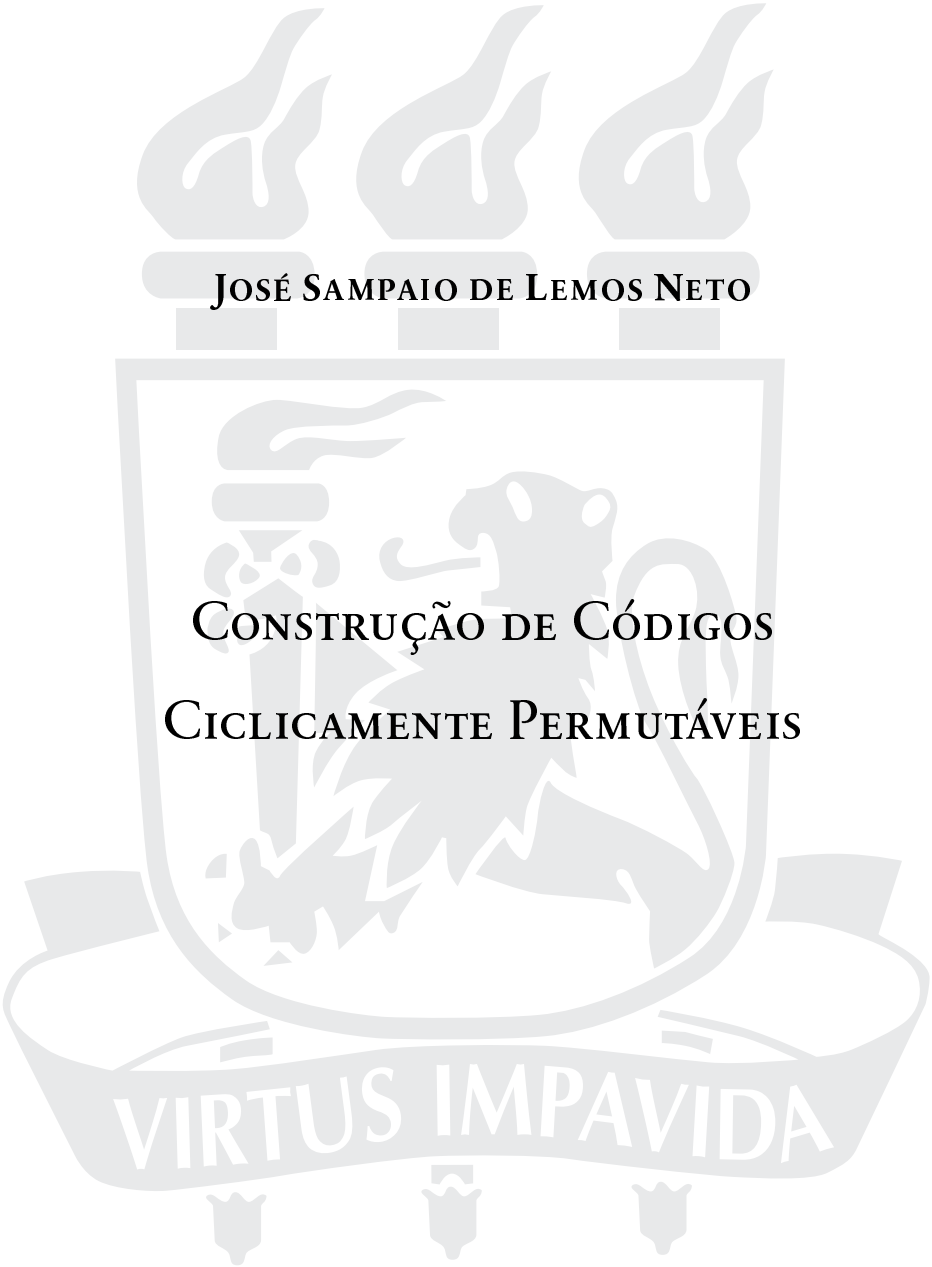
Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Programa de Pós-graduação em Engenharia Elétrica



José Sampaio de Lemos Neto

Construção de Códigos Ciclicamente Permutáveis

Recife, Fevereiro de 2015.

José Sampaio de Lemos Neto

Construção de Códigos Ciclicamente Permutáveis

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de **Doutor em Engenharia Elétrica**

Orientador: Prof. Valdemar Cardoso da Rocha Júnior, Ph.D.

Recife
2015

Catalogação na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

L557c Lemos Neto, José Sampaio de.
Construção de códigos ciclicamente permutáveis / José Sampaio de Lemos Neto. - Recife: O Autor, 2015.
86 folhas, il., gráfs.

Orientador: Prof. Dr. Valdemar Cardoso da Rocha Júnior.
Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2015.
Inclui Referências.

1. Engenharia Elétrica. 2. Códigos corretores de erro. 3. Códigos de bloco. 4. Códigos cíclicos. 5. Códigos constacíclicos. 6. Códigos ciclicamente permutáveis. I. Rocha Júnior, Valdemar Cardoso da. (Orientador). II. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2015-165

Universidade Federal de Pernambuco
Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE TESE DE DOUTORADO

JOSÉ SAMPAIO DE LEMOS NETO

TÍTULO

"CONSTRUÇÃO DE CÓDIGOS CICLICAMENTE PERMUTÁVEIS"

A comissão examinadora composta pelos professores: VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE; RICARDO MENEZES CAMPELLO DE SOUZA, DES/UFPE; CECILIO JOSÉ LINS PIMENTEL, DES/UFPE; MARIA DE LOURDES MELO GUEDES ALCOFORADO, POLI/UPE e DANIEL PEDRO BEZERRA CHAVES, DES/UFPE sob a presidência do primeiro, consideram o candidato **JOSÉ SAMPAIO DE LEMOS NETO APROVADO.**

Recife, 23 de fevereiro de 2015.

CECILIO JOSÉ LINS PIMENTEL
Coordenador do PPGEE

VALDEMAR CARDOSO DA ROCHA JÚNIOR
Orientador e Membro Titular Interno

MARIA DE LOURDES MELO GUEDES ALCOFORADO
Membro Titular Externo

RICARDO MENEZES CAMPELLO DE SOUZA
Membro Titular Interno

DANIEL PEDRO BEZERRA CHAVES
Membro Titular Externo

CECILIO JOSÉ LINS PIMENTEL
Membro Titular Interno

Aos meus pais,

José Sampaio Filho e

Maria de Lourdes

Agradecimentos

Em primeiro lugar, a Deus por sempre iluminar meus pensamentos e, desta forma, permitir que eu sempre supere as dificuldades que encontro no decorrer da minha vida, guiando-me pelos caminhos dos quais Ele julga-me ser merecedor de trilhar.

Aos meus pais, José Sampaio Filho e Maria de Lourdes, pois o amor e o apoio incondicional deles motivam-me a sempre lutar para realizar meus sonhos. Qualquer que seja o patamar profissional e pessoal que eu alcance, sempre lembrarei que só foi possível alcançá-lo porque eles sempre estiveram ao meu lado.

Aos professores do grupo de Comunicações: Ricardo Campello, Márcia Mahon, Cecílio Pimentel e Hélio Magalhães pelas disciplinas ministradas durante a graduação, mestrado e doutorado, além dos exemplos de competência e dedicação pessoal e profissional. Em especial, ao meu orientador, Prof. Valdemar da Rocha, por sua dedicação, incentivo, amizade e, principalmente, por ter acreditado no meu potencial para realizar vários trabalhos de pesquisa com sua parceria, incluindo esta tese. Além do mais, por ser um exemplo de professor e pesquisador, o qual pretendo seguir.

Aos inesquecíveis amigos de graduação: Frederico Basto, Raffaello Bruno, Júlio Jansen, Jairo Amaral, Carolina Bastos, Alinson Clementino, Daniel Façanha, Igor Gouveia, Roberto Cássio e Bergson José. Aos amigos de mestrado: Caio Marcelo, Paulo Freitas, Paulo Martins, Maurício Cordeiro e Daniel Simões; assim como aos amigos de mestrado que continuaram a jornada no doutorado: Marilú Gomes e Paulo Hugo. À Profa. Danielle Camara pela amizade e parceria em nossos trabalhos de pesquisa. Aos amigos Bezerrenses de todas as horas, Romero Gomes e Danilo Pereira. A todos os citados, agradeço muito pela amizade, companheirismo, incentivo, boas conversas, conselhos e confiança.

Por fim, a Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) e ao Programa de Pós-Graduação em Engenharia Elétrica pelo apoio.

José Sampaio de Lemos Neto

Universidade Federal de Pernambuco

23 de Fevereiro de 2015

Não se pode ensinar tudo a alguém, pode-se, apenas,
ajudá-lo a encontrar por si mesmo.

— Galileu Galilei

RESUMO

Um código ciclicamente permutável (código CP) é um código de bloco binário cujas palavras-código são ciclicamente distintas e possuem ordem cíclica plena, isto é, ordem cíclica igual ao comprimento do bloco. Um código CP pode ser construído por meio de um código cíclico. Para isto, selecionam-se as palavras do código cíclico que são ciclicamente distintas e possuem ordem cíclica plena. Um procedimento que seleciona diretamente, por meio de uma condição matemática, as palavras de um código CP a partir de um código cíclico é denominado de *construção*. Sendo M e n, respectivamente, o número de palavras e o comprimento do bloco de um código cíclico, se o número de palavras do código CP for igual ao limitante superior M/n, então a construção é ótima neste sentido. Além do mais, a distância mínima do código cíclico deve ser a maior possível para os valores de M e n. Nesta tese, é proposto um método para construir códigos CP por meio de códigos lineares cíclicos q-ários, sendo q uma potência de um número primo, assim como também por meio de códigos lineares constacíclicos p-ários, sendo p um número primo. Para ambos os casos, mostra-se que o procedimento proposto para gerar códigos CP é direto, logo pode ser qualificado como construção. Além do mais, em ambos os casos, a construção é ótima pois atinge o limitante superior. Por fim, uma construção proposta nesta tese é usada na aplicação de códigos CP como sequências de protocolo para o canal de colisão sem realimentação.

Palavras-chaves: Códigos corretores de erro. Códigos de bloco. Códigos cíclicos. Códigos constacíclicos. Códigos ciclicamente permutáveis.

ABSTRACT

A cyclically permutable code (CPC) is a binary code the codewords of which are cyclically distinct and have full cyclic order, i.e., cyclic order equal to the block length. A CPC can be constructed by means of a cyclic code. In this way, the codewords of the cyclic code which are cyclically distinct and have full cyclic order should be selected. A procedure that selects codewords of a CPC from a cyclic code in a straightforward manner, by means of a mathematical condition, is called a *construction*. Let M and n be, respectively, the number of codewords and the block length of a cyclic code. If the number of codewords of a CPC reaches the upper bound M/n, then this construction is optimum in this sense. Furthermore, the minimum distance of the cyclic code should be the highest possible for the values of M and n. In this thesis we propose a method to construct CPC's using q-ary linear cyclic codes, where q is a power of a prime, as well as using p-ary linear constacyclic codes, where p is a prime number. In both cases, it is shown that the proposed procedure to generate CPC's is straightforward, so can be qualified as a construction. Moreover, in both cases, the construction is optimal in the sense that the number of codewords selected for the CPC reaches the upper bound. Finally, a construction proposed in this thesis is used in the application of CPC's as protocol sequences for the collision channel without feedback.

