blocos por linhas n que podem ser suportados é 255 que inclui 253 unidades de dados e 2 unidades de paridade.

Figura 11 – Exemplo RAID 6.



Fonte: Elaborado pela autora.

Tabela 18 – Vantagens e desvantagens do RAID 6.

Vantagens	**Desvantagens**
- Alta tolerância à falhas; - Alta eficiência de armazenamento; - Operação de leitura rápida;	- Tempo de reconstrução pode levar até 24 horas; - Baixo desempenho de gravação - Implementação complexa;

Fonte: Produção da Própria Autora.

Na seção a seguir iremos mostrar uma forma de realizar a codificação e a decodificação no RAID 6.

4.2 CODIFICAÇÃO E DECODIFICAÇÃO NO RAID 6

Os dados do RAID 6 podem ser representados por *bytes*, em notação hexadecimal ou binária. A codificação usa dois blocos de paridade rotativos para criar redundância denotadas como P e Q, mostrados na Figura 12.

Figura 12 – RAID 6 representado na forma de tabela.

Discos / Linhas	0	1	2	3	4	5	6
0	$D_{(0,0)}$	$D_{(0,1)}$	$D_{(0,2)}$	$D_{(0,3)}$	$D_{(0,4)}$	$P_{(0)}$	$Q_{(0)}$
1	$D_{(1,0)}$	$D_{(1,1)}$	$D_{(1,2)}$	$D_{(1,3)}$	$P_{(1)}$	$Q_{(1)}$	$D_{(1,6)}$
2	$D_{(2,0)}$	$D_{(2,1)}$	$D_{(2,2)}$	$P_{(2)}$	$Q_{(2)}$	$D_{(2,5)}$	$D_{(2,6)}$
3	$D_{(3,0)}$	$D_{(3,1)}$	$P_{(3)}$	$Q_{(3)}$	$D_{(3,4)}$	$D_{(3,5)}$	$D_{(3,6)}$
4	$D_{(4,0)}$	$P_{(4)}$	$Q_{(4)}$	$D_{(4,3)}$	$D_{(4,4)}$	$D_{(4,5)}$	$D_{(4,6)}$
5	$P_{(5)}$	$Q_{(5)}$	$D_{(5,2)}$	$D_{(5,3)}$	$D_{(5,4)}$	$D_{(5,5)}$	$D_{(5,6)}$
6	$Q_{(6)}$	$D_{(6,1)}$	$D_{(6,2)}$	$D_{(6,3)}$	$D_{(6,4)}$	$D_{(6,5)}$	$P_{(6)}$

Fonte: Elaborado pela autora.

O primeiro bloco de paridade P usa uma paridade XOR da forma

$$P = \sum_{i=0}^{N_D-1} D_i, \tag{4.1}$$

sendo D_i os blocos de dados na linha que está sendo codificado e N_D o número de discos de armazenamento de dados na matriz.

Já o segundo bloco de paridade Q também é criado de maneira semelhante, porém usa um GF multiplicação para multiplicar 2^i que representa cada bloco de dados

$$Q = \sum_{i=0}^{N_D-1} (2^i \otimes D_i). \tag{4.2}$$

Quando os blocos de dados são codificados, cada dado é dividido em palavras binárias de 8 bits e a palavra correlatada em cada bloco de dados é chamada de *word stripe*. Cada *word stripe* possui uma palavra-código P e Q calculada individualmente e armazenada nas unidades de paridade.

Exemplo 4.2.1 *Utilizando o polinômio primitivo* $p(x) = x^8 + x^4 + x^3 + x^2 + 1$ *e dada uma mensagem recebida de 3 bytes* $u = [10101010\ 00001111\ 11111111]$, *tem-se a distribuição de dados convertidos para hexadecimais mostrada abaixo.*

0	1	2	P	Q
0xAA	0x0F	0xFF		

Primeiramente, é necessário calcular a paridade P dada por 4.1

$$P = \sum_{i=0}^{N_D-1} D_i = 1010\ 1010\ \oplus 0000\ 1111\ \oplus 1111\ 1111 = 0101\ 1010 = 0x5A,$$

e determinar a outra paridade Q a partir de 4.2

$$\begin{aligned} Q &= \sum_{i=0}^{N_D-1} (2^i \otimes D_i) = (0x01 \cdot 0xAA)\ \oplus (0x02 \cdot 0x0F)\ \oplus (0x04 \cdot 0xFF) \\ &= (0000\ 0001 \cdot 1010\ 1010)\ \oplus (0000\ 0010 \cdot 0000\ 1111)\ \oplus (0000\ 0100 \cdot 1111\ 1111) \\ &= [x^7 + x^5 + x^3 + x]\ \oplus [x^4 + x^3 + x^2 + x]\ \oplus [(x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2)\ \bmod F] \\ &= 1010\ 1010\ \oplus 0001\ 1110\ \oplus 1101\ 1011 = 0110\ 1111 = 0x6F. \end{aligned}$$

Portanto, as duas paridades calculadas são:

0	**1**	**2**	**P**	**Q**
0xAA	0x0F	0xFF	0x5A	0x6F

A decodificação no RAID 6 permite a recuperação de até dois blocos de dados corrompidos ou perdidos em uma mesma linha, por meio do uso dos blocos de paridade P e Q. Esse processo é essencial para garantir a integridade e disponibilidade das informações armazenadas, mesmo em caso de falha simultânea de unidades. Existem várias maneiras de detectar esses erros, seja do arquivo sistema, ou o hardware detectando falha do dispositivo de armazenamento, ou incorporando um CRC ou detecção de erro de código nos dados antes da codificação. Quando esses erros são detectados, existem 4 casos possíveis que teriam que ser tratados para corrigir todas as permutações possíveis de onde dois erros poderiam estar. A exemplificação destes casos é mostrada na Figura 13, onde cada linha representa um cenário possível.

Figura 13 – Possibilidades de erros de blocos.

Linhas \ Discos	0	1	2	3	4	5	6
0	$D_{(0,0)}$	$D_{(0,1)}$	$D_{(0,2)}$	$D_{(0,3)}$	$D_{(0,4)}$	$P_{(0)}$	$Q_{(0)}$
1	$D_{(1,0)}$	$D_{(1,1)}$	$D_{(1,2)}$	$D_{(1,3)}$	$P_{(1)}$	$Q_{(1)}$	$D_{(1,6)}$
2	$D_{(2,0)}$	$D_{(2,1)}$	$D_{(2,2)}$	$P_{(2)}$	$Q_{(2)}$	$D_{(2,5)}$	$D_{(2,6)}$
3	$D_{(3,0)}$	$D_{(3,1)}$	$P_{(3)}$	$Q_{(3)}$	$D_{(3,4)}$	$D_{(3,5)}$	$D_{(3,6)}$
4	$D_{(4,0)}$	$P_{(4)}$	$Q_{(4)}$	$D_{(4,3)}$	$D_{(4,4)}$	$D_{(4,5)}$	$D_{(4,6)}$
5	$P_{(5)}$	$Q_{(5)}$	$D_{(5,2)}$	$D_{(5,3)}$	$D_{(5,4)}$	$D_{(5,5)}$	$D_{(5,6)}$
6	$Q_{(6)}$	$D_{(6,1)}$	$D_{(6,2)}$	$D_{(6,3)}$	$D_{(6,4)}$	$D_{(6,5)}$	$P_{(6)}$

Fonte: Elaborado pela autora.

- **Caso 1: Dois erros nos blocos de paridade (P e Q).** Se os dois blocos corrompidos forem os dois blocos de paridade, como mostrado na primeira linha da Figura 12, eles podem ser regenerados a partir dos dados originais usando o mesmo método de como eles são gerados na codificação.

- **Caso 2: Um erro no bloco de paridade (Q) e um erro no bloco de dados (D_i).** Se um dos erros estiver em um bloco de dados e o outro estiver no bloco Q, como mostrado na segunda linha da Figura 13, o bloco de dados corrompido pode ser encontrado usando o bloco de paridade P e os outros blocos de dados aplicando XOR. Para corrigir o erro encontrado em um dos blocos de dados D_i é feito XOR entre os blocos de dados corretos e a paridade P da forma

$$D_i = P \oplus \cdots \oplus D_{N_D-1}. \tag{4.3}$$

Como alternativa, D_i também pode ser encontrado calculando uma nova paridade P', apenas com os blocos de dados corretos:

$$P' = D_0 \oplus \cdots \oplus D_{N_D-1}, \tag{4.4}$$

e posteriormente, fazendo XOR entre P e P':

$$D_i = P \oplus P'. \tag{4.5}$$

Assim, depois de encontrar o bloco de dados correto, é possível calcular Q usando o mesmo método mostrado em (4.2).

Exemplo 4.2.2 *Após a gravação de dados nos discos do RAID 6, foi calculado os blocos de paridade P e Q. Quando foi feita a solicitação de leitura dos dados, ocorreu um problema de apagamento nos blocos D_1 e Q, mostrado abaixo.*

0	1	2	P	Q
0xAA	——	0xFF	0x5A	——

Primeiramente, é calculado uma nova paridade P', através de (4.4),

$$P' = D_0 \oplus \cdots \oplus D_{N_D-1} = 0xAA \oplus 0xFF = 1010\ 1010\ \oplus 1111\ 1111 = 0101\ 0101 = 0x55,$$

e feito XOR entre P' e P, dado por (4.5), assim, corrigindo o problema de apagamento do bloco D_1:

$$D_1 = P \oplus P' = 0x5A \oplus 0x55 = 0101\ 1010\ \oplus 0101\ 0101 = 0000\ 1111 = 0x0F.$$

Por fim, é calculada a paridade Q, apresentada em (4.2), com os dados gravados nos discos.

$$
\begin{aligned}
Q &= \sum_{i=0}^{N_D-1} (2^i \otimes D_i) = (0x01 \cdot 0xAA) \ \oplus (0x02 \cdot 0x0F) \ \oplus (0x04 \cdot 0xFF) \\
&= (0000\ 0001 \cdot 1010\ 1010) \ \oplus (0000\ 0010 \cdot 0000\ 1111) \ \oplus (0000\ 0100 \cdot 1111\ 1111) \\
&= [x^7 + x^5 + x^3 + x] \ \oplus [x^4 + x^3 + x^2 + x] \ \oplus [x^7 + x^6 + x^4 + x^3 + x + 1] \\
&= 1010\ 1010 \ \oplus 0001\ 1110 \ \oplus 1101\ 1011 = 0110\ 1111 = 0x6F.
\end{aligned}
$$

Portanto, o dado D_1 e a paridade Q são:

0	1	2	P	Q
0xAA	0x0F	0xFF	0x5A	0x6F

- **Caso 3: Um erro no bloco de paridade (P) e um erro no bloco de dados (D_i).** Se houver um erro no bloco de dados e um erro no bloco P, exemplificado na terceira linha da Figura 13, os dados são encontrados recalculando a nova paridade Q' com todos os blocos de dados:

$$Q' = \sum_{i=0}^{N_D-1} (2^i \otimes D_i). \tag{4.6}$$

A partir de Q', os dados originais podem ser encontrados através de

$$D_i = (Q \oplus Q') \oslash 2^i. \tag{4.7}$$

Todavia, devido a complexidade da divisão, a Equação 4.7 pode ser simplificada, utilizando logaritmos e exponenciais:

$$D_i = \exp\left[(\log(Q \oplus Q') - \log(2^i)) \mod 0xFF\right]. \tag{4.8}$$

Então, depois de encontrar o bloco de dados correto, consultando as Tabelas 7 e 8, apresentadas na Seção 2.1.2, é possível calcular P usando o mesmo método mostrado em (4.1).