Keywords: Error correction codes. Block codes. Cyclic codes. Constacyclic codes. Cyclically permutable codes.

Lista de Figuras

1.1 Diagrama de blocos de um típico sistema de comunicação digital. 14

2.1 Processo de codificação dos códigos de bloco. 24

Lista de Tabelas

2.1 *Classes conjugadas e polinômios mínimos sobre* $\mathrm{GF}(2)$ *para* $x^{15}+1$ 34

3.1 *Classes conjugadas e polinômios mínimos sobre* $\mathrm{GF}(5)$ *para* x^6-3 41
3.2 *Deslocamentos constacíclicos de* $g(x)=4+2x^2+x^4 \leftrightarrow \mathbf{g}=(4,0,2,0,1,0)$. . . . 43
3.3 *Palavras não-nulas do código* $(6,2,3)$ *gerado por* $g(x)=4+2x^2+x^4$ *(Exemplo 3.3) . A primeira coluna corresponde à quantidade de deslocamentos constacíclicos para direita.* 45

4.1 *Classes de equivalência cíclica para as palavras do código CP do Exemplo 4.1.* 51
4.2 *Correspondência entre os elementos do arranjo* $A_{3\times 3}$ *e os elementos da 9-upla* $\mathbf{b}$. . . . 61
4.3 *representação-*$\mathbf{V}$ *para o elementos de* $\mathrm{GF}(7)$. 64

5.1 *Parâmetros de comparação para as sequências de protocolo. Sequências-Constacíclicas com* $p\geq 5$, $4\leq k\leq p-1$ *e* $w(\mathbf{v}')\geq 3$. *Sequências-RS e Sequências-BCH com* $p\geq 5$, $3\leq k\leq p-1$ *e* $r>1$. 76

Sumário

1 Introdução 13
1.1 Sistemas de Comunicação Digital . . . 13
1.1.1 Sistemas de Comunicação Ponto-a-Ponto . . . 13
1.2 Códigos Ciclicamente Permutáveis . . . 16
1.3 Motivação . . . 18
1.4 Objetivos . . . 19
1.5 Contribuições da Tese . . . 19
1.6 Organização da Tese . . . 20

2 Códigos Corretores de Erro 22
2.1 Códigos de Bloco . . . 23
2.2 Códigos de Bloco Lineares . . . 27
2.2.1 Matriz Geradora e Matriz de Verificação de Paridade . . . 28
2.3 Códigos Cíclicos . . . 31
2.3.1 Matriz Geradora e Matriz de Verificação de Paridade . . . 34
2.3.2 Códigos BCH . . . 36
2.3.3 Códigos Reed-Solomon . . . 37

3 Códigos Constacíclicos 39
3.1 Códigos Constacíclicos . . . 39
3.1.1 Códigos Constacíclicos de Comprimento $p+1$ . . . 40
3.1.2 Ordem Constacíclica das Palavras-Código . . . 42
3.1.3 Classes de Equivalência Constacíclica . . . 48

4 Construção de Códigos Ciclicamente Permutáveis 49
4.1 Códigos Ciclicamente Permutáveis . . . 49
4.2 Códigos CP construídos por meio de Códigos Cíclicos . . . 52
4.3 Códigos CP construídos por meio de Códigos Constacíclicos . . . 55
4.3.1 Mapeamento de códigos constacíclicos p-ários para binário . . . 59

5 Aplicação: Sequências de Protocolo para o Canal de Colisão sem Realimentação 68
5.1 O canal de Colisão sem Realimentação (CCsR) 68
5.1.1 Um caso particular 69
5.2 Sequências de Protocolo 70
5.2.1 Sequências-BCH e Sequências-RS 73
5.2.2 Sequências-Constacíclicas 74
5.3 Comparação das Sequências de Protocolo 76
5.3.1 Análise dos Parâmetros das Sequências 76

6 Considerações Finais e Contribuições 79
6.1 Sugestões para Trabalhos Futuros 79
6.2 Publicações 80

Referências 82

Capítulo I

Introdução

O único lugar onde o sucesso vem antes do trabalho é no dicionário.

— Albert Einstein

Este capítulo tem por objetivo apresentar alguns conceitos que são utilizados ao longo desta tese. Inicialmente, são abordados sistemas de comunicação digital que envolvem um emissor e um destinatário. Posteriormente, apresentam-se a motivação, o objetivo do trabalho proposto para esta tese e um resumo das contribuições. Por fim, é dada uma breve descrição do conteúdo dos capítulos posteriores.

1.1 Sistemas de Comunicação Digital

1.1.1 Sistemas de Comunicação Ponto-a-Ponto

Um sistema de comunicação digital tem por objetivo transportar os dados de uma fonte de informação até um destino. O sistema é denominado *digital* pelo fato de que a informação é representada por meio de um alfabeto finito ou infinito contável de símbolos, sendo esta a diferença básica com relação a sistemas de comunicação analógicos, em que as mensagens são representadas por um alfabeto cujos símbolos variam continuamente em um certo intervalo [1]. Desde a publicação dos trabalhos de Shannon [2], [3] e Hamming [4], e mais

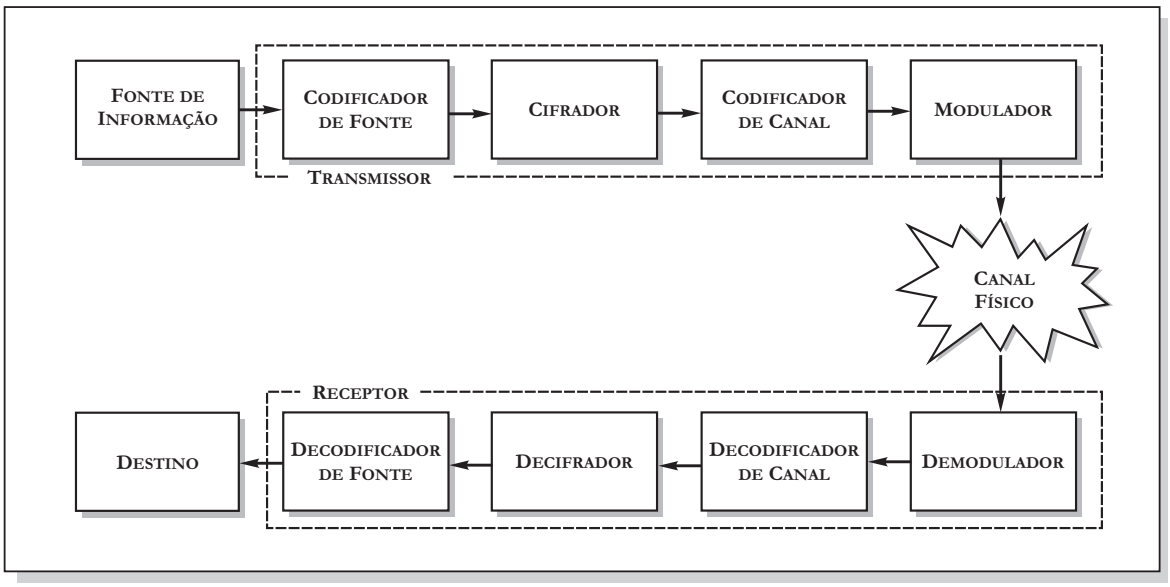


Figura 1.1: *Diagrama de blocos de um típico sistema de comunicação digital.*

recentemente devido à evolução da tecnologia de circuitos integrados em larga escala de integração, técnicas digitais têm substituído técnicas analógicas em sistemas de comunicação. Ao utilizar a informação em formato digital, habilita-se o uso de técnicas poderosas de processamento digital de sinais, incluindo o uso da codificação de fonte, da cifragem da informação a ser transmitida e dos códigos corretores de erro. A Figura 1.1 mostra, por meio de um diagrama de blocos, um típico sistema de comunicação digital[1]. A seguir, é dada uma breve descrição das funções desempenhadas por cada um dos blocos da Figura 1.1.

A *fonte de informação* gera a informação a ser transmitida seja ela dados, voz ou vídeo. Por exemplo, o sistema de telefonia móvel é um sistema de comunicação digital cujo principal objetivo é transmitir sinais de voz. O bloco que representa a fonte de informação, neste caso, é composto (1) pelo ser humano que gera a mensagem a ser transmitida, (2) pelo microfone que converte a voz em sinais elétricos (transdutor), e (3) por um conversor analógico-digital que converte a informação para o formato digital (os dois últimos, componentes de um aparelho móvel digital). Como as fontes de informação digital são representadas por um alfabeto finito, elas podem ser caracterizadas pela distribuição de probabilidade dos símbolos deste alfabeto e, portanto, a informação produzida pode ser medida por meio da *entropia da fonte* [5]. Por fim, a sequência de símbolos é emitida pela fonte a uma taxa média de R_s símbolos por

[1] É comum o uso dos blocos Cifrador/Decifrador em aplicações que exigem uma segurança quanto ao conteúdo da informação transmitida. Em geral, eles são omitidos nos diagramas.

segundo.

O *codificador de fonte* é utilizado para remover a redundância não-controlada que é naturalmente produzida pela fonte de informação. Além do mais, a codificação utilizada deve permitir que o destinatário seja capaz de recuperar a mensagem original sem ambiguidade, trabalho a ser realizado pelo *decodificador de fonte*. Para mais detalhes sobre este assunto, recomenda-se as referências [5]–[7].

O bloco *cifrador* da Figura 1.1 criptografa a informação transmitida de modo que só o destinatário seja capaz de entendê-la. Uma introdução a este assunto pode ser encontrada na referência [8].

O *codificador de canal* tem como objetivo inserir redundância controlada na sequência de informação, proveniente dos blocos anteriores, de tal forma que ao chegar no receptor, o *decodificador de canal* seja capaz de detectar e possivelmente corrigir erros que surjam durante a transmissão. Em outras palavras, a informação transmitida torna-se mais imune aos efeitos do ruído oriundo do *canal físico* (ar, fibra ótica, fio metálico, etc.). Os códigos utilizados pelo codificador de canal são conhecidos como *códigos corretores de erro*. Eles são abordados no Capítulo 2 e podem ser consultados nas referências [9]–[14].

O *modulador* tem a função de mapear os símbolos discretos emitidos pelo codificador de canal em formas de onda apropriadas para transmissão por um canal físico. Existem vários tipos de modulação digital que oferecem diferentes desempenhos ao sistema [15], [16] e cuja escolha depende da aplicação. Na maioria dos casos, porém, a escolha da modulação para um sistema de comunicação digital é limitada por questões de economia de energia ou da disponibilidade da largura de banda. Por exemplo, em um sistema de comunicação via satélite, a modulação escolhida visa minimizar a energia utilizada pelos receptores-transmissores localizados no satélite, ao passo que em um sistema de telefonia móvel, a modulação escolhida visa minimizar a largura de faixa utilizada por cada usuário [10].

Para finalizar a descrição do modelo do sistema de comunicação digital da Figura 1.1, destaca-se que o projeto do *demodulador* e do *decodificador de canal* é, em geral, mais complexo que o projeto dos respectivos, *modulador* e *codificador de canal*. Projetos de demoduladores podem ser encontrados em [15], [16], enquanto decodificadores de canal podem ser encontrados em [9]–[14]. Existem também projetos que contemplam uma junção entre os blocos *codificador de canal* e *modulador*, e os respectivos *decodificador de canal* e *demodulador* (por exemplo, sistemas TCM), os quais podem ser encontrados em [16].

1.2 Códigos Ciclicamente Permutáveis

Gilbert [17] definiu um *código ciclicamente permutável* (código CP) como um código de bloco binário cujas palavras-código são ciclicamente distintas e possuem ordem cíclica plena, isto é, ordem cíclica igual ao comprimento do bloco. O objetivo dele era resolver um problema de comunicação de um sistema de múltiplo acesso assíncrono, em que os usuários (bases) utilizam uma única frequência de rádio e cada usuário é identificado por uma sequência de pulsos distinta. Neste caso, o problema de encontrar uma lista de sequências de pulsos adequada para a aplicação em questão, era similar a um problema particular na área de códigos corretores de erro. Desta forma, as sequências de pulsos são determinadas por meio das palavras de um código CP binário.

Um código CP pode ser construído por meio de um código cíclico. Para isto, selecionam-se as palavras do código cíclico que são ciclicamente distintas e possuem ordem cíclica plena. Dado um código cíclico, o conjunto formado por uma palavra-código e seus deslocamentos cíclicos é denominado de *classe de equivalência cíclica*. Desta forma, o procedimento para construir um código CP por meio de um código cíclico, equivale a particionar o conjunto de palavras do código cíclico em classes de equivalência cíclica, e selecionar as classes que são constituídas por palavras-código de ordem cíclica plena. Em [18], A, Györfi e Massey propuseram a geração de códigos CP binários de peso constante por meio de códigos cíclicos não-binários, Reed-Solomon (RS) [9] ou Berlekamp-Justesen (BJ) [19], cujas coordenadas das palavras-códigos são elementos em GF(q) [20], em que q denota a potência de um número primo. Györfi e Vajda [21] ampliaram a gama de códigos CP binários de peso constante utilizando a mesma proposta dada em [18], porém utilizando códigos BCH [9]. Tanto em [18] como em [21], para realizar um mapeamento um-a-um entre as palavras selecionadas de um código cíclico não-binário e as palavras de um código CP binário de peso constante, é necessário utilizar uma representação cíclica adequada para os elementos de GF(q), assim como uma relação apropriada entre arranjos bidimensionais e vetores. Em [18], [21], é discutida a aplicação dos códigos CP binários como sequências de protocolo para o canal de colisão sem realimentação (CCsR) [22], [23].

Bitan e Etzion [24] também investigaram a construção de códigos CP de peso constante, porém utilizando um método diferente daquele proposto em [18]. Para Bitan e Etzion [24], se $\mathcal{C}_1$ é um código CP de peso constante, então $\mathcal{C}_1$ é dito ser ótimo caso não exista nenhum outro código CP de peso constante $\mathcal{C}_2$, com os mesmos parâmetros de $\mathcal{C}_1$, tal que o número de

palavras-código de $\mathcal{C}_2$ seja maior que o número de palavras-código de $\mathcal{C}_1$. De acordo com [24], códigos CP possuem várias aplicações, além da aplicação como sequências de protocolo para o CCsR citada anteriormente. Outras aplicações incluem: sistemas de comunicação óptico de múltiplo acesso por código [25], espalhamento espectral por saltos de frequência e projetos de radares e sonares [26]. Outra aplicação é apresentada em [27], em que códigos CP não-binários são aplicados em sistemas DS-CDMA (*Direct Sequence Code Division Multiple Access*) com bases assíncronas.

Xia e Fu [28] investigaram a construção de códigos CP por meio de códigos lineares cíclicos q-ários com comprimento de bloco n. Em [28] é dada uma fórmula que permite calcular o número exato de classes de equivalência cíclica com palavras-código de ordem cíclica n, ou seja, o número máximo de palavras de um código CP que podem ser obtidas a partir de um código cíclico. Entretanto, segundo [28], é difícil obter um procedimento que gere diretamente todas as palavras de um código CP a partir de um código cíclico. Ainda em [28], são apresentadas algumas construções algébricas de códigos CP. Em especial, códigos CP binários de peso constante são construídos a partir de códigos BJ generalizados, de acordo com o procedimento dado em [18], e é discutida sua aplicação como sequências de protocolo para o CCsR.

Pode-se afirmar que em [29], o problema de encontrar um procedimento que gere diretamente todas as palavras de um código CP a partir de um código cíclico, de acordo com [28], é resolvido para uma classe específica de códigos cíclicos. Kuribayashi e Tanaka [29] propuseram um método algébrico e sistemático que permite particionar o conjunto de palavras de um código cíclico em classes de equivalência cíclica, deste modo, gerando de forma eficiente um código CP por meio de um código cíclico. Entretanto, o método é restrito a códigos lineares cíclicos binários de comprimento $n = 2^m - 1$, em que n é um número primo de Mersenne [30, pág. 225]. Além do mais, o polinômio gerador, $g(x)$, do código cíclico não pode conter o polinômio $x - 1$ como um de seus fatores. Em [29], destaca-se a aplicação de códigos CP em sistemas de marca d'água digital [31].

Trabalhos mais recentes sobre códigos CP são encontrados em [32]–[35]. Em [32]–[34], os códigos CP são construídos utilizando a ideia proposta em [18]. Porém, uma classe de códigos que ainda não havia sido explorada nesse contexto foi utilizada, a classe de códigos lineares constacíclicos [12]. Embora a ideia seja a mesma usada em [18], para se construir códigos CP binários por meio de códigos lineares constacíclicos p-ários, faz-se necessário utilizar uma

representação cíclica adequada para os elementos de GF(p) diferente da usada em [18], assim como uma relação apropriada entre arranjos bidimensionais e vetores, também diferente da usada em [18]. A depender da representação usada para os elementos de GF(p), as palavras dos códigos CP binários apresentados em [32]–[34] podem ser de peso constante ou de peso variável, um diferencial em relação às construções apresentadas em [18] e [21], por exemplo. Em [36], é discutida a aplicação dos códigos CP apresentados em [33], [34] como sequências de protocolo para o CCsR.

Outros trabalhos abordando códigos CP podem ser encontrados nas referências [37]–[41].

1.3 Motivação

Baseado em [18], um procedimento que seleciona diretamente, por meio de uma condição matemática, as palavras de um código CP a partir de um código cíclico é denominado de *construção*. Sendo M e n, respectivamente, o número de palavras e o comprimento do bloco de um código cíclico, se o número de palavras do código CP for igual ao limitante superior M/n, então a construção é ótima neste sentido. Além do mais, a distância mínima do código cíclico deve ser a maior possível para os valores de M e n.

Conforme mencionado na Seção 1.2, dado um código cíclico q-ário $\mathcal{C}$, é difícil encontrar um procedimento geral que permita selecionar diretamente todas as palavras não-nulas de $\mathcal{C}$ que são ciclicamente distintas e que possuem ordem cíclica plena [28]. Segundo [27], dado um código cíclico q-ário $\mathcal{C}$, não há na literatura uma solução geral para o problema descrito até o presente momento.

Novamente, conforme mencionado na Seção 1.2, o problema foi resolvido para a classe de códigos lineares cíclicos binários com comprimento de bloco $n = 2^m - 1$, em que n é um primo de Mersenne, e desde que o polinômio $x - 1$ não seja um fator do polinômio gerador $g(x)$ [29]. As palavras do código CP são obtidas diretamente por meio de uma expressão que seleciona todas as palavras não-nulas e com ordem cíclica plena do código linear cíclico binário. Entretanto, no caso em que $x - 1$ é um fator de $g(x)$, pelo menos uma palavra do código CP não é selecionada usando o procedimento em [29]. Além do mais, o procedimento estabelecido em [29] não inclui muitos outros casos de interesse, por exemplo, códigos lineares cíclicos binários cujo comprimento de bloco não é necessariamente um número primo de Mersenne, assim como códigos lineares cíclicos não-binários (RS e BCH) e códigos lineares constacíclicos, que podem ser efetivamente mapeados para binário [33].

1.4 Objetivos

Nesta tese é proposto um procedimento que permite selecionar diretamente as palavras de um código CP a partir de um código linear cíclico q-ário, assim como para um código linear constacíclico p-ário, de modo que tal procedimento possa ser qualificado como uma construção. O procedimento apresentado nesta tese baseia-se no método apresentado em [29], porém os resultados incluem outras classes de códigos além da que é utilizada em [29], desta forma, ampliando a gama de escolhas para códigos CP.

1.5 Contribuições da Tese

- O Teorema 2.8 foi publicado originalmente em [42]. Ele estabelece uma condição para o conjunto de raízes do polinômio gerador $g(x)$ de um código linear cíclico q-ário tal que todas as palavras-código não-nulas tenham ordem cíclica plena;
- O Teorema 3.1 estabelece uma condição para o conjunto de raízes do polinômio gerador $g(x)$ de um código linear constacíclico p-ário, de comprimento de bloco $p+1$, tal que todas as palavras-código não-nulas tenham ordem constacíclica plena. Não há na literatura, que seja do conhecimento do autor desta tese, resultado semelhante publicado;
- O Teorema 4.1 foi publicado originalmente em [42]. Por meio dele, é possível obter diretamente as palavras de um código CP a partir de um código linear cíclico q-ário cujo polinômio gerador satisfaz o Teorema 2.8. Tal procedimento pode ser qualificado como uma construção e é ótimo no sentido de que gera precisamente $(q^k-1)/n$ classes de equivalência cíclica, que é o limitante superior usando a classe de códigos em questão. Por fim, amplia-se o número de códigos lineares cíclicos q-ários que podem ser usados para gerar códigos CP;
- O Teorema 4.2 mostra como construir códigos CP por meio dos códigos lineares constacíclicos p-ários, de comprimento de bloco $p+1$, cujo polinômio gerador satisfaz o Teorema 3.1. Tal procedimento pode ser qualificado como uma construção e é ótimo no sentido de que gera precisamente $(p^k-1)/(p^2-1)$ classes de equivalência constacíclica, que é o limitante superior usando a classe de códigos em questão. Por fim, amplia-se a gama de possibilidades para gerar códigos CP e não há na literatura, que seja do conhecimento do autor desta tese, resultado semelhante publicado.
- Por meio dos códigos CP da Construção 4.1, são propostas novas sequências de protocolo

para o CCsR (Subseção 5.2.2). Tais sequências dão suporte a usuários com diferentes fatores de trabalho, além do mais, elas possuem um desempenho satisfatório (Subseção 5.3.1) principalmente por terem um número maior de sequências quando comparadas com as Sequências-RS [18] e as Sequências-Constacíclicas tipo-I [33].

1.6 Organização da Tese

O conteúdo desta tese está dividido em 6 capítulos. As referências encontram-se nas páginas finais e são ordenadas de acordo com a ordem em que foram citadas no texto. A seguir, apresenta-se um resumo dos capítulos seguintes desta tese.

Capítulo 2. Neste capítulo, os códigos utilizados pelo bloco *codificador de canal* da Figura 1.1 são apresentados. Embora o objetivo não seja utilizá-los no contexto de codificação de canal, são introduzidos os conceitos básicos destes códigos. São discutidos códigos de bloco e as propriedades que podem ser aproveitadas ao se utilizar uma estrutura algébrica para eles. O Teorema 2.8, citado na Seção 1.5, é apresentado nesse capítulo.