Exemplo 4.2.3 *Após a gravação de dados nos discos do RAID 6, foi calculado os blocos de paridade P e Q. Quando foi feita a solicitação de leitura dos dados, ocorreu um problema de apagamento nos blocos D_2 e P, mostrado abaixo.*

0	1	2	P	Q
0xAA	0x0F	—	—	0x6F

Inicialmente, é calculado a nova paridade Q' utilizando (4.6):

$$Q' = \sum_{i=0}^{N_D-1} (2^i \otimes D_i) = (0x01 \cdot 0xAA) \oplus (0x02 \cdot 0x0F)$$
$$= [x^7 + x^5 + x^3 + x] \oplus [x^4 + x^3 + x^2 + x] = 1011\ 0100 = 0xB4,$$

e posteriormente, aplicada a Equação 4.7 com o auxilio das Tabelas 7 e 8 para corrigir o bloco de dados apagado:

$$D_2 = (Q \oplus Q') \oslash 2^2 = (0x6F \oplus 0xB4) \oslash 2^2 = (0110\ 1111 \oplus 1011\ 0100) \oslash 0x04$$
$$= (1101\ 1011) \oslash 0x04 = (0xDB) \oslash 0x04 = \exp\left[\log(0xDB) - \log(0x04)\right]$$
$$= \exp\left[0xB1 - 0x02\right] = \exp\left(0xAF\right) = 0xFF.$$

Portanto, o dado e a paridade corretos são:

0	1	2	P	Q
0xAA	0x0F	0xFF	0x5A	0x6F

- **Caso 4: Dois erros nos blocos de dados (D_j e D_i).** O último caso é quando ambos os erros estão nos blocos de dados em locais conhecidos, mostrados na quarta linha da Figura 13, definidos como j e i, assumindo que $j < i$. Assumido essa condição, são calculados novos blocos de paridade P' e Q', como anteriormente em (4.4) e (4.6), respectivamente. Assim, é possível, primeiramente, corrigir o bloco D_j como:

$$D_j = [(2^{-j} \otimes (Q \oplus Q')) \oplus (2^{i-j} \otimes (P \oplus P'))] \oslash [2^{i-j} \oplus 1], \tag{4.9}$$

e posteriormente, corrigir o bloco D_i da forma:

$$D_i = P \oplus P' \oplus D_j. \tag{4.10}$$

Exemplo 4.2.4 *Após a gravação de dados nos discos do RAID 6, foi calculado os blocos de paridade P e Q. Quando foi feita a solicitação de leitura dos dados, ocorreu um problema de apagamento nos blocos de dados D_1 e D_2, mostrado abaixo.*

0	1	2	P	Q
0xAA	——	——	0x5A	0x6F

Primeiramente, é calculado uma nova paridade P', através de (4.4):

$$P' = D_0 \oplus \cdots \oplus D_{N_D-2} = D_0 = 0xAA.$$

Calculamos também a nova paridade Q' utilizando (4.6):

$$Q' = \sum_{i=0}^{N_D-1} (2^i \otimes D_i) = 2^0 \otimes D_0 = 0xAA.$$

Sabendo que j < i, ou seja, $j = 1$ e $i = 2$ e consultando as Tabelas 7 e 8 para calcular D_j através de (4.9), tem-se

$$\begin{aligned} D_1 &= [2^{-1} \otimes (0x6F \oplus 0xAA) \oplus 2^{2-1} \otimes (0x5A \oplus 0xAA)] \oslash [2^{2-1} \oplus 1] \\ &= [2^{-1} \otimes (1100\ 0101) \oplus 2 \otimes (1111\ 0000)] \oslash [2 \oplus 1] = [(0xC5 \oslash 2) \oplus (2 \otimes 0xF0)] \oslash 0x03 \\ &= [(\exp[\log(0xC5) - \log(0x02)]) \oplus (x \cdot (x^7 + x^6 + x^5 + x^4))] \oslash 0x03 \\ &= [0xEC \oplus 1111\ 1101] \oslash 0x03 = [1110\ 1100 \oplus 1111\ 1101] \oslash 0x03 \\ &= 0001\ 0001 \oslash 0x03 = \exp[\log(0x11) - \log(0x03)] = \exp[0x4B] = 0x0F. \end{aligned}$$

Por fim, é possível corrigir o bloco D_i a partir da Equação 4.10:

$$D_2 = P \oplus P' \oplus D_1 = 0101\ 1010 \oplus 1010\ 1010 \oplus 0000\ 1111 = 1111\ 1111 = 0xFF.$$

Portanto, os dois blocos de dados corretos são:

0	1	2	P	Q
0xAA	0x0F	0xFF	0x5A	0x6F

A técnica de decodificação de apagamento é baseada na suposição de que as localizações dos blocos apagados são conhecidas. Outro caso que pode ocorrer são quando os dados no sistema são corrompidos enquanto todas as unidades estão em boas condições, o que é chamado de corrupção silenciosa de dados. Neste cenário, os erros são aleatórios e o sistema precisa encontrar suas localizações e corri-los.

O RAID 6 pode corrigir um erro em um bloco de dados usando apenas as paridades incorporadas quando não se sabe a localização. Isso permite que muitos erros nos dados sejam reconhecidos e corrigidos, desde que haja apenas um erro de blocos por linha. Para aplicativos, a probabilidade de ter mais de uma palavra corrompida por linhas é muito baixa. Se houver mais de uma corrupção de blocos por faixa, há uma chance de ser detectável, mas também há uma chance de que múltiplos erros sejam diagnosticados erroneamente como apenas erro de um bloco.

Se houver vários erros na faixa de palavras, aparecerá como um único erro em um local de dados. A probabilidade desse erro de diagnóstico ocorrer quando há vários erros em uma faixa é

$$P_{erro} = \frac{N_D}{2^n - 1}, \tag{4.11}$$

sendo N_D o número total de discos e n o número de bits em cada bloco.

Exemplo 4.2.5 *A probabilidade de ocorrer erro de diagnóstico (quando múltiplos erros são diagnosticados como apenas um erro) para o RAID 6 de 6 discos com 1 byte em cada bloco é*

$$P_{erro} = \frac{N_D}{2^n - 1} = \frac{6}{2^8 - 1} = \frac{6}{255} = 0,0235.$$

Supondo que uma única palavra seja corrompida após a codificação. Essa palavra corrompida está em uma posição desconhecida i, na qual a palavra de dados original D_i foi alterada para algum valor X. A localização e a correção do erro podem ser obtidas da seguinte maneira.

Primeiramente, recalculam-se os dois valores de paridade correspondentes aos dados corrompidos denotados por P' e Q'. Se P' e Q' diferirem dos valores originais P e Q gerados durante a codificação, existe a presença de um erro na linha. Assim, a posição i do erro pode ser determinada aplicando a seguinte relação:

$$i = \log_{\alpha}[(Q \oplus Q') \oslash (P \oplus P')], \tag{4.12}$$

sendo α um elemento primitivo do corpo finito $\mathbb{F}_{2^8}$.

Exemplo 4.2.6 *Após a gravação de dados nos discos do RAID 6, foi calculado os blocos de paridade P e Q. Quando foi feita a solicitação de leitura dos dados, foi calculado novamente as paridades e notou-se diferença com as paridades gravadas, mostrando que ocorreu corrupção em algum bloco mostrado abaixo.*

0	1	2	P	Q
0xAA	0x0F	0xEF	0x5A	0x6F

Primeiramente, utilizando (4.4) para encontrar o valor de P', tem-se

$$P' = D_0 \oplus \cdots \oplus D_{N_D-1} = 1010\ 1010 \oplus 0000\ 1111 \oplus 1110\ 1111 = 0100\ 1010 = 0x4A.$$

E, usando (4.6) para calcular Q', tem-se

$$\begin{aligned} Q' &= \sum_{i=0}^{N_D-1} (2^i \otimes D_i) = (0000\ 0001 \cdot 1010\ 1010) \ \oplus (0000\ 0010 \cdot 0000\ 1111) \ \oplus (0000\ 0100 \cdot 1110\ 1111) \\ &= [1 \cdot (x^7 + x^5 + x^3 + x)] \ \oplus [x \cdot (x^3 + x^2 + x + 1)] \ \oplus [x^2 \cdot (x^7 + x^6 + x^5 + x^3 + x^2 + x + 1)] \\ &= [x^7 + x^5 + x^3 + x] \ \oplus [x^4 + x^3 + x^2 + x] \ \oplus [x^7 + x^4 + x^3 + x + 1] \\ &= 1010\ 1010 \ \oplus 0001\ 1110 \ \oplus 1001\ 1011 = 0010\ 1111 = 0x2F. \end{aligned}$$

Depois, é aplicado (4.12) para encontrar a localização do erro:

$$\begin{aligned} i &= \log[(Q \oplus Q') \oslash (P \oplus P')] = \log[(0110\ 1111 \oplus 0010\ 1111) \oslash (0101\ 1010 \oplus 0100\ 1010)] \\ &= \log[(0x40) \oslash (0x10)] = \log[\exp[(\log(0x40) - \log(0x10)) \mod 0xFF]] \\ &= \log[\exp[(6 - 4) \mod 0xFF]] = 2 \end{aligned}$$

Por fim, sabendo que $i = 2$, é feito XOR entre a paridade e os dados corretos para corrigir D_2:

$$D_2 = P \oplus D_0 \oplus D_1 = 0101\ 1010 \oplus 1010\ 1010 \oplus 0000\ 1111 = 1111\ 1111 = 0xFF.$$

Portanto, a gravação correta dos dados nos discos do RAID 6 é:

0	1	2	P	Q
0xAA	0x0F	0xFF	0x5A	0x6F

5 CODIFICAÇÃO E DECODIFICAÇÃO DE CÓDIGOS MATRICIAIS MDS

Neste capítulo, são abordados os processos de codificação e decodificação em códigos matriciais MDS. Na Seção 5.1, descreve-se o procedimento de codificação desses códigos a partir da construção da matriz de verificação de paridade, utilizando a matriz companheira em conjunto com matrizes superregulares, como discutido na Seção 2.3. Na Seção 5.2, é apresentado um algoritmo capaz de detectar e corrigir até duas rajadas de erro, conforme proposto em (CARDELL; CLIMENT; REQUENA, 2013) e (ZANITTI, 2021), além de uma simplificação desse algoritmo empregando matrizes superregulares de Vandermonde. Por fim, na Subeção 5.2.1, apresenta-se a extensão da capacidade de correção para até três rajadas de erro, incluindo também uma simplificação específica para o caso de matrizes superregulares de Vandermonde.

5.1 CODIFICAÇÃO

Nesta seção, é apresentada a metodologia de codificação aplicada aos códigos matriciais MDS. Inicialmente, define-se a matriz companheira de Frobenius, juntamente com um isomorfismo que associa um elemento primitivo de um corpo finito à respectiva matriz companheira, estabelecendo assim a base teórica necessária para a construção dos códigos.

Definição 5.1.1 *Seja $p(x) = x^b + p_{b-1}x^{b-1} + ... + p_1x + p_0 \in \mathbb{F}_q[x]$ um polinômio primitivo. Podemos associar a p(x) uma matriz chamada matriz companheira de Frobenius, que é uma matriz que contém 1's na subdiagonal e a última coluna é dada em função dos coeficientes de p(x), sendo os demais elementos todos nulos, como segue*

$$C = \begin{bmatrix} 0 & 0 & ... & 0 & -p_0 \\ 1 & 0 & ... & 0 & -p_1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & ... & 0 & -p_{b-2} \\ 0 & 0 & ... & 1 & -p_{b-1} \end{bmatrix}. \tag{5.1}$$

Seja $\alpha \in \mathbb{F}_{q^b}$ um elemento primitivo do corpo finito $\mathbb{F}_{q^b}$, ou seja, uma raiz de um polinômio primitivo $p(x) \in \mathbb{F}_q[x]$. Existe então um isomorfismo $\phi : \mathbb{F}_{q^b} \longrightarrow \mathbb{F}_q[C]$, definido por $\phi(\alpha) = C$. O comportamento desse isomorfismo é exemplificado a seguir.

Exemplo 5.1.1 *Sejam $\{0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, ...\alpha^{14}\}$ os elementos do corpo finito $\mathbb{F}_{2^4}$ obtidos através do polinômio primitivo $p(x) = x^4 + x + 1 \in \mathbb{F}_2[x]$. A matriz companheira de Frobenius gerada a partir deste polinômio é dada por*

$$C = \begin{bmatrix} 0 & 0 & 0 & -p_0 \\ 1 & 0 & 0 & -p_1 \\ 0 & 1 & 0 & -p_2 \\ 0 & 0 & 1 & -p_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{5.2}$$

Assim, na Tabela 19, tem-se o isomorfismo $\phi : \mathbb{F}_{2^4} \longrightarrow \mathbb{F}_2[C]$ *definido por* $\phi(\alpha) = C$.

Tabela 19 – Isomorfismo $\phi : \mathbb{F}_{2^4} \longrightarrow \mathbb{F}_2[C]$.

Elemento em $\mathbb{F}_{2^4}$	$\phi(\alpha)$
0	0
$\alpha^0 = 1$	I_4
α	C
α^2	C^2
α^3	C^3
α^4	C^4
α^5	C^5
α^6	C^6
α^7	C^7
α^8	C^8
α^9	C^9
α^{10}	C^{10}
α^{11}	C^{11}
α^{12}	C^{12}
α^{13}	C^{13}
α^{14}	C^{14}

Fonte: Produção da Própria Autora.