Capítulo 3. Após a introdução dos códigos corretores de erro no capítulo anterior, é apresentada a classe de códigos constacíclicos [12], também denominados de códigos pseudocíclicos. Trata-se de uma generalização dos códigos cíclicos apresentados no Capítulo 2. Devido ao fato de que tal classe de códigos ser pouco difundida na literatura, um capítulo é dedicado para a apresentação deles. Assim como os códigos cíclicos do Capítulo 2, os códigos constacíclicos são usados para construção de códigos ciclicamente permutáveis. O Teorema 3.1, citado na Seção 1.5, é apresentado nesse capítulo.

Capítulo 4. Os códigos ciclicamente permutáveis introduzidos por Gilbert [17]são apresentados neste capítulo. Para construir os códigos CP propostos nesta tese, são utilizados os códigos cíclicos do Capítulo 2 assim como os códigos constacíclicos do Capítulo 3. Nesse capítulo são apresentados os teoremas 4.1 e 4.2, assim como a Construção 4.1, todos citados na na Seção 1.5.

Capítulo 5. Uma vez que os códigos CP foram apresentados no Capítulo 4, este capítulo tem por objetivo mostrar a aplicação de tais códigos como sequências de protocolo para o CCsR [18], [22], [23]. É dada uma breve descrição do CCsR e são apresentados alguns conjuntos sequências de protocolo cujos parâmetros são comparados com o intuito de avaliar do desempenho das sequências.

Capítulo 6. Neste capítulo são apresentadas sugestões para trabalhos futuros. Também são apresentadas as publicações de trabalhos relacionados ao trabalho de pesquisa da tese e de trabalhos de pesquisa em atividades paralelas.

Capítulo 2

Códigos Corretores de Erro

Science is facts; just as houses are made of stones, so is science made of facts; but a pile of stones is not a house and a collection of facts is not necessarily science.

— Jules Henri Poincaré

Shannon estabeleceu, por meio do teorema para codificação de canais ruidosos [2], que existem códigos capazes de permitir a transmissão da informação por um canal ruidoso com uma taxa de erro de *bit* arbitrariamente pequena, desde que a taxa de informação transmitida seja menor que uma grandeza definida como capacidade de canal. Embora o teorema em questão garanta a existência desses códigos, ele não mostra como obtê-los. Desde então, um novo ramo surgiu na área de comunicações, a *codificação de canal*. Em sistemas de comunicação digital, a finalidade dessa codificação é inserir, por meio de *códigos corretores de erro*, redundância de maneira controlada na informação a ser transmitida. Deste modo, o receptor deve ser capaz de detectar, ou de detectar e corrigir, eventuais erros que ocorram durante a transmissão da mensagem por um canal ruidoso. Nesta tese, códigos corretores de erro bastante estudados na literatura são usados com outro objetivo. O intuito é utilizá-los para construir códigos ciclicamente permutáveis.

Neste capítulo, são apresentados os conceitos básicos sobre códigos corretores de erro. São discutidos códigos de bloco e as propriedades que podem ser usufruídas ao se utilizar uma estrutura algébrica para eles. Ao leitor que desejar um aprofundamento sobre o tema, recomenda-se a leitura das referências bibliográficas [9]–[14]. Para um bom entendimento deste capítulo é necessário um conhecimento básico de corpos finitos e álgebra linear. Uma excelente introdução a esses conteúdos pode ser encontrada em [9, Cap. 2] e [10, Caps. 2 e 3].

2.1 Códigos de Bloco

A seguir, é apresentada a definição dos códigos de bloco e são discutidas algumas de suas características e propriedades. Nesta tese, só são abordados códigos de bloco, de modo que sempre que se utilizar o termo *código*, com relação a códigos corretores de erro, ele está referindo-se a códigos de bloco.

Definição 2.1 – Código de bloco

*Um código de bloco q-ário $\mathcal{C}$ é o conjunto $\{\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{L-1}\}$ formado por L n-uplas q-árias de comprimento n denotadas por $\mathbf{c}_i = (c_0, c_1, c_2, \ldots, c_{n-1})$, $i = 0, 1, 2, \ldots, L-1$, e denominadas **palavras-código** ou **vetores-código**.* □

Ao código $\mathcal{C}$ da Definição 2.1, é associado um codificador que é responsável pelo mapeamento das mensagens, emitidas por uma fonte de informação q-ária, nas palavras-código $\mathbf{c}_i$. Vale ressaltar que, para um dado código $\mathcal{C}$, o codificador não é único [14]. Isto é, o mapeamento entre mensagens e vetores-código não é único.

O processo de codificação consiste, primeiramente, em dispor os dados emitidos pela fonte de informação em blocos de comprimento k e, em seguida, mapeá-los em palavras-código. Esse mapeamento é um-a-um, para permitir que o destinatário seja capaz de recuperar a mensagem original enviada. Se a fonte de informação emite símbolos de um alfabeto q-ário, então as possíveis mensagens a serem codificadas correspondem a k-uplas, $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$, que formam um espaço vetorial sobre GF(q). Visto que cada vetor $\mathbf{m}$ possui k símbolos e cada símbolo pode assumir q valores distintos, o total de possíveis *vetores-mensagem* é igual a q^k. Desta forma, o valor máximo de L é q^k. O processo de codificação é ilustrado na Figura 2.1. Entretanto, há situações em que $L \neq q^k$. Nesses casos, a implementação, em geral, torna-se mais complexa. Em [10, pág. 69] mostra-se um exemplo em que o codificador tem de arranjar as mensagens em blocos com comprimento variável para tratar o caso em que $L \neq q^k$.

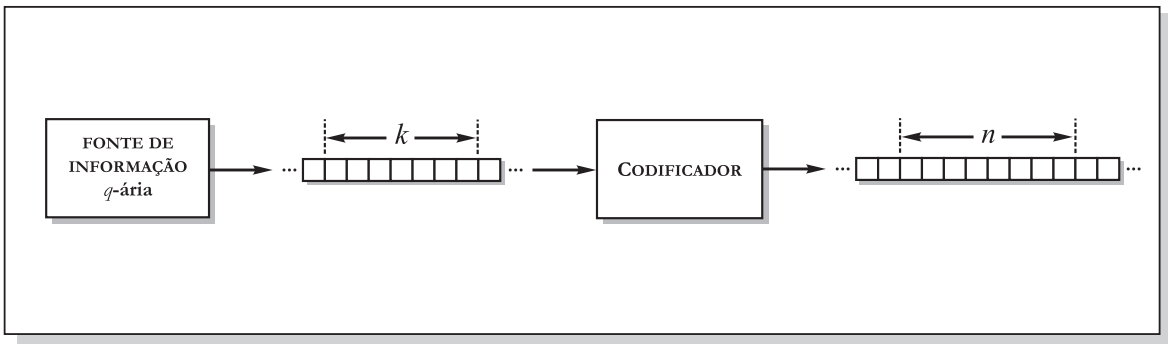


Figura 2.1: *Processo de codificação dos códigos de bloco.*

É analisado, neste ponto, como a utilização dos códigos de bloco permite inserir redundância, de modo controlado, nas mensagens codificadas. O conjunto de todas as n-uplas q-árias de comprimento n constituem o espaço vetorial $\mathbf{V}$ sobre $\mathrm{GF}(q)$ contendo um total de q^n vetores. O conjunto de vetores que pertencem a $\mathcal{C}$ é um subconjunto de $\mathbf{V}$, portanto existem vetores de $\mathbf{V}$ que não estão em $\mathcal{C}$. Diz-se, então, que um código de bloco possui *redundância* quando o número de vetores que pertencem ao código é menor que o número total de vetores q-ários de comprimento n, ou seja, $L < q^n$. A redundância r pode ser expressa em forma logarítmica [10] por

$$r = n - \log_q L. \tag{2.1}$$

Em geral, utilizam-se códigos em que $L = q^k$. Portanto, a Fórmula (2.1) torna-se $r = n-k$. Ou seja, dos n símbolos transmitidos, k símbolos são de informação e o restante, $n - k$, são de redundância. Uma maneira mais usual para expressar a redundância de um código, é definindo a sua taxa.

Definição 2.2 – Taxa de um código

Seja C um código de bloco q-ário com L palavras-código, cada uma de comprimento n. A taxa R do código C é dada por

$$R = \frac{\log_q L}{n}. \tag{2.2}$$

Para os casos em que $L = q^k$, a Fórmula (2.2) reduz-se a

$$R = \frac{k}{n}. \tag{2.3}$$

□

A seguir, são apresentadas algumas definições e teoremas importantes na teoria dos códigos de bloco.

Definição 2.3 – Peso de Hamming de um vetor

*O **peso de Hamming**, ou simplesmente **peso**, de um vetor q-ário, em geral denotado por $w(\mathbf{v})$, é o número de coordenadas não nulas deste vetor.* □

Exemplo 2.1

Considere os seguintes vetores: $\mathbf{v}_1 = (1,0,0,1,1,0,1,0,1,1)$ *um vetor binário,* $\mathbf{v}_2 = (2,0,2,1,1,0,1,0,1,1)$ *um vetor ternário e* $\mathbf{v}_3 = (\alpha^3,0,0,0,0,0,\alpha^4,0,0,0)$ *cujos elementos pertencem a* $\mathrm{GF}(2^3)$*. Os respectivos pesos são:* $w(\mathbf{v}_1) = 6$*,* $w(\mathbf{v}_2) = 7$ *e* $w(\mathbf{v}_3) = 2$*.* □

Definição 2.4 – Distância de Hamming

*A **distância de Hamming** entre dois vetores* $\mathbf{v}$ *e* $\mathbf{w}$*, de mesmo comprimento* n*, é o número de coordenadas em que eles diferem.*

$$d_{\mathrm{Hamming}}(\mathbf{v},\mathbf{w}) \triangleq d(\mathbf{v},\mathbf{w}) = \#\{i \mid v_i \neq w_i,\ i = 0,1,\ldots,n-1\}, \tag{2.4}$$

em que $\#\{.\}$ *denota a cardinalidade*[1] *do conjunto.* □

Exemplo 2.2

No exemplo 2.1, os vetores $\mathbf{v}_1$*,* $\mathbf{v}_2$ *e* $\mathbf{v}_3$ *são todos de mesmo comprimento,* $n = 10$*. Desta forma, a distância de Hamming entre eles é:* $d(\mathbf{v}_1,\mathbf{v}_2) = 2$*,* $d(\mathbf{v}_1,\mathbf{v}_3) = 6$ *e* $d(\mathbf{v}_2,\mathbf{v}_3) = 7$*.* □

A Definição 2.4 tem uma importância significativa na teoria de códigos corretores de erro. A partir dela, pode-se definir um importante parâmetro para caracterizar a capacidade de detecção de erro, a capacidade de correção de erro e a capacidade de correção de apagamento. Tal parâmetro é definido a seguir.

Definição 2.5 – Distância mínima de um código

*A **distância mínima**, denotada por* $d_{\min}$*, de um código de bloco* $\mathcal{C}$ *é a menor distância de Hamming entre todos os pares distintos de palavras-código pertencentes a* $\mathcal{C}$*.* □

Exemplo 2.3

Considere um código binário $\mathcal{C}$ *formado pelo seguinte conjunto de palavras:* $\{\mathbf{c}_0;\mathbf{c}_1;\mathbf{c}_2;\mathbf{c}_3\}$*, em que* $\mathbf{c}_0 = (0,0,0,1)$*,* $\mathbf{c}_1 = (1,1,0,1)$*,* $\mathbf{c}_2 = (0,1,0,1)$ *e* $\mathbf{c}_3 = (1,0,1,0)$*. Para determinar a distância mínima do código* $\mathcal{C}$*, calcula-se a distância de Hamming entre todos os pares de palavras-código. Como o código possui quatro palavras no total e* $d(\mathbf{v},\mathbf{w}) = d(\mathbf{w},\mathbf{v})$*, o total de distâncias*

[1] Termo utilizado para designar o número de elementos de um conjunto.

calculadas que é dado por $\binom{4}{2} = 6$. *Os resultados são:* $d(\mathbf{c}_0, \mathbf{c}_1) = 2$, $d(\mathbf{c}_0, \mathbf{c}_2) = 1$, $d(\mathbf{c}_0, \mathbf{c}_3) = 3$, $d(\mathbf{c}_1, \mathbf{c}_2) = 1$, $d(\mathbf{c}_1, \mathbf{c}_3) = 3$ *e* $d(\mathbf{c}_2, \mathbf{c}_3) = 4$. *Logo,* $d_{\min} = 1$. □

À medida que o número de palavras de um código aumenta, maior é o número de comparações que devem ser feitas para encontrar sua distância mínima. Para ser mais específico, o número de comparações necessárias é dado por $\binom{L}{2} = L(L-1)/2$ ou $q^k(q^k-1)/2$, quando $L = q^k$, o que mostra que a complexidade do cálculo é exponencial em k. Uma vez definida a distância mínima de um código, pode-se determinar a capacidade de detecção de erro, a capacidade de correção de erro e a capacidade de correção de apagamento para um código de bloco. Daqui em diante, denomina-se *padrão de erro* o vetor que representa os possíveis erros que surgem durante a transmissão da palavra-código.

Teorema 2.1 – Capacidade de detecção de erro [9]

Um código $\mathcal{C}$ *com distância mínima* $d_{\min}$ *é capaz de detectar todos os padrões de erro com peso menor ou igual a* $d_{\min} - 1$. □

Demonstração: Vide [9, pág. 78]. ■

Teorema 2.2 – Capacidade de correção de erro [11]

Um código $\mathcal{C}$ *com distância mínima* $d_{\min}$ *é capaz de corrigir todos os padrões de erro com peso menor ou igual a* $t = \lfloor \frac{d_{\min}-1}{2} \rfloor$, *em que* $\lfloor x \rfloor$ *denota o único número inteiro* i *tal que* $i \leq x < i + 1$. *Se* $d_{\min}$ *é um número par, então o código é capaz de corrigir* $(d_{\min} - 2)/2$ *erros ou detectar* $d_{\min}/2$. □

Demonstração: Vide [11, pág. 10]. ■

Códigos corretores de erro também são utilizados em canais com *apagamento*. Por exemplo, um *canal binário com apagamento* [5], em que os símbolos na entrada do canal são 0 ou 1 e na saída do canal são 0, 1 ou $\star$, em que $\star$ denota o apagamento. Em geral, um canal p-ário com apagamento possui p símbolos para o alfabeto de entrada e $p+1$ para o alfabeto de saída, tendo $\star$ como o símbolo adicional no alfabeto de saída para indicar apagamento. O número de apagamentos que um código de bloco com distância mínima $d_{\min}$ pode corrigir é dado no teorema a seguir. Similar à definição para padrões de erro, daqui em diante, denomina-se *padrão de apagamento* o vetor que representa apagamentos que afetam uma palavra-código após ser transmitida através de um canal ruidoso.

Teorema 2.3 – Capacidade de correção de apagamento [13]

Um código C com distância mínima $d_{\min}$ é capaz de corrigir todos os padrões de apagamento com peso menor ou igual a ρ se $d_{\min} \geq \rho + 1$. Além do mais, quaisquer padrões de erro e apagamento em que ocorram t ou menos erros e ρ ou menos apagamentos simultaneamente, podem ser corrigidos se $d_{\min} \geq 2t + \rho + 1$. □

Demonstração: Vide [13, pág. 14] ■

Um dos problemas com os códigos de bloco, em geral, é que é necessário armazenar todas as palavras do código para poder executar os processos de codificação e decodificação e quando o valor de k aumenta, o número de palavras-código aumenta exponencialmente. É visto na próxima seção que a complexidade desse problema pode ser reduzida se os códigos de bloco forem *lineares*.

2.2 Códigos de Bloco Lineares

Nesta seção, mostra-se que a *linearidade* provê uma estrutura matemática aos códigos de bloco que permite fazer várias simplificações com relação às propriedades discutidas na seção anterior. Inicialmente, define-se um código de bloco linear.

Definição 2.6 – Códigos de bloco lineares

Um código de bloco q-ário C com q^k palavras-código é um ***código de bloco linear*** $(n, k, d_{\min})$ *se e somente se suas q^k palavras-código formam um subespaço de dimensão k do espaço vetorial de todas as n-uplas sobre* $\mathrm{GF}(q)$. □

A Definição 2.6 permite que se utilizem várias propriedades de espaços vetoriais amplamente conhecidas da álgebra linear [10, Cap. 2].

Propriedade 2.1 – [10]

A combinação linear de qualquer conjunto de palavras-código é uma palavra-código. Uma consequência disto é que um código linear sempre contém a palavra-código toda nula, daqui por diante, denotada por **0**. □

Demonstração: A prova é uma consequência direta da definição de espaço vetorial. Vide [10, pág. 31]. ■

Propriedade 2.2 – [10]

A distância mínima de um código de bloco linear é igual ao peso da palavra-código de menor peso entre todas as palavras do código não-nulas. □

Demonstração: Vide [10, pág. 83] ■

Pela Propriedade 2.2 percebe-se como a complexidade para encontrar a distância mínima de um código de bloco linear é reduzida. Anteriormente, foi mencionado que a complexidade para encontrar a $d_{\min}$ de um código de bloco é dada por $q^k(q^k-1)/2$. Para códigos lineares, este valor diminui para q^k-1, pois só é preciso encontrar a palavra-código não-nula com menor peso.

2.2.1 Matriz Geradora e Matriz de Verificação de Paridade

Dado um espaço vetorial $\mathbf{V}$, pode-se escolher um subconjunto finito de vetores, $\{\mathbf{v}_i\}$, tal que $\mathbf{v}_i \in \mathbf{V}$, de modo que qualquer outro vetor pertencente a $\mathbf{V}$ pode ser obtido como uma combinação linear dos vetores que estão no subconjunto $\{\mathbf{v}_i\}$. Entretanto, se os vetores que constituem o subconjunto $\{\mathbf{v}_i\}$ forem linearmente independentes, obtém-se uma base vetorial para o espaço $\mathbf{V}$. Por meio de uma base vetorial, o mapeamento entre as combinações lineares dos vetores-base e todos os vetores que pertencem a $\mathbf{V}$ é um-a-um. A definição de códigos lineares como subespaços vetoriais, permite, então, obter uma eficiente representação para esses códigos.

Seja $\{\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}\}$ uma base de vetores-código de um código linear q-ário $(n, k, d_{\min})$. Então, cada palavra-código $\mathbf{c}$ pode ser representada de modo único por $\mathbf{c} = m_0\mathbf{g}_0 + m_1\mathbf{g}_1 + \ldots + m_{k-1}\mathbf{g}_{k-1}$, em que $m_i \in \mathrm{GF}(q)$, para $0 \leq i \leq k-1$, representam as coordenadas do vetor-mensagem $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$. Esta situação pode ser representada matricialmente caso definam-se os vetores-base $\mathbf{g}_i$ como linhas de uma *matriz geradora*, denotada por $\mathbf{G}$, do seguinte modo

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}. \tag{2.5}$$

Pode-se usar diretamente a matriz $\mathbf{G}$ para codificar os vetores-mensagem $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$ em vetores-código $\mathbf{c}$ procedendo da seguinte forma

$$\mathbf{c} = \mathbf{mG} = (m_0, m_1, \ldots, m_{k-1}) \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = m_0\mathbf{g}_0 + m_1\mathbf{g}_1 + \ldots + m_{k-1}\mathbf{g}_{k-1}. \tag{2.6}$$

A representação dos códigos lineares, por meio da matriz $\mathbf{G}$, soluciona o problema de armazenar todas as palavras de um código de bloco para executar o processo de codificação. Basta, neste caso, armazenar k palavras-código linearmente independentes. Vale ressaltar que o número de matrizes $\mathbf{G}$ distintas que podem representar um mesmo código linear q-ário $(n, k, d_{\min})$ é dado por [8]

$$\prod_{i=0}^{k-1} \left(q^k - q^i\right). \tag{2.7}$$

É interessante destacar que o número de possíveis codificadores para um código de bloco não-linear com q^k palavras-código é igual $(q^k)!$ [8]. Esse número é bem maior que o número de matrizes geradoras distintas para um código linear dado por (2.7). A redução, deve-se ao fato da restrição de que as linhas de $\mathbf{G}$ são linearmente independentes.

Dentre todos os possíveis codificadores previstos por (2.7), um deles destaca-se dentre os demais, é o *codificador sistemático*. Ao utilizar esse codificador, é possível distinguir na palavra-código quais são os símbolos de informação e quais são os símbolos de redundância. Em situações práticas, os códigos construídos na forma sistemática são preferidos. A matriz geradora de um código sistemático possui a seguinte forma

$$\mathbf{G} = [\mathbf{P}|\mathbf{I}_k] = \begin{bmatrix} p_{0,0} & p_{0,1} & \ldots & p_{0,n-k-1} & 1 & 0 & 0 & \ldots & 0 \\ p_{1,0} & p_{1,1} & \ldots & p_{1,n-k-1} & 0 & 1 & 0 & \ldots & 0 \\ p_{2,0} & p_{2,1} & \ldots & p_{2,n-k-1} & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \ldots & p_{k-1,n-k-1} & 0 & 0 & 0 & \ldots & 1 \end{bmatrix}. \tag{2.8}$$

Exemplo 2.4

O código linear binário $(7, 4, 3)$ *possui a seguinte matriz geradora:*

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

As palavras-código $\mathbf{g}_0$, $\mathbf{g}_1$, $\mathbf{g}_2$ *e* $\mathbf{g}_3$ *são linearmente independentes e formam uma base vetorial para o subespaço de dimensão* k *que corresponde às palavras do código.* □

Outra importante matriz associada a um código de bloco linear é a *matriz de verificação de paridade* denotada por $\mathbf{H}$. As linhas desta matriz são elementos do conjunto de vetores $\{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{n-k-1}\}$ que formam uma base vetorial do espaço $\mathcal{C}^{\perp}$, o qual denota o espaço dual do código $\mathcal{C}$ gerado por $\mathbf{G}$,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{n-k-1} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{0,1} & \ldots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \ldots & h_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \ldots & h_{n-k-1,n-1} \end{bmatrix}. \tag{2.9}$$

Na forma sistemática, a matriz $\mathbf{H}$ pode ser obtida diretamente da matriz $\mathbf{G}$ sistemática (2.8),

$$\mathbf{H} = [\mathbf{I}_{n-k} | -\mathbf{P}^T] = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & -p_{0,0} & -p_{1,0} & \ldots & p_{k-1,0} \\ 0 & 1 & 0 & \ldots & 0 & -p_{0,1} & -p_{1,1} & \ldots & p_{k-1,1} \\ 0 & 0 & 1 & \ldots & 0 & -p_{0,2} & -p_{1,2} & \ldots & p_{k-1,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & -p_{0,n-k-1} & -p_{1,n-k-1} & \ldots & p_{k-1,n-k-1} \end{bmatrix}. \tag{2.10}$$

Teorema 2.4 – [10]

Um vetor $\mathbf{c}$ *é uma palavra do código* $\mathcal{C}$ *se e somente se* $\mathbf{c}\mathbf{H}^T = \mathbf{0}$, *em que* $\mathbf{H}^T$ *denota a matriz transposta da matriz de verificação de paridade* $\mathbf{H}$. □

Demonstração: Vide [10, pág. 84]. ■

Teorema 2.5 – [10]

Seja C um código linear com matriz de paridade **H**. *A distância mínima de C é igual ao menor número, diferente de zero, de colunas da matriz* **H** *cuja combinação linear resulta em* $\mathbf{0}^T$. □

Demonstração: Vide [10, pág. 84]. ■

Teorema 2.6 – Cota de Singleton [10]

A distância mínima de um código $(n, k, d_{\min})$ *é limitada superiormente por*

$$d_{\min} \leq n - k + 1. \tag{2.11}$$

□

Demonstração: Vide [10, pág. 84]. ■

Códigos que satisfazem a cota de Singleton com igualdade são conhecidos como *códigos MDS* (*Maximum Distance Separable*), ou seja, códigos cuja distância mínima é a máxima possível.