Considerando as matrizes $M_{m\times n}(\mathbb{F}_{q^b})$ de ordem $m \times n$ com elementos no corpo finito $\mathbb{F}_{q^b}$, pode-se estender o isomorfismo mostrado acima, como

$$\begin{array}{ccc} \psi : M_{m\times n}(\mathbb{F}_{q^b}) & \longrightarrow & M_{m\times n}(\mathbb{F}_q[C]) \\ \psi(A) & \longmapsto & [\phi(\alpha_{ij})] \end{array}. \tag{5.3}$$

O comportamento deste isomorfismo é mostrado no exemplo abaixo.

Exemplo 5.1.2 *Considere o polinômio dado no Exemplo 5.1.1 e seja* A *uma matriz de Vandermonde como segue*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{bmatrix}.$$

Portanto, para o isomorfismo $\phi : M_{3\times3}(\mathbb{F}_{2^4}) \longrightarrow M_{3\times3}(\mathbb{F}_2[C])$*, tem-se*

$$\psi(A) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ C & C^2 & C^3 & C^4 \\ C^2 & C^4 & C^6 & C^8 \\ C^3 & C^6 & C^9 & C^{12} \end{bmatrix}.$$

Seja C um código matricial linear sobre $\mathbb{F}_{q^b}$ com parâmetros $[n', k', d']$, conforme apresentado na Seção 3.4. Por conveniência visual, denota-se esse código por $[n, k, d]$, sendo representado por blocos de comprimento b. Se o código de distância d satisfaz a igualdade expressa em (3.21), então C é um código matricial MDS. Nessas condições, o resultado a seguir apresenta uma matriz de verificação de paridade de um código matricial linear definido sobre $\mathbb{F}_{q^b}$, que preserva a propriedade MDS.

Teorema 5.1.1 *Se* $A = [\alpha_{ij}] \in M_{(n-k\times k)}(\mathbb{F}_{q^b})$ *é uma matriz superregular, então* $H = [\varphi(A)|I_{(n-k)\cdot b}]$*, onde* ψ *é o isomorfismo dado em (5.3), é uma matriz controle de paridade de um* $[n, k, n-k+1]$ *código matricial linear MDS.*

Dado um código matricial MDS $\mathcal{C}$ sobre $\mathbb{F}_{2^b}$, sendo b um inteiro positivo, construído de acordo com as condições estabelecidas pelo Teorema 5.1.1. Os parâmetros desse código são $[n, k, n-k+1] = [m+k, k, m+1]$, observando que $m = n-k$. A matriz de controle de paridade deste código é da forma

$$H = \left[\begin{array}{cccc|c} A_{11} & A_{12} & ... & A_{1k} & \\ A_{21} & A_{22} & ... & A_{2k} & I_{mb} \\ \vdots & \vdots & & \vdots & \\ A_{m1} & A_{m2} & ... & A_{mk} & \end{array}\right], \tag{5.4}$$

com $A_{ij} = C^{\sigma(i,j)}$, em que $C \in M_{b\times b}(\mathbb{F}_2)$ é a matriz companheira de um polinômio primitivo, dada em (5.1), e $\sigma(i, j)$ é a potência de cada elemento (i, j) da matriz superregular A de dimensão $m \times k$.

Exemplo 5.1.3 *Considere o polinômio primitivo* $p(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$*. Deseja-se construir um código matricial MDS com parâmetros [4, 2, 3] sobre* $\mathbb{F}_{2^3}$ *, ou seja, m = k = 2. Primeiramente, para a construção da matriz verificação de paridade é preciso encontrar a matriz companheira de Frobenius que é dada por*

$$C = \begin{bmatrix} 0 & 0 & -p_0 \\ 1 & 0 & -p_1 \\ 0 & 1 & -p_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Seja $\alpha \in \mathbb{F}_{2^3}$ um elemento primitivo tal que $\alpha^3+\alpha+1=0$. Uma matriz superregular de Vandermonde de dimensão 2×2 é construída como

$$A = \begin{bmatrix} 1 & 1 \\ \alpha & \alpha^2 \end{bmatrix}.$$

Por fim, de acordo com (5.4), tem-se que a matriz verificação de paridade para este código é

$$H = \left[\begin{array}{cc|c} I_3 & I_3 & I_6 \\ C & C^2 & \end{array}\right] = \left[\begin{array}{ccc:ccc|cccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]. \tag{5.5}$$

5.2 DECODIFICAÇÃO

O objetivo desta seção é apresentar um algoritmo capaz de detectar e corrigir até duas rajadas de erros de comprimento b de um código matricial MDS com parâmetros $[m+k, k, m+1]$ para todo $m \geq 4$ e $m = n-k$, como apresentado em (CARDELL; CLIMENT; REQUENA, 2013) e (ZANITTI, 2021).

Primeiramente, será apresentado o algoritmo capaz de corrigir apenas uma rajada de erro de comprimento b em um código matricial MDS. Quando $m = 2$ e 3 o código será capaz de corrigir até uma rajada de erro de comprimento b, pois a capacidade de correção de erros de um código linear é

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

Algoritmo 5.2.1 *(ZANITTI, 2021) Seja $\mathcal{C}$ um código matricial MDS com parâmetros $[m+k, k, m+1]$, $m \geq 2$, e H sua matriz controle de paridade dada por (5.4). Para $v = [v_1, ..., v_n]$ um vetor recebido, onde $n = m+k$, tem-se:*

1. *Calcular a síndrome $s = [s_1\ s_2\ ...\ s_m]$ de v, a partir de*

$$s_i^T = \sum_{j=1}^{k} A_{ij} v_j^T + v_{k+i}^T = \sum_{j=1}^{k} A_{ij} e_j^T + e_{k+i}^T, \tag{5.6}$$

   para cada $i = 1, ..., m$.

2. *Se $s_j \neq 0$ e $s_i = 0$, para todo $i \neq j$ e $i = 1, ..., m$, então fazer $e_{k+j} = s_j$ e a posição do erro está em $k+j$. Caso contrário, fazer $l_1 = 0$.*

3. *Fazer $l_1 = l_1 + 1$. Se $l_1 = k$, então o algoritmo finaliza pois mais de um erro ocorreu. Caso contrário, ir para o próximo passo.*

4. *Calcular os vetores*

$$\begin{aligned} y_1^T &= s_1^T + A_{1l_1}A_{ml_1}^{-1}s_m^T \\ y_2^T &= s_2^T + A_{2l_1}A_{1l_1}^{-1}s_1^T \\ y_3^T &= s_3^T + A_{3l_1}A_{2l_1}^{-1}s_2^T \\ &\vdots \\ y_m^T &= s_m^T + A_{ml_1}A_{(m-1)l_1}^{-1}s_{m-1}^T \end{aligned} \quad . \tag{5.7}$$

5. *Caso os cálculos de todos os y forem nulos, então existe uma rajada de erros de comprimento b na posição l_1 dada por*

$$e_{l_1}^T = A_{il_1}^{-1}s_i^T, \tag{5.8}$$

para todo $i = 1, ..., m$, e seguir com o algoritmo. Caso contrário, voltar para o Passo 3.

6. *Corrigir o vetor recebido v através de*

$$c = v - e. \tag{5.9}$$

A seguir, apresentamos dois exemplos para mostrar o funcionamento do Algoritmo 5.2.1.

Exemplo 5.2.1 *Considere o código matricial MDS construído no Exemplo 5.1.3. Este código possui capacidade de correção de uma rajada de erro com comprimento $b = 3$, uma vez que $\left\lfloor \frac{m}{2} \right\rfloor = \left\lfloor \frac{2}{2} \right\rfloor = 1$. Suponha que o vetor recebido seja $v = [101\ 110\ 110\ 011]$. De acordo com o Passo 1 do algoritmo, é necessário calcular as síndromes através da Equação 5.6, obtendo*

$$s_1^T = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad e \quad s_2^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Uma vez obtidas as síndromes e somente uma é diferente de zero, o erro ocorreu no símbolo de paridade, e assim, seguindo o passo 2 do algoritmo, tem- se

$$e_3 = s_1 = [101].$$

E, portanto,

$$c = v - e = [101\ 110\ 110\ 011] - [000\ 000\ 101\ 000] = [101\ 110\ \mathbf{011}\ 011],$$

foi a palavra-código enviada.

Exemplo 5.2.2 *Seja o polinômio primitivo $p(x) = x^4 + x^3 + 1 \in \mathbb{F}_2[x]$ e seja $\mathcal{C}$ um código matricial MDS com parâmetros $[8, 4, 5]$ sobre $\mathbb{F}_{2^4}$. A matriz companheira de Frobenius C e a matriz superregular de Vandermonde A são dadas por*

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad e \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{bmatrix}.$$

Através destas matrizes, obtemos a matriz verificação de paridade para o código $\mathcal{C}$

$$H = \left[\begin{array}{cccc|c} I_4 & I_4 & I_4 & I_4 & \\ C & C^2 & C^3 & C^4 & I_{16} \\ C^2 & C^4 & C^6 & C^8 & \\ C^3 & C^6 & C^9 & C^{12} & \end{array}\right].$$

Supondo que a palavra recebida foi $v = [1000\ 1001\ 0110\ 0111\ 0110\ 1011\ 1010\ 1110]$. *Ao calcular as síndromes tem-se:*

$$s_1^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad s_2^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad s_3^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad s_4^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Como todas as síndromes são diferentes de 0, segue que os erros ocorreram nos símbolos de informação. Considerando $l_1 = 1$, *calcula-se os vetores expressos na Equação 5.7,*

$$y_1^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad y_2^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad y_3^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad y_4^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Como todos os resultados foram diferentes de zero, deve-se voltar ao Passo 3 do algoritmo e atribuir um novo valor para a possibilidade de erro, sendo $l_1 = 2$*:*

$$y_1^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad y_2^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad y_3^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad y_4^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Obtendo-se o valor nulo nos vetores, é confirmado o erro na posição $l_1 = 2$. *Para obter a magnitude deste erro, basta realizar o cálculo da Equação 5.8,*

$$e_2^T = A_{12}^{-1} \cdot s_1^T = I_4^{-1} \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

E, portanto,

$$c = v - e = [1000\ \mathbf{1111}\ 0110\ 0111\ 0110\ 1011\ 1010\ 1110].$$

A seguir, o algoritmo enunciado faz referência a correção de dois erros.

Algoritmo 5.2.2 *(ZANITTI, 2021) Seja $\mathcal{C}$ um código matricial MDS com parâmetros $[m+k, k, m+1]$, $m \geq 4$, e H sua matriz controle de paridade dada por (5.4). Para $v = [v_1, ..., v_n]$ um vetor recebido, onde $n = m + k$, tem-se:*

1. *Calcular a síndrome $s = [s_1\ s_2\ ...\ s_m]$ de v, a partir de (5.6).*

2. *Se $s_j \neq 0$ e $s_i = 0$, para todo $i \neq j$ e $i = 1, ..., m$, então faça $e_{k+j} = s_j$ e a posição do erro está em $k + j$.*

3. *Se $s_{j1} \neq 0$, $s_{j2} \neq 0$ e $s_i = 0$, para todo $i \neq j_1 \neq j_2$ e $i = 1, ..., m$, então fazer $e_{k+j_1} = s_{j1}$ e $e_{k+j_2} = s_{j_2}$. Nesse caso os erros ocorreram nas posições $k + j_1$ e $k + j_2$. Caso contrário, seja $l_1 = 0$.*

4. *Fazer $l_1 = l_1 + 1$. Se $l_1 = k$, então o algoritmo finaliza pois ocorreram mais de dois erros. Caso contrário, ir para o próximo passo.*

5. *Calcular os vetores*

$$y_i^T = s_i^T + A_{il_1} A_{(i-1)l_1}^{-1} s_{i-1}^T, \tag{5.10}$$

sendo $i = 1, \ldots, m$.

6. *Se os cálculos de todos y forem nulos, então existe uma rajada de erros de comprimento b na posição l_1 dado por*

$$e_{l_1}^T = A_{il_1}^{-1} \cdot s_i^T, \tag{5.11}$$

para todo $i = 1, ...m$. Caso contrário, ir para o próximo passo.

7. *Se $(y_i, y_{i+1}) = (0, 0)$ ou $(y_m, y_1) = (0, 0)$, mas não todos nulos, então existe uma rajada de erros de comprimento b na posição l_1 e outra rajada nos bits de paridade, respectivamente, dados por*

$$e_{l_1}^T = A_{il_1}^{-1} \cdot s_i^T, \tag{5.12}$$

e

$$e_{k+j} = s_j^T - A_{jl_1} \cdot e_{l_1}^T, \tag{5.13}$$

para todo $i \neq j$ e $i = 1, ..., m - 1$, onde j corresponde à síndrome em comum nos cálculos de y não nulos. Caso contrário, ir para o próximo passo.

8. *Para $l_2 \in \{l_1 + 1, l_1 + 2, \ldots, k\}$, se*

$$y_i^T = C^{r_{i-2}} y_{i-1}^T, \tag{5.14}$$

para todo $i = 3, 4, \ldots, m$, com

$$\begin{aligned} r_{i-2} = \sigma(i,l_2) - \sigma(i-1,l_2) + Z(\sigma(i,l_1) - \sigma(i,l_2) - \sigma(i-1,l_1) + \sigma(i-1,l_2)) \\ - Z(\sigma(i-1,l_1) - \sigma(i-1,l_2) - \sigma(i-2,l_1) + \sigma(i-2,l_2)), \end{aligned} \tag{5.15}$$

ocorreram dois erros, um na posição l_1 e outro na posição l_2.

9. *Se o algoritmo declarar que os erros ocorreram nas posições l_1 e l_2, para obter a magnitude dos erros basta resolver o seguinte sistema linear*

$$\begin{cases} A_{il_1} e_{l_1}^T + A_{il_2} e_{l_2}^T = s_i^T \\ A_{jl_1} e_{l_1}^T + A_{jl_2} e_{l_2}^T = s_j^T \end{cases}, \tag{5.16}$$

para $i, j = 1, 2, \ldots, m$ e $i \neq j$. Caso contrário, retornar para o Passo 4.

10. *Corrigir o vetor recebido v através de $c = v - e$.*

A prova do funcionamento do Algoritmo 5.2.2 é dada no seguinte resultado.

Teorema 5.2.1 *Se C é um código matricial linear MDS com parâmetros $[m+k, k, m+1]$, com $m \geq 4$, k, $b \in \mathbb{N}$ e matriz controle de paridade H dada em (5.4), então o Algoritmo 5.2.2 corrige até duas rajadas de erros de comprimento b.*

Demonstração 5.2.1 *Como $n = m + k$, os erros na palavra recebida $v = [v_1, \ldots, v_n]$ podem ocorrer nos k símbolos de informação, como mostrado no Exemplo 5.2.6, ou nos m símbolos de paridade. Será analisada cada uma dessas possibilidades. No caso em que $m > k$, a eficiência de armazenamento é menor. Quando $m = k$, obtém-se um equilíbrio entre proteção e uso de armazenamento, e para $m < k$, a eficiência aumenta devido à menor redundância.*

- ***Caso 1: Dois erros nos símbolos de paridade (m).*** *Se 1 ou 2 erros ocorreram nos símbolos de paridade, então $m - 1$ ou $m - 2$ síndromes são nulos, respectivamente. Se os erros ocorreram nas posições $k + j_1$ e $k + j_2$, então as síndromes não nulas são, respectivamente, s_{j1} e s_{j2}, e as magnitudes dos erros são dadas por*

$$e_{k+j1} = s_{j1} \quad e \quad e_{k+j2} = s_{j2}. \tag{5.17}$$

Exemplo 5.2.3 *Considere o polinômio primitivo $p(x) = x^5 + x^2 + 1 \in \mathbb{F}_2[x]$. Deseja-se construir um código matricial MDS com parâmetros [11, 5, 7] sobre $\mathbb{F}_{2^5}$, ou seja, m > k. Primeiramente, para a construção da matriz verificação de paridade é preciso encontrar a matriz companheira de $p(x)$ que é dada por*

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Seja $\alpha \in \mathbb{F}_{2^5}$ *um elemento primitivo raiz de* $p(x)$ *tal que* $\alpha^5 + \alpha^2 + 1 = 0$*. A matriz superregular de Vandermonde de dimensão* 6×5 *é construída como*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} \\ \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} \\ \alpha^5 & \alpha^{10} & \alpha^{15} & \alpha^{20} & \alpha^{25} \end{bmatrix}.$$

Agora, de acordo com (5.4), tem-se que a matriz verificação de paridade para este código é

$$H = \left[\begin{array}{ccccc|c} I_5 & I_5 & I_5 & I_5 & I_5 & \\ C & C^2 & C^3 & C^4 & C^5 & \\ C^2 & C^4 & C^6 & C^8 & C^{10} & I_{30} \\ C^3 & C^6 & C^9 & C^{12} & C^{15} & \\ C^4 & C^8 & C^{12} & C^{16} & C^{20} & \\ C^5 & C^{10} & C^{15} & C^{20} & C^{25} & \end{array}\right].$$

Supondo que a palavra recebida foi $v = [00001\ 00010\ 10101\ 10000\ 10100\ 10010\ 10101\ 01011\ 11000\ 00000\ 01001]$*. Ao calcular as síndromes dada por (5.6), tem-se:*

$$s_1^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad s_2^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad s_3^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad s_4^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad s_5^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad s_6^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Como apenas duas síndromes são diferentes de 0, temos que dois erros ocorreram nos símbolos de paridade. O primeiro na posição $k + 2$*, sendo* $e_7 = s_2$*, e o segundo na posição k + 4, dado por* $e_9 = s_4$*. Assim, a palavra-código corrigida de acordo com (5.9) é*

$$c = v - e = [00001\ 00010\ 10101\ 10000\ 10100\ 10010\ \mathbf{00011}\ 01011\ \mathbf{10110}\ 00000\ 01001].$$

- ***Caso 2: Um erro no símbolo de informação****. Supondo que um único erro ocorreu em um símbolo de informação na posição* l_1*. Então, as síndromes e a magnitude do erro é obtida utilizando (5.6) e (5.8), respectivamente, apresentados no Algoritmo 5.2.1.*

- ***Caso 3: Um erro no símbolo de informação e um erro no símbolo de paridade.*** *Supondo que o erro no símbolo de informação ocorreu na posição l_1 e que o erro no símbolo de paridade ocorreu na posição $k+1$. Então através de (5.6), as síndromes são dadas por*

$$s_1^T = A_{11}e_1^T + e_{k+1}^T \quad e \quad s_i^T = A_{i1}e_1^T, \tag{5.18}$$

para todo $i = 2, \ldots, m$. Neste caso, tem que $y_1, y_2 \neq 0$, pois s_1 está presente no cálculo destes vetores, e $y_3 = \cdots = y_m = 0$. Dessa forma, a magnitude dos erros são dadas por

$$e_{k+1}^T = s_1^T \quad e \quad e_1^T = A_{21}^{-1}s_2^T = \cdots = A_{m1}^{-1}s_m^T, \tag{5.19}$$

sendo $e_{k+1}^T = s_1^T$ o valor do erro na posição $k+1$ após a correção do erro no símbolo de informação, e_1.

Exemplo 5.2.4 *Considere o polinômio primitivo $p(x) = x^5 + x^2 + 1 \in \mathbb{F}_2[x]$. Deseja-se construir um código matricial MDS com parâmetros [9, 5, 5] sobre $\mathbb{F}_{2^5}$, ou seja, m < k. Novamente, para a construção da matriz verificação é preciso encontrar a matriz companheira de Frobenius, já calculada no Exemplo 5.2.3. Seja $\alpha \in \mathbb{F}_{2^5}$ um elemento primitivo raiz de $p(x)$ tal que $\alpha^5 + \alpha^2 + 1 = 0$. A matriz superregular de Vandermonde de dimensão 4×5 é construída como*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} \end{bmatrix}.$$

Por fim, de acordo com (5.4), tem-se que a matriz verificação de paridade para este código é

$$H = \left[\begin{array}{ccccc|c} I_5 & I_5 & I_5 & I_5 & I_5 & \\ C & C^2 & C^3 & C^4 & C^5 & I_{20} \\ C^2 & C^4 & C^6 & C^8 & C^{10} & \\ C^3 & C^6 & C^9 & C^{12} & C^{15} & \end{array}\right].$$

Supondo que a palavra recebida foi $v = [00011\ 10001\ 10100\ 01110\ 10010\ 11001\ 00111\ 01110\ 01111]$. Ao calcular as síndromes dadas por (5.6), tem-se:

$$s_1^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad s_2^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad s_3^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad s_4^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Como todas as síndromes são diferentes de 0, temos que pelo menos 1 erro ocorreu no simbolo de informação. Considerando $l_1 = 1, 2, 3$, todos y_i são diferentes de 0. Para $l_1 = 4$ aplica-se o algoritmo para calcular os vetores definidos em (5.10),

$$y_1^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad y_2^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad y_3^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad y_4^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Note, que para este caso, tem-se dois vetores nulos, que confirma que houve um erro na posição $l_1 = 4$ e dois vetores não nulos. Destes, os diferentes de zero possuem em comum a síndrome s_3, assim, o segundo erro ocorreu na posição referente aos bits de paridade k + 3. Para corrigí-la, primeiramente, encontramos o erro na posição $l_1 = 4$ através da Equação 5.12, encontrando

$$e_4^T = A_{44}^{-1} \cdot s_4^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Para o segundo erro na posição dos bits de paridade, é possível calculá-lo através de (5.13)

$$e_8^T = s_3^T - A_{34} \cdot e_4^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Portanto, a palavra-código corrigida de acordo com (5.9) é

$$c = v - e = [00011\ 10001\ 10100\ \mathbf{01101}\ 10010\ 11001\ 00111\ \mathbf{10111}\ 01111].$$

- ***Caso 4: Dois erros nos símbolos de informação (k).*** *Suponha que os erros nos símbolos de informação estejam nas posições l_1 e l_2. Através de (5.6) as síndromes são calculadas por*

$$s_i^T = A_{il_1} e_{l_1}^T + A_{il_2} e_{l_2}^T, \tag{5.20}$$

para $i = 1, \ldots, m$. Substituindo (5.20) nos vetores y_i dados em (5.7), tem-se $y_i^T \neq 0$, para todo $i = 1, \ldots, m$. Para y_2, é obtido

$$\begin{aligned} y_2^T &= s_2^T + A_{2l_1} A_{1l_1}^{-1} s_1^T \\ &= (A_{2l_1} e_{l_1}^T + A_{2l_2} e_{l_2}^T) + A_{2l_1} A_{1l_1}^{-1} (A_{1l_1} e_{l_1}^T + A_{1l_2} e_{l_2}^T) \\ &= A_{2l_1} e_{l_1}^T + A_{2l_2} e_{l_2}^T + A_{2l_1} A_{1l_1}^{-1} A_{1l_1} e_{l_1}^T + A_{2l_1} A_{1l_1}^{-1} A_{1l_2} e_{l_2}^T \\ &= A_{2l_2} e_{l_2}^T + A_{2l_1} A_{1l_1}^{-1} A_{1l_2} e_{l_2}^T = (A_{2l_2} + A_{2l_1} A_{1l_1}^{-1} A_{1l_2}) e_{l_2}^T. \end{aligned} \tag{5.21}$$

Sendo $A_{ij} = C^{\sigma(i,j)}$, onde $C \in M_{b \times b}(\mathbb{F}_2)$ é como em (5.1) e $\sigma(i,j)$ é a potência de cada elemento (i,j) da matriz superregular A de dimensão $m \times k$, então

$$y_2^T = \left(C^{\sigma(2,l_2)} + C^{\sigma(2,l_1) - \sigma(1,l_1) + \sigma(1,l_2)}\right) e_{l_2}^T. \tag{5.22}$$

Utilizando o logaritmo de Zech apresentado na Seção 2.2 e definido por 2.2.1, tem-se

$$y_2^T = C^{\sigma(2,l_2)+Z(\sigma(2,l_1)-\sigma(1,l_1)+\sigma(1,l_2)-\sigma(2,l_2))} e_{l_2}^T. \tag{5.23}$$

Ou seja,

$$e_{l_2}^T = C^{-\sigma(2,l_2)-Z(\sigma(2,l_1)-\sigma(1,l_1)+\sigma(1,l_2)-\sigma(2,l_2))} y_2^T. \tag{5.24}$$

Da mesma forma,

$$y_3^T = C^{\sigma(3,l_2)+Z(\sigma(3,l_1)-\sigma(2,l_1)+\sigma(2,l_2)-\sigma(3,l_2))} e_{l_2}^T. \tag{5.25}$$

Substituindo (5.24) em (5.25), é obtido

$$y_3^T = C^{r_1} y_2^T, \tag{5.26}$$

onde

$$\begin{aligned} r_1 = {} & \sigma(3,l_2) - \sigma(2,l_2) + Z(\sigma(3,l_1) - \sigma(3,l_2) + \sigma(2,l_1) - \sigma(2,l_2)) \\ & - Z(\sigma(2,l_1) - \sigma(2,l_2) + \sigma(1,l_1) - \sigma(1,l_2)). \end{aligned} \tag{5.27}$$

Portanto, de modo análogo, para todo $i = 3, ..., m$, tem-se

$$y_i^T = C^{r_{i-2}} y_{i-1}^T, \tag{5.28}$$

sendo

$$\begin{aligned} r_{i-2} = {} & \sigma(i,l_2) - \sigma(i-1,l_2) + Z(\sigma(i,l_1) - \sigma(i,l_2) - \sigma(i-1,l_1) + \sigma(i-1,l_2)) \\ & - Z(\sigma(i-1,l_1) - \sigma(i-1,l_2) - \sigma(i-2,l_1) + \sigma(i-2,l_2)). \end{aligned} \tag{5.29}$$

Se as igualdades das Equações 5.26 e 5.28 forem satisfeitas, os erros de fato ocorreram nas posições l_1 e l_2. Assim, para encontrar a magnitude dos erros é necessário resolver o sistema linear com duas equações e duas incógnitas dado em (5.16).