Para evitar dúvidas quando se fizer referência a códigos de bloco lineares e códigos de bloco não-lineares, daqui por diante, usa-se a terminologia *códigos não-lineares* para designar códigos de bloco que não são lineares.

2.3 Códigos Cíclicos

Assim como a linearidade agregou mais propriedades aos códigos de bloco, mais propriedades podem ser acrescentadas se estes códigos também forem *cíclicos*. Os códigos cíclicos obtiveram grande destaque em aplicações práticas como, por exemplo, no *compact disc* (CD) e no *NASA Deep Space Standard* para comunicações via satélite [10].

Definição 2.7 – Códigos cíclicos

Um código de bloco C é denominado ***cíclico*** *se qualquer deslocamento cíclico de uma palavra-código resulta em uma palavra-código.* □

Embora seja bastante comum empregar o termo *código cíclico* a um código que é simultaneamente linear e cíclico [9], [10], [14], a Definição 2.7 abrange os códigos lineares e os códigos não-lineares. De modo que, para evitar ambiguidade, usa-se a expressão *código cíclico linear* para designar os casos em que o código em questão é linear e cíclico.

Ao trabalhar com códigos cíclicos, é comum representar as palavras-código por meio de polinômios. Isto é, a palavra-código $\mathbf{c} = (c_0, c_1, c_2, \ldots, c_{n-1})$ pode ser representada pelo *polinômio-código* $c(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-1}x^{n-1}$.

Teorema 2.7 – Propriedades [10]

Seja $\mathcal{C}$ um código cíclico linear q-ário $(n, k, d_{\min})$.

a. *Existe um único polinômio mônico $g(x)$ de grau $n-k$, o qual representa uma palavra-código* $\mathbf{g}$*, que é o polinômio de menor grau entre todos os polinômios que representam as palavras-código de $\mathcal{C}$. $g(x)$ é denominado* ***polinômio gerador*** *do código $\mathcal{C}$.*

b. *O polinômio gerador $g(x)$ de um código cíclico linear $(n, k, d_{\min})$ é fator de $x^n - 1$.*

c. *Cada polinômio-código $c(x) \in \mathcal{C}$ é expresso unicamente como $c(x) = m(x)g(x)$, em que $m(x)$ é o* ***polinômio-mensagem*** *de grau menor que k e $g(x)$ é o polinômio gerador de $\mathcal{C}$.* □

Demonstração: Vide [11, pág. 191]. ■

Uma observação importante é que para garantir a Propriedade ***a*** no Teorema 2.7, $g_0 \neq 0$. Pois, deslocar ciclicamente $g(x)$ uma posição para a esquerda produz $g'(x) = g_1 + g_2x + \ldots + x^{n-k-1} + g_0x^{n-k}$. Logo, $g_0 \neq 0$, senão um polinômio de grau menor que $n-k$ seria um polinômio-código, o que é impossível.

Considere, neste ponto, como os deslocamentos cíclicos realizados nas palavras-código podem ser reproduzidos na representação polinomial. Se $\mathbf{c} = (c_0, c_1, c_2, \ldots, c_{n-1}) \in \mathcal{C}$, então a palavra-código obtida ao deslocar $\mathbf{c}$ uma posição para direita é $\mathbf{c}' = (c_{n-1}, c_0, c_1, \ldots, c_{n-2}) \in \mathcal{C}$. Este resultado é obtido na representação polinomial ao se efetuar a seguinte operação

$$\begin{aligned} xc(x) &= (c_0x + c_1x^2 + c_2x^3 + \ldots + c_{n-1}x^n) \bmod (x^n - 1) \\ &= (c_{n-1} + c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1}) \bmod (x^n - 1) \\ &= c'(x) \bmod (x^n - 1). \end{aligned}$$

Sendo assim, efetuar dois deslocamentos para a direita em $\mathbf{c}$ seria equivalente a multiplicar $c(x)$ por $x^2 \bmod (x^n - 1)$ e assim sucessivamente até $n-1$ deslocamentos, os quais seriam obtidos, na forma polinomial, multiplicando $c(x)$ por x^{n-1}. Obviamente, multiplicar por x^n seria o mesmo que multiplicar por 1, pois $x^n = 1 \bmod (x^n - 1)$. Na forma vetorial, significa dizer que ao deslocar-se ciclicamente n vezes a palavra-código original é obtida. A partir desta constatação, pode-se definir a *ordem cíclica* de uma palavra-código.

Definição 2.8 – Ordem cíclica

*A **ordem cíclica** de uma palavra-código é o menor inteiro positivo i tal que $x^i c(x) = c(x) \bmod (x^n - 1)$.* □

Embora a Definição 2.8 refira-se a vetores que pertencem a um código cíclico, o conceito de ordem cíclica pode ser expandido a qualquer n-upla q-ária de comprimento n. É importante ressaltar que os possíveis valores para a ordem cíclica de uma palavra-código são os divisores de n [30, pág. 148]. Diz-se, então, que uma palavra-código tem ***ordem cíclica plena*** quando a ordem cíclica desta palavra é igual a n.

O Teorema 2.8 enunciado na sequência trata-se de uma contribuição desta tese e foi publicado originalmente em [42]. Ele estabelece uma condição para o conjunto de raízes do polinômio gerador $g(x)$ de um código linear cíclico q-ário tal que todas as palavras-código não-nulas tenham ordem cíclica plena.

Teorema 2.8

Seja n um divisor de $q^m - 1$, em que m é um número inteiro positivo, sendo q a potência de um número primo p, e seja $\mathcal{C}$ um código linear cíclico q-ário $(n, k, d_{\min})$ cujo polinômio gerador é $g(x)$. Todas as palavras-código não nulas de $\mathcal{C}$ tem ordem cíclica plena se e somente se o conjunto de raízes de $g(x)$ em $\mathrm{GF}(q^m)$ inclui todas as raízes de $x^n - 1$ que possuem ordem multiplicativa menor que n. □

Demonstração: Suponha que entre as raízes de $g(x)$ estão todas as raízes de $x^n - 1$ com ordem multiplicativa menor que n. Além do mais, suponha que $c(x) \in \mathcal{C}$ é uma palavra-código com ordem cíclica $i < n$, ou seja,

$$x^i c(x) = c(x) \mod (x^n - 1). \tag{2.12}$$

Uma vez que $c(x) = m(x)g(x)$ (Teorema 2.7), então (2.12) pode ser reescrita da seguinte forma

$$(x^i - 1)m(x)g(x) = 0 \mod (x^n - 1). \tag{2.13}$$

A condição expressa em (2.13) implica que todas as raízes de $x^n - 1$ estão contidas em $(x^i - 1)m(x)g(x)$. Como o grau$[m(x)g(x)] \leq n - 1$, no mínimo uma raiz de $x^n - 1$ é comum a $x^i - 1$. Entretanto, todas as raízes de $x^n - 1$ que possuem ordem multiplicativa menor que n estão entre as raízes de $g(x)$ por hipótese. Portanto, as raízes de $x^n - 1$ em comum com $x^i - 1$ têm ordem multiplicativa n, ou seja, $i = n$ é o menor valor de i para o qual (2.13) é satisfeita. Logo, todas as palavras não nulas de $\mathcal{C}$ têm ordem cíclica plena.

Tabela 2.1: *Classes conjugadas e polinômios mínimos sobre* GF(2) *para* $x^{15}+1$

Classe conjugada	Polinômio mínimo
$\{0\}$	$M_0(x) = 1 + x$
$\{1, 2, 4, 8\}$	$M_1(x) = 1 + x + x^4$
$\{3, 6, 12, 9\}$	$M_3(x) = 1 + x + x^2 + x^3 + x^4$
$\{5, 10\}$	$M_5(x) = 1 + x + x^2$
$\{7, 14, 13, 11\}$	$M_7(x) = 1 + x^3 + x^4$

Por outro lado, suponha que todas as palavras-código não nulas de $\mathcal{C}$ possuem ordem cíclica plena, ou seja, n é o menor valor de i que satisfaz (2.13). Pelos mesmos argumentos já expostos, no mínimo uma raiz de x^n-1 é comum a x^i-1 em (2.13). Se esta raiz comum, denotada por α^{n_1}, tem ordem multiplicativa $n_2 < n$, tal que $\alpha^{n_1 n_2} = \alpha^n = 1$, então (2.13) é satisfeita para $i = n_2$. Porém, isto implica que pelo menos uma palavra-código possui ordem cíclica $n_2 < n$. Como isto não é possível, dada a hipótese assumida que $i = n$ é o menor valor para o qual (2.13) é satisfeita, então qualquer raiz com ordem multiplicativa $i < n$ deve estar no conjunto de raízes de $m(x)$ ou de $g(x)$. Uma vez que o polinômio $m(x)$ pode conter ou não raízes em GF(q^m), o único modo de garantir a hipótese assumida é que, em geral, todas as raízes com ordem multiplicativa $i < n$ devem estar no conjunto de raízes de $g(x)$. ■

Exemplo 2.5

De acordo com a Tabela 2.1, $x^{15}+1 = M_1(x)M_3(x)M_5(x)M_7(x)$. *Considere o código linear cíclico binário* $(15, 8, 4)$ *cujo polinômio gerador é* $g(x) = M_0(x)M_3(x)M_5(x) = (x+1)(1+x+x^2+x^3+x^4)(1+x+x^2)$. *Todas as raízes de* $x^{15}+1$ *com ordem multiplicativa menor que* 15 *estão no conjunto de raízes de* $g(x)$, *pois* $M_0(x)$ *pertence ao expoente* 1, $M_3(x)$ *pertence ao expoente* 5 *e* $M_5(x)$ *pertence ao expoente* 3. *Portanto, pelo Teorema 2.8 esse código possui* $2^8 - 1 = 255$ *palavras-código não-nulas com ordem cíclica plena.* □