Exemplo 5.2.5 *(ZANITTI, 2021) Seja o polinômio primitivo $p(x) = x^4 + x^3 + 1 \in \mathbb{F}_2[x]$. Através desse polinômio é possível construir um código matricial MDS denotado por $\mathcal{C}$ de parâmetros $[8, 4, 5]$ sobre $\mathbb{F}_{2^4}$, capaz de corrigir até duas rajadas de erro de comprimento $b = 4$, com $m = k$. Para este exemplo será utilizada a matriz superregular de Cauchy denotada por A. Através da matriz companheira de Frobenius dada no Exemplo 5.2.2 e da matriz de Cauchy, é possível obter a matriz verificação de paridade para o código $\mathcal{C}$:*

$$A = \begin{bmatrix} \alpha^{12} & \alpha^{11} & \alpha^{10} & \alpha^{9} \\ \alpha^{11} & \alpha^{12} & \alpha^{5} & \alpha^{7} \\ \alpha^{5} & \alpha^{1} & \alpha^{1} & \alpha^{4} \\ \alpha^{4} & \alpha^{4} & \alpha^{10} & \alpha^{6} \end{bmatrix} \quad e \quad H = \left[\begin{array}{cccc|c} C^{12} & C^{11} & C^{10} & C^{9} & \\ C^{11} & C^{12} & C^{5} & C^{7} & \\ C^{5} & C^{1} & C^{1} & C^{4} & I_{16} \\ C^{4} & C^{4} & C^{10} & C^{6} & \end{array}\right].$$

Suponha que a palavra recebida foi $v = [1011\ 0101\ 0111\ 1001\ 1000\ 1011\ 0111\ 1001]$. Ao calcular as síndromes em (5.6), obtém-se

$$s_1^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad s_2^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad s_3^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad s_4^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Como somente uma síndrome é nula, segue que os erros ocorreram nos símbolos de informação. Considerando $l_1 = 1$*, aplica-se o algoritmo para calcular os vetores definidos em 5.7,*

$$y_1^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad y_2^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad y_3^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad y_4^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Uma vez que todos os y_i *são diferentes de zero, dá-se seguimento ao algoritmo supondo um segundo erro em* $l_2 = 2$*. Assim, testa-se a Equação 5.14 através dos cálculos de* r_{i-2} *dados em (5.15), para* $i = 3$ *e* 4*:*

$$\begin{aligned} r_1 &= \sigma(3,2) - \sigma(2,2) + Z(\sigma(3,1) - \sigma(3,2) - \sigma(2,1) + \sigma(2,2)) - Z(\sigma(2,1) - \sigma(2,2) \\ &\quad - \sigma(1,1) + \sigma(1,2)) = 10 - 12 + Z(5 - 10 - 11 + 12) + Z(11 - 12 - 12 + 11) \\ &= -2 + Z(-4) - Z(-2) = 5, \end{aligned}$$

e

$$\begin{aligned} r_2 &= \sigma(4,2) - \sigma(3,2) + Z(\sigma(4,1) - \sigma(4,2) - \sigma(3,1) + \sigma(3,2)) - Z(\sigma(3,1) - \sigma(3,2) \\ &\quad - \sigma(2,1) + \sigma(2,2)) = 4 - 10 + Z(1 - 4 - 5 + 10) + Z(5 - 10 - 11 + 12) \\ &= -6 + Z(2) - Z(-4) = -11. \end{aligned}$$

Assim,

$$y_3^T \neq C^5 \cdot y_2^T \quad e \quad y_4^T \neq C^{-11} \cdot y_3^T,$$

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Uma vez que para $l_2 = 2$ *as equações falham, dá-se seguimento para outros valores de* l_2*. A seguir, é apresentada a verificação dos cálculos da Equação 5.15 para* $l_2 = 3$*:*

$$\begin{aligned} r_1 &= \sigma(3,3) - \sigma(2,3) + Z(\sigma(3,1) - \sigma(3,3) - \sigma(2,1) + \sigma(2,3)) - Z(\sigma(2,1) - \sigma(2,3) \\ &\quad - \sigma(1,1) + \sigma(1,3)) = 11 - 5 + Z(5 - 11 - 11 + 5) - Z(11 - 5 - 12 + 10) \\ &= 6 + Z(-12) - Z(4) = 7, \end{aligned}$$

e

$$\begin{aligned} r_2 &= \sigma(4,3) - \sigma(3,3) + Z(\sigma(4,1) - \sigma(4,3) - \sigma(3,1) + \sigma(3,3)) - Z(\sigma(3,1) - \sigma(3,3) \\ &\quad - \sigma(2,1) + \sigma(2,3)) = 9 - 11 + Z(1 - 9 - 5 + 11) - Z(5 - 11 - 11 + 5) \\ &= -2 + Z(-2) - Z(-12) = 1. \end{aligned}$$

Agora, para tais valores encontrados, tem-se

$$y_3^T = C^7 \cdot y_2^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad e \qquad y_4^T = C^1 \cdot y_3^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Confirmado através dos cálculos que os erros ocorreram nas posições $l_1 = 1$ e $l_2 = 3$, aplica-se o algoritmo para encontrar a magnitude dos erros através da Equação 5.16, obtendo

$$e_1^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad e \quad e_3^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Portanto, a palavra-código é

$$c = v - e = [\mathbf{1101}\ 0101\ \mathbf{1001}\ 1001\ 1000\ 1011\ 0111\ 1001].$$

O exemplo apresentado evidencia a complexidade envolvida no processo de decodificação quando ocorrem dois erros nos símbolos de informação, em particular, ao efetuar o Passo 8 do Algoritmo 5.2.2. Com base nessa análise, um dos objetivos desse trabalho é contribuir com uma simplificação do processo de decodificação baseada na utilização de matrizes superregulares de Vandermonde.

A seguir, é apresentado um corolário que mostra a simplificação do Caso 4 da demonstração do Teorema 5.2.1 quando a matriz superregular utilizada é uma matriz de Vandermonde.

Corolário 5.2.1 *Se A é matriz de Vandermonde definida em (2.15), então o Passo 8 do Algoritmo 5.2.2 é simplificado por*

$$r_{i-2} = l_2, \tag{5.30}$$

para todo $i = 3, 4, \ldots, m$, onde l_2 é a posição do segundo erro.

Demonstração 5.2.2 *Seja $A = [a_{ij}]$ uma matriz superregular de Vandermonde. Primeiramente, será demonstrado que o cálculo dos logaritmos de Zech, da Equação 5.15, se anulam para todo $i = 3, \ldots, m$, ou seja, que*

$$\begin{aligned} &Z(\sigma(i, l_1) - \sigma(i, l_2) - \sigma(i-1, l_1) + \sigma(i-1, l_2)) = \\ &Z(\sigma(i-1, l_1) - \sigma(i-1, l_2) - \sigma(i-2, l_1) + \sigma(i-2, l_2)), \end{aligned} \tag{5.31}$$

para todo $i = 3, \ldots, m$, em que $\sigma(i, j)$ é a potência de α em cada elemento a_{ij} de A. Igualando seus argumentos, tem-se

$$\begin{aligned} &\sigma(i, l_1) - \sigma(i, l_2) - \sigma(i-1, l_1) + \sigma(i-1, l_2) = \sigma(i-1, l_1) - \sigma(i-1, l_2) - \sigma(i-2, l_1) + \sigma(i-2, l_2) \\ &\implies \sigma(i, l_1) - 2\sigma(i-1, l_1) + \sigma(i-2, l_1) - \sigma(i, l_2) + 2\sigma(i-1, l_2) - \sigma(i-2, l_2) = 0. \end{aligned} \tag{5.32}$$

Como A é uma matriz de Vandemonde, então pela Definição 2.3.3, as suas linhas estão em progressão geométrica. Assim, considerando as linhas consecutivas $i-2$, $i-1$ e i, e sendo l_1 e l_2 duas colunas quaisquer, segue que

$$\sigma(i, l_1) = 2\sigma(i-1, l_1) - \sigma(i-2, l_1) \tag{5.33}$$

e

$$\sigma(i, l_2) = 2\sigma(i-1, l_2) - \sigma(i-2, l_2). \tag{5.34}$$

Substituindo (5.33) e (5.34) em (5.32), obtém-se que de fato os logaritmos de Zech se anulam. Logo, a Equação 5.15 pode ser simplificada por

$$r_{i-2} = \sigma(i, l_2) - \sigma(i-1, l_2), \tag{5.35}$$

para todo $i = 3, \ldots, m$. Novamente pelo fato de A ser uma matriz de Vandermonde, se considerar duas linhas consecutivas $i-1$ e i, e uma coluna l_2 fixa, segue que a diferença entre as potências de α em $\sigma(i, l_2)$ e $\sigma(i-1, l_2)$ é igual a l_2, o que nos fornece (5.30), e finaliza a demonstração.

Exemplo 5.2.6 *Seja o polinômio primitivo $p(x) = x^4 + x^3 + 1 \in \mathbb{F}_2[x]$. Deseja-se construir um código matricial MDS de parâmetros $[8, 4, 5]$ sobre $\mathbb{F}_{2^4}$. Usando a matriz companheira de Frobenius C, a matriz superregular de Vandermonde A e a matriz H de paridade calculadas no Exemplo 5.2.2, e supondo que a palavra recebida foi $v = [0110\ 0001\ 0100\ 0001\ 1100\ 0011\ 0001\ 1001]$. Ao calcular as síndromes dadas em (5.6), tem-se:*

$$s_1^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad s_2^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad s_3^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad s_4^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Como todas as síndromes são diferentes de 0, segue que os erros ocorreram nos símbolos de informação. Considerando $l_1 = 1$, segue o algoritmo calculando os vetores expressos na Equação 5.10,

$$y_1^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad y_2^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad y_3^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad y_4^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Como todos os y_i são diferentes de zero, dá-se seguimento ao algoritmo supondo um segundo erro em $l_2 = 2$, assim, testa-se a Equação 5.14 através dos cálculos de r_{i-2} como (5.30), para $i = 3$ e 4:

$$r_1 = r_2 = l_2 = 2,$$

obtendo

$$y_3^T \neq C^2 \cdot y_2^T \qquad e \qquad y_4^T \neq C^2 \cdot y_3^T$$

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Uma vez que para $l_2 = 2$ as equações falham, dá-se seguimento para outros valores de l_2. Para

$$r_1 = r_2 = l_2 = 3,$$

segue que

$$y_3^T = C^3 \cdot y_2^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad e \qquad y_4^T = C^3 \cdot y_3^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

O que mostra que os erros ocorreram nas posições 1 e 3, e suas magnitudes são obtidas resolvendo o sistema linear da Equação 5.16 que fornece: $e_1 = [1011]$ e $e_2 = [0101]$. Assim, a palavra-código corrigida é

$$c = v - e = [\mathbf{1101}\ 0001\ \mathbf{0001}\ 0001\ 1100\ 0011\ 0001\ 1001].$$

Como evidenciado nos exemplos apresentados para cada possibilidade do Algoritmo 5.2.2, observamos que a relação entre m e k desempenha um papel decisivo no equilíbrio entre eficiência de armazenamento e tolerância a falhas. Embora, sob a condição $m \geq 4$, o código seja sempre capaz de corrigir no máximo duas rajadas de erro, essa limitação estabelece um ponto de atenção, pois impõe um limite à robustez alcançada em comparação ao custo adicional de armazenamento.

Quando $m > k$, a redundância torna-se superior à quantidade de dados, resultando em alta robustez e maior tolerância a falhas. Contudo, esse ganho vem acompanhado de menor aproveitamento do espaço e maior custo de armazenamento, configurando um regime altamente seguro, porém menos eficiente. Para o caso de $m = k$, obtém-se um ponto de equilíbrio: a redundância se iguala à informação útil, produzindo uma eficiência de armazenamento de aproximadamente 50%. Esse caso representa um compromisso estável entre proteção e custo, constituindo um resultado relevante para aplicações que buscam desempenho moderado com segurança consistente.