2.3.1 Matriz Geradora e Matriz de Verificação de Paridade

A forma geral da matriz **G** de um código cíclico linear pode ser obtida por meio da Propriedade ***c*** no Teorema 2.7. Sendo o polinômio mensagem $m(x) = m_0 + m_1 x + \ldots + m_{k-1}x^{k-1}$,

a multiplicação $c(x) = m(x)g(x)$ pode ser escrita da seguinte forma

$$\begin{aligned} c(x) &= (m_0 + m_1 x + \ldots + m_{k-1}x^{k-1})g(x) \\ &= m_0 g(x) + m_1 x g(x) + \ldots + m_{k-1}x^{k-1}g(x). \end{aligned} \tag{2.14}$$

A Expressão (2.14) pode ser reescrita na forma de multiplicação de matrizes como pode ser observado em (2.15),

$$c(x) = [m_0\ m_1\ \ldots\ m_{k-1}] \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}. \tag{2.15}$$

Lembrando que multiplicar $g(x)$ por $x^i \bmod (x^n - 1)$, sendo i um número inteiro positivo, é equivalente a deslocar ciclicamente $\mathbf{g}$ de i posições para a direita, então a matriz $\mathbf{G}$ pode ser escrita conforme (2.16). Também é possível construir a matriz $\mathbf{G}$ na forma sistemática para um código cíclico linear. Um algoritmo mostrando como realizar este procedimento pode ser visto em [10, pág. 107].

$$\mathbf{G} = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix} \equiv \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & & & & \mathbf{0} \\ & g_0 & g_1 & \cdots & g_{n-k} & & & \\ & & \ddots & \ddots & \ddots & \ddots & & \\ & & & g_0 & g_1 & \cdots & g_{n-k} & \\ \mathbf{0} & & & & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}. \tag{2.16}$$

Segue da Propriedade ***b*** que para cada polinômio gerador $g(x)$, existe um *polinômio de verificação de paridade*, ou simplesmente *polinômio-paridade*, denotado por $h(x)$, que é mônico, com $h_0 \neq 0$ e de grau k, tal que $g(x)h(x) = x^n - 1$. Sabe-se que $c(x)$ é um polinômio-código se e somente se ele é um múltiplo de $g(x)$, alternativamente, $c(x)$ é um polinômio-código se e somente se $c(x)h(x) = 0 \bmod (x^n - 1)$. Esta equação pode ser manipulada [10] de modo a ser escrita na forma matricial como $\mathbf{cH}^T = \mathbf{0}$. A matriz $\mathbf{H}_{(n-k)\times n}$ em questão é a matriz de verificação de paridade para o código gerado por $g(x)$,

$$\mathbf{H} = \begin{bmatrix} h_k & \ldots & h_1 & h_0 & & & \mathbf{0} \\ & h_k & \ldots & h_1 & h_0 & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & h_k & \ldots & h_1 & h_0 \\ \mathbf{0} & & & & h_k & \ldots & h_1 & h_0 \end{bmatrix}^T. \tag{2.17}$$

Como pode ser visto em (2.17), as linhas da matriz **H** são formadas pelo deslocamento cíclico de um polinômio que não é o polinômio-paridade $h(x) = h_0 + h_1x + \ldots + h_kx^k$, mas é um polinômio cujos coeficientes são os mesmos de $h(x)$ só que em ordem reversa. Tal polinômio, $h^*(x) = h_k + h_{k-1}x + \ldots + h_1x^{k-1} + h_0x^k$, é denominado *polinômio-recíproco*[2] de $h(x)$. O teorema a seguir mostra uma importante relação entre $h^*(x)$ e o código dual $\mathcal{C}^\perp$ de um código cíclico linear $\mathcal{C}$ gerado por $g(x)$.

Teorema 2.9 – [10]

Seja $\mathcal{C}$ um código linear cíclico q-ário (n, k) com polinômio gerador $g(x)$. O código dual $\mathcal{C}^\perp$, do código $\mathcal{C}$, é um código cíclico linear q-ário $(n, n-k)$ cujo polinômio gerador é $h^*(x)$. □

Demonstração: Vide [10, pág. 104] ■

2.3.2 Códigos BCH

Os códigos BCH são códigos cíclicos lineares q-ários e são assim denominados em homenagem aos pesquisadores que os investigaram inicialmente: A. Hocquenghem em 1959 [43], Bose e Ray-Chaudhuri em 1960 [44], [45]. Entretanto, coube a outro pesquisador, W. Peterson, mostrar que esses códigos eram cíclicos e construir o primeiro algoritmo de decodificação algébrica para códigos BCH [46].

Em geral, quando se escolhe um polinômio gerador $g(x)$ e gera-se um código cíclico linear não se tem ideia do valor da distância mínima deste código. Por isto, os códigos BCH destacam-se em relação a códigos cíclicos arbitrários, por garantirem um limite inferior para a distância mínima do código. Tal resultado é obtido impondo algumas restrições à escolha do polinômio gerador $g(x)$. O limitante inferior para a distância mínima do código é denominado *distância projetada* do código e é denotado por δ. O teorema a seguir especifica como escolher as raízes do polinômio gerador a fim de garantir a distância projetada do código.

Teorema 2.10 – cota BCH [10]

Seja $\mathcal{C}$ um código linear cíclico q-ário $(n, k, d_{\min})$ cujo polinômio gerador é $g(x)$. Além do mais, seja m a ordem multiplicativa de $q \bmod n$ tal que $\mathrm{GF}(q^m)$ é o menor corpo de extensão de $\mathrm{GF}(q)$ que contém a n-ésima raiz da unidade. Considere α uma n-ésima raiz primitiva da unidade. Assegure-se que $g(x)$ foi escolhido como o polinômio de menor grau tal que $g(\alpha^b) = g(\alpha^{b+1}) = g(\alpha^{b+2}) = \ldots = g(\alpha^{b+\delta-2}) = 0$, em que b é um número inteiro $b \geq 0$ e $\delta \geq 1$. Desta forma, $g(x)$ tem $(\delta - 1)$

[2] Em geral, o polinômio-recíproco de $a(x) = a_0 + a_1x + \ldots + a_{n-1}x^{n-1} + a_nx^n$ é o polinômio $a^*(x) = x^n a(x^{-1}) = a_n + a_{n-1}x + \ldots + a_1x^{n-1} + a_0x^n$.

potências consecutivas de α como raízes. Então, o código C gerado por $g(x)$ tem distância mínima $d_{\min} \geq \delta$. □

Demonstração: Vide [10, pág. 176–180] ■

Vale destacar que em muitas situações a distância mínima, $d_{\min}$, do código BCH coincide com a distância projetada, δ. Entretanto, há também várias situações em que $d_{\min}$ é maior que δ [9].

Uma vez que a distância mínima do código BCH está garantida pela cota BCH do Teorema 2.10, um procedimento para construir um código BCH q-ário de comprimento n e corretor de t erros pode ser o seguinte [10]:

i. Encontre uma n-ésima raiz primitiva α de $x^n - 1$ em um corpo $\mathrm{GF}(q^m)$ para o menor valor possível de m;

ii. Selecione $(\delta - 1) = 2t$ potências consecutivas de α: α^b, α^{b+1}, α^{b+2},..., α^{b+2t}, sendo b um número inteiro não negativo;

iii. Seja $M_i(x)$ o polinômio mínimo de α^i e seus conjugados [9, pág. 48]. Então faça $g(x)$ o mínimo múltiplo comum entre os polinômios mínimos de todas as potências consecutivas de α.

$$g(x) = \mathrm{MMC}\{M_b(x), M_{b+1}(x), M_{b+2}(x), \ldots, M_{b+2t}(x)\}. \tag{2.18}$$

Para $b = 1$, os códigos BCH são denominados de *sentido restrito*. Para $n = q^m - 1$, m sendo um número inteiro positivo, o código BCH é denominado *primitivo*. Quando $n \neq q^m - 1$, os códigos BCH são denominados *não-primitivos* e os possíveis valores de n são os divisores de $q^m - 1$.

2.3.3 Códigos Reed-Solomon

Códigos Reed-Solomon (RS), assim como ocorre com os códigos BCH, recebem esta denominação em homenagem aos primeiros pesquisadores a investigá-los, I. S. Reed e G. Solomon [47]. Entretanto, há várias formas de definir um código RS [10]. A definição usada nesta tese segue aquela em que esses códigos são uma extensão natural dos códigos BCH.

Definição 2.9 – Códigos Reed-Solomon [10]

*Um **código Reed-Solomon** é um código BCH de comprimento $n = q^m - 1$ cujos símbolos pertencem a $\mathrm{GF}(q^m)$.* □

A partir da Definição 2.9, pode-se destacar que, diferentemente dos códigos BCH mencionados na subseção anterior, a n-ésima raiz da unidade não precisa ser procurada num corpo de extensão GF(q^m), pois os códigos RS são definidos sobre este corpo e, portanto, $\alpha \in \mathrm{GF}(q^m)$. Além do mais, os polinômios mínimos $M_s(x)$ são da forma $(x - \alpha^s)$, pois os elementos não nulos de GF(q^m) são as raízes de $x^{q^m-1} - 1$ e, desta forma, não há raízes conjugadas.

Códigos RS destacam-se com relação a outros códigos BCH por possuírem várias propriedades que estes outros códigos não compartilham. Não é por acaso que códigos RS constituem a subclasse dos códigos BCH não-binários mais conhecida, haja vista as várias aplicações em que eles são utilizados [48]. Entre essas propriedades, a mais importante está relacionada à distância mínima dos códigos RS como mostra o teorema a seguir.

Teorema 2.11 – Distância mínima de um código RS [10]

Um código RS $(n, k, d_{\min})$ *possui distância mínima igual a* $d_{\min} = n - k + 1$. □

Demonstração: Vide [10, pág. 188] ■

Desta forma, códigos RS são códigos MDS. Códigos MDS possuem propriedades interessantes que podem ser vistas em [10]. Uma delas é enunciada no teorema a seguir.

Teorema 2.12 – Distribuição de pesos [10]

O número de palavras-código com peso de Hamming j *em um código q-ário* (n, k) *e MDS é dado por*

$$A_j = \binom{n}{j}(q-1)\sum_{i=0}^{j-d_{\min}}(-1)^i\binom{j-1}{i}q^{j-i-d_{\min}}. \tag{2.19}$$

□

Demonstração: Vide [12, pág. 429–431]. ■

Capítulo 3

Códigos Constacíclicos

A teoria é o general; os experimentos são os soldados.

— Leonardo da Vinci

Códigos constacíclicos [12], também conhecidos como *códigos pseudocíclicos* [49]–[51], são uma generalização dos códigos cíclicos apresentados no Capítulo 2, conforme é visto na sequência.

3.1 Códigos Constacíclicos

Seja $c(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-1}x^{n-1}$ um polinômio cujos coeficientes pertencem a GF(p), em que p denota um número primo. Multiplicar $c(x)$ por x e reduzir o produto módulo $x^n - a$, sendo a um elemento não-nulo de GF(p), resulta no polinômio $c'(x) = ac_{n-1} + c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1}$. Diz-se que $c'(x)$ corresponde a um deslocamento *constacíclico* de $c(x)$ para a direita [12]. Assim, temos a seguinte definição para código constacíclico.