Por fim, o caso $m < k$ destaca-se como o mais promissor em termos de eficiência, pois reduz a redundância sem comprometer significativamente a confiabilidade. Os resultados indicam que essa configuração proporciona melhor aproveitamento de armazenamento e menor complexidade computacional, sendo especialmente adequada para aplicações práticas de grande escala, nas quais o volume de dados é elevado e a taxa de falhas permanece baixa. Em comparação com sistemas RAID, em particular o RAID 6, que apresentam dois discos de redundância o que permite a correção de

até dois discos defeituosos desde que suas posições sejam previamente conhecidas, os códigos MDS apresentam maior complexidade. Entretanto, estes códigos não requerem o conhecimento prévio da localização dos erros para realizar a correção.

5.2.1 Correção de Três Rajadas de Erros

Apresenta-se a seguir o desenvolvimento da ampliação da capacidade de correção de erros, ou seja, o que precisaria ser alterado para um algoritmo capaz de detectar e corrigir até três rajadas de erro em um código matricial MDS $[m+k, k, m+1]$, com $m \geq 6$. Para generalizar o procedimento para três rajadas, mantém-se a mesma metodologia, levando-se em conta, entretanto, o aumento da complexidade das formulações matemáticas envolvidas.

Nos casos em que ocorrem até três rajadas de erro nos bits de paridade, ou uma rajada nos bits de informação e duas nos bits de paridade, ou ainda duas rajadas nos bits de informação e uma nos bits de paridade, o comportamento da decodificação se assemelha ao Algoritmo 5.2.2 com os casos demonstrados no Teorema 5.2.1, com pequenas adaptações. O caso analisado a seguir corresponde à ocorrência de três rajadas de erros nos símbolos de informação.

Se três rajadas de erros ocorreram nas posições l_1, l_2 e l_3 então as síndromes são dadas por

$$s_i = A_{il_1} e_{l_1}^T + A_{il_2} e_{l_2}^T + A_{il_3} e_{l_3}^T, \tag{5.36}$$

para $i = 1, 2 \ldots, m$. Substituindo nos vetores $y_i^T \neq 0$ dados em (5.7), para todo $i = 1, 2, \ldots, m$, temos:

$$\begin{aligned}
\mathbf{y}_i^T &= \mathbf{s}_i^T + A_{il_1} A_{(i-1)l_1}^{-1} \mathbf{s}_{i-1}^T \\
&= \left(A_{il_2} + A_{il_1} A_{(i-1)l_1}^{-1} A_{(i-1)l_2}\right) \mathbf{e}_{l_2}^T + \left(A_{il_3} + A_{il_1} A_{(i-1)l_1}^{-1} A_{(i-1)l_3}\right) \mathbf{e}_{l_3}^T \\
&= \left(C^{\sigma(i,l_2)} + C^{\sigma(i,l_1)-\sigma(i-1,l_1)+\sigma(i-1,l_2)}\right) \mathbf{e}_{l_2}^T + \left(C^{\sigma(i,l_3)} + C^{\sigma(i,l_1)-\sigma(i-1,l_1)+\sigma(i-1,l_3)}\right) \mathbf{e}_{l_3}^T \\
&= C^{r_{i1}} \mathbf{e}_{l_2}^T + C^{r_{i2}} \mathbf{e}_{l_3}^T,
\end{aligned} \tag{5.37}$$

onde

$$r_{i1} = \sigma(i, l_2) + Z(\sigma(i, l_1) - \sigma(i, l_2) - \sigma(i-1, l_1) + \sigma(i-1, l_2)) \tag{5.38}$$

e

$$r_{i2} = \sigma(i, l_3) + Z(\sigma(i, l_1) - \sigma(i, l_3) - \sigma(i-1, l_1) + \sigma(i-1, l_3)). \tag{5.39}$$

Agora, analisando para y_2^T tem-se:

$$y_2^T = C^{\sigma(2,l_2)+Z(\sigma(2,l_1)-\sigma(1,l_1)+\sigma(1,l_2)-\sigma(2,l_2))} \cdot e_{l_2}^T + C^{\sigma(2,l_3)+Z(\sigma(2,l_1)-\sigma(1,l_1)+\sigma(1,l_3)-\sigma(2,l_3))} \cdot e_{l_3}^T$$

$$\implies y_2^T = C^{r_{21}} e_{l_2}^T + C^{r_{22}} e_{l_3}^T, \tag{5.40}$$

onde

$$r_{21} = \sigma(2, l_2) + Z\left(\sigma(2, l_1) - \sigma(1, l_1) + \sigma(1, l_2) - \sigma(2, l_2)\right)$$

e

$$r_{22} = \sigma(2, l_3) + Z\left(\sigma(2, l_1) - \sigma(1, l_1) + \sigma(1, l_3) - \sigma(2, l_3)\right).$$

Fazendo a mesma análise para y_3^T tem-se:

$$y_3^T = C^{\sigma(3,l_2)+Z(\sigma(3,l_1)-\sigma(2,l_1)+\sigma(2,l_2)-\sigma(3,l_2))} \cdot e_{l_2}^T + C^{\sigma(3,l_3)+Z(\sigma(3,l_1)-\sigma(2,l_1)+\sigma(2,l_3)-\sigma(3,l_3))} \cdot e_{l_3}^T$$

$$\implies y_3^T = C^{r_{31}} e_{l_2}^T + C^{r_{32}} e_{l_3}^T, \tag{5.41}$$

sendo

$$r_{31} = \sigma(3, l_2) + Z\left(\sigma(3, l_1) - \sigma(2, l_1) + \sigma(2, l_2) - \sigma(3, l_2)\right)$$

e

$$r_{32} = \sigma(3, l_3) + Z\left(\sigma(3, l_1) - \sigma(2, l_1) + \sigma(2, l_3) - \sigma(3, l_3)\right).$$

Por fim, para y_4^T:

$$y_4^T = C^{\sigma(4,l_2)+Z(\sigma(4,l_1)-\sigma(3,l_1)+\sigma(3,l_2)-\sigma(4,l_2))} \cdot e_{l_2}^T + C^{\sigma(4,l_3)+Z(\sigma(4,l_1)-\sigma(3,l_1)+\sigma(3,l_3)-\sigma(4,l_3))} \cdot e_{l_3}^T$$

$$\implies y_4^T = C^{r_{41}} e_{l_2}^T + C^{r_{42}} e_{l_3}^T, \tag{5.42}$$

onde

$$r_{41} = \sigma(4, l_2) + Z\left(\sigma(4, l_1) - \sigma(3, l_1) + \sigma(3, l_2) - \sigma(4, l_2)\right)$$

e

$$r_{42} = \sigma(4, l_3) + Z\left(\sigma(4, l_1) - \sigma(3, l_1) + \sigma(3, l_3) - \sigma(4, l_3)\right).$$

Isolando $e_{l_2}^T$ na Equação 5.40, tem-se:

$$y_2^T = C^{r_{21}} e_{l_2}^T + C^{r_{22}} e_{l_3}^T \implies y_2^T - C^{r_{22}} e_{l_3}^T = C^{r_{21}} e_{l_2}^T$$

$$\implies e_{l_2}^T = (y_2^T - C^{r_{22}} \cdot e_{l_3}^T) \cdot C^{-r_{21}}. \tag{5.43}$$

Substituindo $e_{l_2}^T$ dado por (5.43) na Equação 5.41 e isolando $e_{l_3}^T$, obtém-se:

$$\begin{aligned} y_3^T &= C^{r_{31}} e_{l_2}^T + C^{r_{32}} e_{l_3}^T = C^{r_{31}} (y_2^T - C^{r_{22}} e_{l_3}^T) \cdot C^{-r_{21}} + C^{r_{32}} e_{l_3}^T \\ &= C^{r_{31}-r_{21}} y_2^T - C^{r_{31}+r_{22}-r_{21}} e_{l_3}^T + C^{r_{32}} e_{l_3}^T \end{aligned}$$

Portanto,

$$\begin{aligned} y_3^T - C^{r_{31}-r_{21}} y_2^T &= (C^{r_{32}} - C^{r_{31}+r_{22}-r_{21}}) \cdot e_{l_3}^T \\ y_3^T - C^{r_{31}-r_{21}} y_2^T &= C^{r_{32}+Z(r_{31}+r_{22}-r_{21}-r_{32})} \cdot e_{l_3}^T \end{aligned}$$

Agora, isolando $e_{l_3}^T$ tem-se

$$e_{l_3}^T = (y_3^T - C^{r_{31}-r_{21}} y_2^T) \cdot C^{-r_{32}-Z(r_{31}+r_{22}-r_{21}-r_{32})}$$

$$\implies e_{l_3}^T = (y_3^T - C^{\overline{r_{31}}} y_2^T) \cdot C^{\overline{r_{32}}}, \tag{5.44}$$

sendo

$$\overline{r_{31}} = r_{31} - r_{21}$$

e

$$\overline{r_{32}} = r_{32} - Z(r_{31} + r_{22} - r_{21} - r_{32}),$$

generalizando, respectivamente,

$$\overline{r_{(i-1)1}} = r_{(i-1)1} - r_{(i-2)2} \tag{5.45}$$

e

$$\overline{r_{(i-1)2}} = r_{(i-1)2} - Z(r_{(i-1)1} + r_{(i-2)2} - r_{(i-2)1} - r_{(i-1)2}). \tag{5.46}$$

Por fim, substituindo (5.43) e (5.44) em (5.42), temos:

$$\begin{aligned}
y_4^T &= C^{r_{41}} e_{l_2}^T + C^{r_{42}} e_{l_3}^T = C^{r_{41}}(y_2^T - C^{r_{22}} e_{l_3}^T) \cdot C^{-r_{21}} + C^{r_{42}}(y_3^T - C^{\overline{r_{31}}} y_2^T) \cdot C^{\overline{r_{32}}} \\
&= C^{r_{41}-r_{21}} y_2^T - C^{r_{41}+r_{22}-r_{21}} e_{l_3}^T + C^{r_{42}+\overline{r_{32}}} y_3^T - C^{r_{42}+\overline{r_{31}}+\overline{r_{32}}} y_2^T \\
&= (C^{r_{41}-r_{21}} - C^{r_{42}+\overline{r_{31}}+\overline{r_{32}}} + C^{r_{41}+r_{22}-r_{21}+\overline{r_{31}}+\overline{r_{32}}}) \cdot y_2^T + (C^{r_{42}+\overline{r_{32}}} - C^{r_{41}+r_{22}-r_{21}+\overline{r_{32}}}) \cdot y_3^T \\
&= C^{r_{41}+r_{22}-r_{21}+\overline{r_{31}}+\overline{r_{32}}+Z(Z(r_{42}+\overline{r_{31}}+\overline{r_{32}}-r_{41}+r_{21})-r_{22}-\overline{r_{31}}-\overline{r_{32}})} y_2^T + C^{r_{42}+\overline{r_{32}}+Z(r_{41}-r_{21}-r_{42}+r_{22})} y_3^T
\end{aligned}$$

$$\implies y_4^T = C^{\overline{r_{41}}} y_2^T + C^{\overline{r_{42}}} y_3^T, \tag{5.47}$$

onde

$$\overline{r_{41}} = r_{41} + r_{22} - r_{21} + \overline{r_{31}} + \overline{r_{32}} + Z(-r_{22} - \overline{r_{31}} - \overline{r_{32}} + Z(r_{42} + \overline{r_{31}} + \overline{r_{32}} - r_{41} + r_{21}))$$

e

$$\overline{r_{42}} = r_{42} + \overline{r_{32}} + Z(r_{41} - r_{21} - r_{42} + r_{22}).$$

De modo análogo podemos obter para todo $i = 4, ..., m$ a relação

$$y_i^T = C^{\overline{r_{i1}}} y_{i-2}^T + C^{\overline{r_{i2}}} y_{i-1}^T, \tag{5.48}$$

que identifica a posição dos 3 erros em l_{i-2}, l_{i-1} e l_i, sendo

$$\overline{r_{i1}} = r_{i2} + \overline{r_{(i-1)2}} + Z(r_{i1} - r_{(i-2)1} - r_{i2} + r_{(i-2)2}) \tag{5.49}$$

e

$$\begin{aligned}
\overline{r_{i2}} &= r_{i1} + r_{(i-2)2} - r_{(i-2)1} + \overline{r_{(i-1)1}} + \overline{r_{(i-1)2}} \\
&\quad + Z(-r_{(i-2)2} - \overline{r_{(i-1)1}} - \overline{r_{(i-1)2}} + Z(r_{i2} + \overline{r_{(i-1)1}} + \overline{r_{(i-1)2}} - r_{i1} + r_{(i-2)1})).
\end{aligned} \tag{5.50}$$

As expressões para $\overline{r_{i1}}$ e $\overline{r_{i2}}$, apresentadas nas Equações 5.49 e 5.50 são altamente complexas

quando se utiliza uma matriz superregular arbitrária A, devido à dependência em múltiplos índices. A seguir serão apresentadas expressões simplificadas para o caso de A ser uma matriz de Vandermonde, motivado pelo Corolário 5.2.1.

Corolário 5.2.2 *Se A for uma matriz de Vandermonde, então $\overline{r_{i1}}$ e $\overline{r_{i2}}$ na Equação 5.48 são dados por*

$$\overline{r_{i1}} = l_3 - Z(l_2 - l_3) + Z(2l_2 - 2l_3) \tag{5.51}$$

e

$$\overline{r_{i2}} = 3l_2 - l_3 - Z(l_2 - l_3) + Z(l_3 - l_2 + Z(l_2 - l_3) + Z(l_3 - l_2 - Z(l_2 - l_3))), \tag{5.52}$$

para $i = 4, \ldots, m$, onde l_2 e l_3 representam as posições do segundo e terceiro erro em rajada, respectivamente.

Demonstração 5.2.3 *Seja $A = [a_{ij}]$ uma matriz superregular de Vandermonde. Dadas $i-2$, $i-1$ e i três linhas consecutivas e l_j e l_k duas colunas quaisquer de A, então, como na demonstração do Corolário 5.2.1, mostra-se que*

$$\begin{aligned} l_k = {} & \sigma(i, l_k) + \sigma(i-1, l_k) + Z(\sigma(i, l_j) - \sigma(i, l_k) - \sigma(i-1, l_j) + \sigma(i-1, l_k)) \\ & + Z(\sigma(i-1, l_j) - \sigma(i-1, l_k) - \sigma(i-2, l_j) + \sigma(i-2, l_k)). \end{aligned} \tag{5.53}$$

Além disso, dadas $i-3$, $i-2$, $i-1$ e i quatro linhas consecutivas e l_j e l_k duas colunas quaisquer de A, então

$$\begin{aligned} 2l_k = {} & \sigma(i, l_k) + \sigma(i-2, l_k) + Z(\sigma(i, l_j) - \sigma(i, l_k) - \sigma(i-1, l_j) + \sigma(i-1, l_k)) \\ & + Z(\sigma(i-2, l_j) - \sigma(i-2, l_k) - \sigma(i-3, l_j) + \sigma(i-3, l_k)), \end{aligned} \tag{5.54}$$

uma vez que os logaritmos de Zech também se cancelam e a diferença entre as potências de α em A nas posições $\sigma(i, l_k)$ e $\sigma(i-2, l_k)$ é igual a $2l_k$, para qualquer coluna l_k. Usando (5.53) e (5.54) em (5.38) e (5.39), obtemos

$$r_{i1} - r_{(i-1)1} = l_2 \quad e \quad r_{i2} - r_{(i-1)2} = l_3, \tag{5.55}$$

e

$$r_{i1} - r_{(i-2)1} = 2l_2 \quad e \quad r_{i2} - r_{(i-2)2} = 2l_3, \tag{5.56}$$

para $i = 4, \ldots, m$.

Agora, usando (5.55) e (5.56) em (5.45) e (5.46), obtemos

$$\overline{r_{(i-1)1}} = l_2 \quad e \quad \overline{r_{(i-1)2}} = l_3 - Z(l_2 - l_3) + Z(2l_2 - 2l_3), \tag{5.57}$$

para $i = 4, \ldots, m$.

Finalmente, usando (5.57) em (5.49) e (5.50), obtemos as Equações 5.51 e 5.52 para $i = 4, \ldots, m$, o que prova o Corolário 5.2.2. Essas expressões dependem apenas das posições das rajadas de erro l_2 e l_3.

Apresenta-se a seguir três exemplos de correção de três rajadas de erro, para $m > k$, $m < k$ e $m = k$, respectivamente. Note que as etapas de decodificação são as mesmas do Algoritmo 5.2.2, porém, para $m \geq 6$, na Etapa 8 será utilizada a Equação 5.48, e as potências da matriz companheira C são obtidas utilizando as Equações 5.51 e 5.52.

Para determinar a magnitude das rajadas de erro, conforme descrito no Passo 9, resolve-se um sistema linear com três equações que envolvem as variáveis $\mathbf{e}_{l_1}^T$, $\mathbf{e}_{l_2}^T$ e $\mathbf{e}_{l_3}^T$.

Exemplo 5.2.7 *Seja o polinômio primitivo $p(x) = x^5 + x^2 + 1 \in \mathbb{F}_2[x]$. Através desse polinômio é possível construir um código matricial MDS de parâmetros $[11, 5, 7]$ sobre $\mathbb{F}_{2^5}$, capaz de corrigir até três rajadas de erro de comprimento $b = 5$, com $m > k$. Através da matriz de Frobenius e da matriz de Vandermonde:*

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad e \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} \\ \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} \\ \alpha^5 & \alpha^{10} & \alpha^{15} & \alpha^{20} & \alpha^{25} \end{bmatrix},$$

é possível obter a matriz verificação de paridade para o código $\mathcal{C}$:

$$H = \left[\begin{array}{ccccc|c} I_5 & I_5 & I_5 & I_5 & I_5 & \\ C & C^2 & C^3 & C^4 & C^5 & \\ C^2 & C^4 & C^6 & C^8 & C^{10} & I_{30} \\ C^3 & C^6 & C^9 & C^{12} & C^{15} & \\ C^4 & C^8 & C^{12} & C^{16} & C^{20} & \\ C^5 & C^{10} & C^{15} & C^{20} & C^{25} & \end{array}\right].$$

Supondo que a palavra recebida foi $v = [01011\ 10010\ 11100\ 00100\ 10001\ 01110\ 00111\ 01101\ 01001\ 01010\ 00001]$. Ao calcular as síndromes dadas em (5.6), tem-se:

$$\boldsymbol{s}_1^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_2^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_3^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_4^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_5^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_6^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Como todas as síndromes são diferentes de zero, segue que os erros ocorreram nos símbolos de informação. Considerando $l_1 = 1$, é calculado $\mathbf{y}_i$, dado na Equação 5.10, para $i = 1, 2, \ldots, 6$,

$$\boldsymbol{y}_1^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_2^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_3^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_4^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_5^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{y}_6^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Note que, neste ponto do algoritmo, deve-se verificar a possibilidade de ocorrência de um segundo erro em rajada combinado com l_1. Seguindo o Algoritmo 5.2.2, supomos um segundo erro em rajada em $l_2 = 2$ e calculado que $r_{i-2} = 2$ para $i = 3, 4, 5, 6$ utilizando a Equação 5.30, é verificado que

$$\boldsymbol{y}_i^T \neq C^2 \cdot \boldsymbol{y}_{i-1}^T,$$

para $i = 3, 4, 5$. Como não há verificação se considerarmos todas as possibilidades para l_2, deve-se assumir que um terceiro erro ocorre em $l_3 = 3$ e encontrar os valores através da Equação 5.48, para $i = 4, 5, 6$. Ao aplicar as Equações 5.51 e 5.52 tem-se:

$$\overline{r_{i1}} = l_3 - Z(l_2 - l_3) + Z(2l_2 - 2l_3) = 3 - Z(-1) + Z(-2) = 3 - 17 + 3 = -11$$

e

$$\begin{aligned} \overline{r_{i2}} &= 3l_2 - l_3 - Z(l_2 - l_3) + Z(l_3 - l_2 + Z(l_2 - l_3) + Z(l_3 - l_2 - Z(l_2 - l_3))) \\ &= 6 - 3 - Z(-1) + Z(1 + Z(-1) + Z(1 - Z(-1))) \\ &= 3 - 17 + Z(18 + Z(-16)) = -14 + 19 = 5, \end{aligned}$$

para $i = 4, 5, 6$. Portanto, a igualdade

$$y_i^T = C^{-11} \cdot y_{i-1}^T + C^5 \cdot y_{i-2}^T,$$

é satisfeita para $i = 4, 5, 6$. Então as posições dos três erros são $l_1 = 1$, $l_2 = 2$ e $l_3 = 3$. Resolvendo o sistema linear com três equações que envolvem as variáveis $\boldsymbol{e}_{l_1}$, $\boldsymbol{e}_{l_2}$ e $\boldsymbol{e}_{l_3}$, é obtido

$$\boldsymbol{e}_1 = [11010], \quad \boldsymbol{e}_2 = [01010] \quad e \quad \boldsymbol{e}_3 = [01110].$$

Portanto, a palavra-código recebida é

$$\boldsymbol{c} = [\mathbf{10001}\ \ \mathbf{11000}\ \ \mathbf{10010}\ \ 00100\ \ 10001\ \ 01110\ \ 00111\ \ 01101\ \ 01001\ \ 01010\ \ 00001].$$

Exemplo 5.2.8 *Considere o polinômio primitivo $p(x) = x^7 + x + 1 \in \mathbb{F}_2[x]$ é possível construir um código matricial MDS de parâmetros $[13, 7, 7]$ sobre $\mathbb{F}_{2^7}$, capaz de corrigir até três rajadas de erro de comprimento $b = 7$, com $m < k$. A matriz superregular de Vandermonde A e a matriz companheira de $p(x)$ são dadas, respectivamente, por*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} & \alpha^{14} \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \alpha^{21} \\ \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} & \alpha^{24} & \alpha^{28} \\ \alpha^5 & \alpha^{10} & \alpha^{15} & \alpha^{20} & \alpha^{25} & \alpha^{30} & \alpha^{35} \end{bmatrix} \quad e \quad C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Então

$$H = \left[\begin{array}{ccccccc|c} I_7 & I_7 & I_7 & I_7 & I_7 & I_7 & I_7 & \\ C & C^2 & C^3 & C^4 & C^5 & C^6 & C^7 & \\ C^2 & C^4 & C^6 & C^8 & C^{10} & C^{12} & C^{14} & I_{42} \\ C^3 & C^6 & C^9 & C^{12} & C^{15} & C^{18} & C^{21} & \\ C^4 & C^8 & C^{12} & C^{16} & C^{20} & C^{24} & C^{28} & \\ C^5 & C^{10} & C^{15} & C^{20} & C^{25} & C^{30} & C^{35} & \end{array}\right]$$

é a matriz verificação de paridade para este código. Assumindo que a palavra recebida é $v = [0010011\ 0100100\ 1010010\ 0011011\ 1101011\ 1111101\ 1000111\ 0000010\ 0001110\ 0001100\ 1000100\ 0010100\ 0010110]$. *Os valores das síndromes são*

$$\boldsymbol{s}_1^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_2^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_3^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_4^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_5^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_6^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Como novamente todas as síndromes são diferentes de zero, segue que os erros ocorreram nos símbolos de informação. Considerando $l_1 = 1$*, é calculado* $\boldsymbol{y}_i$*, dado na Equação 5.10, para* $i = 1, 2, \ldots, 6$ *,*

$$\boldsymbol{y}_1^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_2^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_3^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{y}_4^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_5^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_6^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Verificando a possibilidade de ocorrência de um segundo erro em rajada combinado com l_1*, é calculado que* $r_{i-2} = 2$ *para* $i = 3, 4, 5, 6$ *utilizando a Equação 5.30 tem-se*

$$\boldsymbol{y}_i^T \neq C^2 \cdot \boldsymbol{y}_{i-1}^T,$$

para $i = 3, 4, 5, 6$. Assim, assumindo que um terceiro erro ocorre em $l_3 = 3$ e encontrando os valores através da Equação 5.48, para $i = 4, 5, 6$ e aplicando as Equações 5.51 e 5.52 temos:

$$\overline{r_{i1}} = l_3 - Z(l_2 - l_3) + Z(2l_2 - 2l_3) = 3 - Z(-1) + Z(-2) = 3 - 6 + 12 = 9$$

e

$$\begin{aligned}\overline{r_{i2}} &= 3l_2 - l_3 - Z(l_2 - l_3) + Z(l_3 - l_2 + Z(l_2 - l_3) + Z(l_3 - l_2 - Z(l_2 - l_3))) \\ &= 6 - 3 - Z(-1) + Z(1 + Z(-1) + Z(1 - Z(-1))) = -3 + 8 = 5,\end{aligned}$$

para $i = 4, 5, 6$. Portanto, a igualdade

$$\boldsymbol{y}_i^T = C^9 \boldsymbol{y}_{i-1}^T + C^5 \boldsymbol{y}_{i-2}^T,$$

é satisfeita para $i = 4, 5, 6$. Então as posições dos três erros são $l_1 = 1$, $l_2 = 2$ e $l_3 = 3$. Novamente, resolvendo o sistema linear, é obtido

$$\boldsymbol{e}_1 = [0110011], \quad \boldsymbol{e}_2 = [0011000] \quad e \quad \boldsymbol{e}_3 = [0001000].$$

Por fim, a palavra-código recebida corretamente é $c = [0100000\ 0111100\ 1011010\ 0011011\ 1101011\ 1111101\ 1000111\ 0000010\ 0001110\ 0001100\ 1000100\ 0010100\ 0010110]$.

Exemplo 5.2.9 *Considere o polinômio primitivo $p(x) = x^6 + x^4 + x^3 + x + 1 \in \mathbb{F}_2[x]$ é possível construir um código matricial MDS de parâmetros $[12, 6, 7]$ sobre $\mathbb{F}_{2^6}$, capaz de corrigir até três rajadas de erro de comprimento $b = 6$, com $m = k$. A matriz de Frobenius C, de Vandermonde A e a de paridade H são dadas, respectivamente, por*

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad e \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} \\ \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} \\ \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} & \alpha^{24} \\ \alpha^5 & \alpha^{10} & \alpha^{15} & \alpha^{20} & \alpha^{25} & \alpha^{30} \end{bmatrix}.$$

$$H = \left[\begin{array}{cccccc|c} I_6 & I_6 & I_6 & I_6 & I_6 & I_6 & \\ C & C^2 & C^3 & C^4 & C^5 & C^6 & \\ C^2 & C^4 & C^6 & C^8 & C^{10} & C^{12} & I_{36} \\ C^3 & C^6 & C^9 & C^{12} & C^{15} & C^{18} & \\ C^4 & C^8 & C^{12} & C^{16} & C^{20} & C^{24} & \\ C^5 & C^{10} & C^{15} & C^{20} & C^{25} & C^{30} & \end{array}\right].$$

Supondo que a palavra recebida foi

$\boldsymbol{v} = [011010\ 101011\ 001011\ 100011\ 000101\ 000111\ 100100\ 000001\ 001100\ 000111\ 001010\ 101110]$.

As síndromes são

$$\boldsymbol{s}_1^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_2^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_3^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_4^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_5^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_6^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Como todas as síndromes são diferentes de zero, calculamos $\mathbf{y}_i$ *para* $i = 1, 2, \ldots, 6$*, usando a Equação 5.6, sendo* $l_1 = 1$*,*

$$\boldsymbol{y}_1^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{y}_2^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{y}_3^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_4^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_5^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{y}_6^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Verificando a possibilidade de ocorrência de um segundo erro em rajada combinado com l_1*, é calculado que* $r_{i-2} = 2$ *para* $i = 3, 4, 5, 6$ *e obtido que*

$$\boldsymbol{y}_i^T \neq C^2 \boldsymbol{y}_{i-1}^T,$$

para $i = 3, 4, 5, 6$*. Assumindo que um terceiro erro ocorre em* $l_3 = 3$ *e encontrando os valores através da Equação 5.48, para* $i = 4, 5, 6$ *e aplicando as Equações 5.51 e 5.52 temos:*

$$\overline{r_{i1}} = l_3 - Z(l_2 - l_3) + Z(2l_2 - 2l_3) = 3 - Z(-1) + Z(-2) = 3 - 55 + 47 = -5$$

e

$$\begin{aligned} \overline{r_{i2}} &= 3l_2 - l_3 - Z(l_2 - l_3) + Z(l_3 - l_2 + Z(l_2 - l_3) + Z(l_3 - l_2 - Z(l_2 - l_3))) \\ &= 6 - 3 - Z(-1) + Z(1 + Z(-1) + Z(1 - Z(-1))) \\ &= 3 - 55 + Z(56 + Z(-54)) = -52 + 57 = 5, \end{aligned}$$

para $i = 4, 5, 6$*. Assim, a igualdade*

$$\boldsymbol{y}_i^T = C^{-5} \boldsymbol{y}_{i-1}^T + C^5 \boldsymbol{y}_{i-2}^T,$$

é satisfeita. Então a posição dos três erros são $l_1 = 1$*,* $l_2 = 2$ *e* $l_3 = 3$*. Resolvendo o sistema linear, tem-se*

$$\boldsymbol{e}_1 = [111001], \quad \boldsymbol{e}_2 = [011010] \quad e \quad \boldsymbol{e}_3 = [011100].$$

Portanto, a palavra código corrigida é

$$\boldsymbol{c} = [\mathbf{100011\ 110001\ 010111}\ 100011000101\ 000111\ 100100\ 000001\ 001100\ 000111\ 001010\ 101110].$$

6 CONCLUSÃO

O presente Trabalho de Conclusão de Curso apresentou um estudo sobre os principais conceitos relacionados às matrizes redundantes de discos independentes, destacando as vantagens e desvantagens de cada nível dos RAIDs e demonstrando suas estratégias de paridade, com foco na codificação e decodificação do RAID 6. Esse nível de RAID possui sempre $m = 2$ discos de paridade distribuídos e é capaz de corrigir dois erros em rajada quando suas localizações são conhecidas.

Em seguida, foram abordados os códigos corretores de erros, com ênfase na construção de códigos matriciais MDS baseados nas matrizes superregulares de Vandermonde e de Cauchy. Foram apresentados dois algoritmos de decodificação para códigos matriciais MDS sobre $\mathbb{F}_q^b$: um capaz de corrigir até um erro em rajada para códigos com parâmetros $[m + k, k, m + 1]$, para todo $m \geq 2$. Outro capaz de corrigir até dois erros em rajada para códigos com parâmetros $[m + k, k, m + 1]$, para todo $m \geq 4$, considerando as condições $m > k$, $m < k$ e $m = k$, onde b representa o comprimento da rajada de erro. Uma importante contribuição deste trabalho, foi apresentar uma simplificação do algoritmo de correção de dois erros em rajada para o caso em que o código matricial MDS é construído a partir de uma matriz superregular de Vandermonde. Além disso, também foi apresentada uma expansão na capacidade de correção de erros, analisando o comportamento do algoritmo para o caso de três erros ocorrendo nos símbolos de informação. Foi demonstrado que, para três erros, a decodificação também é simplificada quando utilizada uma matriz superregular de Vandermonde na construção do código matricial MDS. Observou-se, ainda, que para o caso em que todos os erros ocorrem nos símbolos de informação, conforme a capacidade de correção aumenta, a complexidade do algoritmo também cresce devido à correlação entre as posições dos erros. Entretanto, acredita-se que um procedimento análogo pode ser aplicado para determinar suas localizações para até $\lfloor \frac{d-1}{2} \rfloor$ erros em rajada. Nos demais casos, isto é, quando os erros ocorrem nos símbolos de paridade, as etapas do algoritmo para correção de l erros em rajada seguem um comportamento semelhante ao do caso de correção de $l - 1$ erros em rajada.

Ao comparar os processos de codificação e decodificação dos códigos matriciais MDS com os do RAID 6, observa-se uma maior complexidade de correção de erros nos códigos MDS, mesmo na condição $m < k$. No entanto, sua principal vantagem é a capacidade de corrigir dois erros em localizações desconhecidas, enquanto no RAID 6 é necessário conhecer a posição dos erros para que a correção seja possível.

Um algoritmo para correção de dois erros em rajada foi implementado em linguagem Python, e exemplos foram gerados para todos os casos previstos. Como continuidade deste trabalho, espera-se aprimorar as simulações computacionais e projetar esta codificação e decodificação em um sistema de armazenamento sujeito a falhas, de modo a analisar a eficiência do algoritmo em aplicações reais de sistemas de armazenamento distribuído e diversos tipos de canais. Além disso, seria interessante analisar a viabilidade do código MDS e outras simplificações para outras famílias de matrizes superregulares, ou outras estratégias de codificação de códigos matriciais MDS.

REFERÊNCIAS

ARPACI-DUSSEAU, R. H. Redundant Arrays of Inexpensive Disks (Raids). **OPERATING SYSTEMS [VERSION 1.01]**, 2008.

BENEDITO, C. W. de O. Famílias de Reticulados Algébricos e Reticulados Ideais. **Universidade Estadual Paulista**, São José do Rio Preto, São Paulo - Brasil, 2010.

BLAUM, M.; FARRELL, P.; TILBORG, H. V. **Handbook of Coding Theory**. 1. ed. [*S.l.*]: Elsevier Science B.V, 1998.

BLAUM, M.; HAFNER, J. L.; HETZLER, S. Partial-MDS Codes and Their Application to Raid Type of Architectures. **IEEE Transactions on Information Theory**, v. 59, n. 7, p. 4510–4519, 2013.

CARDELL, S. D.; CLIMENT, J.; REQUENA, V. **A Construction of MDS Array Codes**. [*S.l.*]: [*s.n.*], 2013. v. 45. P. 47–58. ISBN 781845647087.

CORPORATION, I. Intelligent Raid 6 Theory Overview and Implementation. **Intel Storage Building Blocks**, 2005. Disponível em: <https://www.lamsade.dauphine.fr/~litwin/cours98/Doc-cours-clouds/raid6-intel.pdf>.

DOMINGUES, H. H.; IEZZI, G. **Álgebra Moderna**. 4. ed. São Paulo: [*s.n.*], 1982.

FANG; WEIJUN; LV. New Constructions of MDS Array Codes and Optimal Locally Repairable Array Codes. **IEEE Transactions on Information Theory**, v. 70, n. 3, p. 1806–1822, 2024.

FIEDLER, M. Notes on Hilbert and Cauchy Matrices. **Linear Algebra and Its Applications - LINEAR ALGEBRA APPL**, 2010.

FIRER, M. **Códigos Corretores de Erros - Notas de Aulas**. 2017. Disponível em: <https://www.ime.unicamp.br/~mfirer/3NotasFoz2006.pdf>.

FRITSCH, A. R. Overlapping Error Correction Codes on Two Dimensional Structures. **Pontifícia Universidade Católica do Rio Grande do Sul**, 2025.

GUIMARÃES, W. P. S. Códigos Corretores de Erros para Gravação Magnética. **Universidade Federal de Pernambuco**, 2003.

HEFEZ, A.; VILLELA, M. L. T. **Códigos Corretores de Erros**. 2. ed. [*S.l.*]: Instituto Nacional de Matemática Pura e Aplicada - IMPA, 2008.

HSIEH, P.-H.; CHEN, I.-Y.; LIN, Y.-T. **An XOR Based Reed-Solomon Algorithm for Advanced RAID Systems**. [*S.l.*]: [*s.n.*], 2004. 165-172 p.

HUBER, K. Some Comments on Zech's Logarithms. **EEE Transactions on Information Theory**, v. 36, 1990.

HUFFMAN, W. C.; PLESS, V. **Fundamentals of Error Correting Codes**. 1. ed. [*S.l.*]: Cambridge University Press, 2003.

JACOB, D. Efficient Implementation of Raid-6 Encoding and Decoding on a Field Programmable Gate Array (FPGA). **VCU Scholars Compass**, 2009. Disponível em: <https://scholarscompass.vcu.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=2980&context=etd>.

MILIES, C. P. **Breve Introdução à Teoria dos Códigos Corretores de Erros**. 2009. Disponível em: <https://shre.ink/aoNi>.

MOUSSA, M.; RYCHLIK, M. Beyond Raid 6 - an Efficient Systematic Code Protecting Against Multiple Errors, Erasures, and Silent Data Corruption. **ArXiv**, abs/1806.08266, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:49336065>.

PRESTES, G. C. K. Análise da Utilização de Códigos de Paridade em Modelos de Raid. **Universidade Estadual Paulista Júlio Mesquita Filho**, São João da Boa Vista, São Paulo - Brasil, 2023.

RAYN, W.; LIN, S. **Channel Codes: Classical and Modern**. 1. ed. [*S.l.*]: Cambridge University Press, 2009. ISBN 978-0-511-64182-4.

ROMAN, S. **Coding and Information Theory**. 1. ed. [*S.l.*]: Springer-Verlag, 1992.

ROTH. **Introduction to Coding Theory.** 1. ed. [*S.l.*]: Cambridge University Press, 2006.

YE, M.; BARG, A. Explicit Constructions of High-Rate MDS Array Codes with Optimal Repair Bandwidth. **IEEE Transactions on Information Theory**, v. 63, p. 2001–2014, 04 2017.

YE, M.; BARG, A. Cooperative repair: Constructions of Optimal MDS Codes for All Admissible Parameters. **IEEE Transactions on Information Theory**, v. 65, n. 3, p. 1639–1656, 2019.

ZANITTI, D. B. C. Codificação e Decodificação de Códigos Matriciais MDS via Matrizes Superregulares para Correção de Erros em Rajada. **Universidade Estadual Paulista Júlio Mesquita Filho**, São João da Boa Vista, São Paulo - Brasil, 2021.