Definição 3.1 – Códigos constacíclicos

*Um código C é dito ser um **código constacíclico** se qualquer **deslocamento constacíclico** de uma palavra-código resulta em uma palavra-código.* □

Neste ponto, é possível perceber que a classe de códigos constacíclicos contém a classe de códigos cíclicos, para isto basta fazer $a = 1$ em $x^n - a$. Além do mais, os *códigos negacíclicos* [12], obtidos fazendo $a = -1$ em $x^n - a$, também são um caso particular dos códigos constacíclicos.

3.1.1 Códigos Constacíclicos de Comprimento $p + 1$

A construção de códigos constacíclicos pode ser obtida como uma generalização da construção de códigos cíclicos. Sendo assim, as palavras-código de um código constacíclico linear p-ário (n, k) são reduzidas $\mathrm{mod}(x^n - a)$, $a \in \mathrm{GF}(p)$, e o polinômio gerador $g(x)$, de grau $(n-k)$, é fator de $x^n - a$. Os polinômios-código são da forma $c(x) = m(x)g(x)$, em que $m(x)$ é o polinômio-mensagem de grau menor ou igual a $k - 1$.

Nesta tese, estamos interessados na construção de códigos constacíclicos de comprimento $n = p + 1$, sendo p um número primo tal que $p > 3$, e com o elemento a de $x^n - a$ sendo um elemento gerador do grupo multiplicativo de $\mathrm{GF}(p)$. Esta escolha deve-se ao fato de que algumas propriedades são conhecidas para códigos constacíclicos com esses parâmetros [49]. Uma descrição da existência de códigos constacíclicos para outros valores de n e a pode ser encontrada em [52]. Daqui por diante e até que se informe o contrário, o elemento a do polinômio $x^n - a$ é um elemento gerador do grupo multiplicativo de $\mathrm{GF}(p)$.

Para $n = p+1$ e a um elemento gerador do grupo multiplicativo de $\mathrm{GF}(p)$, é conhecido [49] que as raízes de $x^{p+1} - a$ pertencem a $\mathrm{GF}(p^2)$ e podem ser escritas na forma $\alpha^{1+(p-1)i}$, para $0 \leq i \leq p$, ou ainda na forma $\alpha^{p+(p-1)i}$, para $-(p-1)/2 \leq i \leq (p+1)/2$. De acordo com [49], o polinômio $x^{p+1} - a$ é fatorado em $(p+1)/2$ polinômios de grau dois. Logo, as raízes de $x^{p+1} - a$ pertencem a classes conjugadas de cardinalidade dois. Sendo assim, também é possível representá-las por meio de seus expoentes como $\{1-(p-1)i,\ p+(p-1)i\}$, para $0 \leq i \leq (p-1)/2$. Uma consequência deste fato é que o grau dos possíveis polinômios geradores $g(x)$ é um número par, isto é, $n - k$ sempre é par. Como $n = p + 1$ também é par, a dimensão do código, k, sempre será um número par entre $2 \leq k \leq p - 1$.

Exemplo 3.1

Considere $p = 5$. *Os elementos de* $\mathrm{GF}(5)$ *são* $\{0, 1, 2, 3, 4\}$ *sendo* 2 *e* 3 *elementos geradores do grupo multiplicativo de* $\mathrm{GF}(5)$, *ou seja, elementos cuja ordem multiplicativa é igual a* $p - 1 = 4$. *Desta forma, podemos construir códigos constacíclicos* 5*-ários* $(n = p + 1, k)$ *cujos polinômios-código são reduzidos* $\mathrm{mod}\,(x^6 - 2)$ *ou* $\mathrm{mod}\,(x^6 - 3)$. *Como a dimensão do código,* k, *é um número par entre* $2 \leq k \leq 4$, *os possíveis códigos constacíclicos* 5*-ários possuem parâmetros* $(6, 2, d_{\min})$ *e*

Tabela 3.1: *Classes conjugadas e polinômios mínimos sobre* GF(5) *para* $x^6 - 3$

Classe conjugada	Polinômio mínimo
$\{1, 5\}$	$M_1(x) = 3 + 2x + x^2$
$\{9, 21\}$	$M_9(x) = 3 + x^2$
$\{13, 17\}$	$M_{13}(x) = 3 + 3x + x^2$

$(6, 4, d_{\min})$. □

Analisa-se, neste ponto, os resultados discutidos no parágrafo anterior para o polinômio $x^6 - 3$ do Exemplo 3.1. Como $p = 5$, considere o corpo de extensão GF(25) gerado por $p(x) = 3 + 2x + x^2$ que é um polinômio primitivo sobre GF(5). Além do mais, considere α um elemento gerador do grupo multiplicativo de GF(25), tal que $p(\alpha) = 3 + 2\alpha + \alpha^2 = 0$. Portanto, as raízes de $x^6 - 3$ em GF(25) escritas na forma α^{1+4i}, para $0 \leq i \leq 5$, são $\{\alpha, \alpha^5, \alpha^9, \alpha^{13}, \alpha^{17}, \alpha^{21}\}$. Alternativamente, escrevendo as raízes pela forma α^{5+4i}, para $-2 \leq i \leq 3$, obtemos $\{\alpha^{17}, \alpha^{21}, \alpha, \alpha^5, \alpha^9, \alpha^{13}\}$. As classes conjugadas para as raízes de $x^6 - 3$ são $\{1, 5\}$, $\{9, 21\}$ e $\{13, 17\}$ e todas as classes possuem cardinalidade dois. Os polinômios mínimos associados a cada uma das classes conjugadas podem ser vistos na Tabela 3.1. Desta forma, o polinômio $x^6 - 3$ é fatorado em três, $(5+1)/2 = 3$, polinômios de grau dois, ou seja, $x^6 - 3 = M_1(x)M_9(x)M_{13}(x)$.

Observando o conjunto de raízes $\{\alpha, \alpha^5, \alpha^9, \alpha^{13}, \alpha^{17}, \alpha^{21}\}$ percebe-se que os expoentes das raízes formam uma progressão aritmética (PA) de razão 4, no caso geral $p-1$, e com primeiro termo 1 ou, ainda, pode-se dizer que estas raízes formam uma progressão geométrica (PG) de razão α^4, no caso geral α^{p-1}, cujo primeiro termo é α. Como é provado em [49], ocorre que, em geral, pode-se escolher um polinômio gerador $g(x)$ com $2t$ raízes consecutivas, pois elas são termos consecutivos de uma PG de razão α^{p-1}, e construir um código constacíclico corretor de t erros cuja distância mínima é limitada inferiormente por $d_{\min} \geq 2t + 1$. É válido salientar que os expoentes de α, nestes casos, são reduzidos $\bmod(p^2 - 1)$. Isto implica, por exemplo, que as raízes α^{21} e α são consideradas consecutivas neste contexto. Esta cota inferior para a distância mínima dos códigos constacíclicos é semelhante à cota BCH do Teorema 2.10. Como o grau do polinômio gerador $g(x)$ é igual a $n - k$ e este polinômio possui $2t$ raízes, $n - k = 2t$ e, assim, $d_{\min} \geq n - k + 1$. Porém, pela cota de Singleton, $d_{\min} \leq 2t + 1$ ou $d_{\min} \leq n - k + 1$. Portanto, desde que o polinômio gerador $g(x)$ possua $2t$ raízes consecutivas de $x^{p+1} - a$, o código gerado por $g(x)$ tem $d_{\min} = n - k + 1$ (MDS) para satisfazer a cota BCH

e a cota de Singleton simultaneamente. A vantagem de construir códigos constacíclicos MDS é que a distribuição dos pesos das palavras-código é conhecida, Fórmula (2.19). Para códigos constacíclicos que não são MDS, não há uma fórmula fechada para obter a distribuição de pesos das palavras-código.

Exemplo 3.2

Para gerar os códigos $(6, 4, d_{\min})$ *do Exemplo 3.1 o grau de* $g(x)$ *é igual a* $n - k = 2$*, logo se tem três opções para o polinômio gerador, são elas:* $g(x) = M_i(x)$ *para* $i = \{1, 9, 13\}$ *(Tabela 3.1). Para* $g_1(x) = M_1(x)$ *e* $g_2(x) = M_{13}(x)$ *os polinômios geradores possuem duas raízes que são termos consecutivos da PG* $\{\alpha, \alpha^5, \alpha^9, \alpha^{13}, \alpha^{17}, \alpha^{21}\}$*, logo* $d_{\min} = 3$ *e estes códigos são MDS com parâmetros* $(n, k, d_{\min}) = (6, 4, 3)$*. Entretanto, para* $g_3(x) = M_9(x)$ *as raízes são* α^9 *e* α^{21} *e estas não são termos consecutivos da PG, logo o código não é MDS. Neste caso,* $d_{\min} = 2$ *e o código possui parâmetros* $(n, k, d_{\min}) = (6, 4, 2)$*.* □

Baseando-se no Exemplo 3.2, para construir os códigos constacíclicos 5-ários de parâmetros $(6, 2, d_{\min})$, previstos no Exemplo 3.1, o grau do polinômio gerador $g(x)$ tem de ser igual a $n-k = 4$. De acordo com a Tabela 3.1, polinômios geradores de grau 4 são obtidos quando $g(x) = M_i(x)M_j(x)$, $i \neq j$ e $i, j = \{1, 9, 13\}$. Logo, os possíveis polinômios geradores são $g_1(x) = M_1(x)M_9(x)$, $g_2(x) = M_1(x)M_{13}(x)$ e $g_3(x) = M_9(x)M_{13}(x)$. Os polinômios $g_1(x)$ e $g_3(x)$ possuem quatro raízes que são termos consecutivos da PG, logo os códigos gerados por eles são MDS de parâmetros $(6, 2, 5)$. O código gerado por $g_2(x)$ não possui quatro raízes como termos consecutivos da PG e, neste caso, há duas raízes consecutivas que são as raízes de $M_1(x)$ ou, equivalentemente, as de $M_{13}(x)$, logo este código possui parâmetros $(6, 2, 3)$.

3.1.2 Ordem Constacíclica das Palavras-Código

Definição 3.2 – Ordem constacíclica

Considere um código constacíclico linear p-ário $(n, k, d_{\min})$ *em que as palavras-código são reduzidas* $\mathrm{mod}(x^n - a)$*, em que* a *é um elemento não nulo de* $\mathrm{GF}(p)$*. A* ***ordem constacíclica*** *de uma palavra-código é o menor inteiro positivo* i *tal que* $x^i c(x) = c(x) \bmod (x^n - a)$*.* □

Analogamente ao que ocorre com os códigos cíclicos, multiplicar um polinômio-código $c(x)$ por x^i equivale a deslocar constaciclicamente (Definição 3.1) em i posições a palavra-código **c**.

Tabela 3.2: *Deslocamentos constacíclicos de* $g(x) = 4 + 2x^2 + x^4 \leftrightarrow \mathbf{g} = (4, 0, 2, 0, 1, 0)$.

i	$x^i g(x) \bmod (x^6 - 3)$
0	$(4, 0, 2, 0, 1, 0)$
1	$(0, 4, 0, 2, 0, 1)$
2	$(3, 0, 4, 0, 2, 0)$
3	$(0, 3, 0, 4, 0, 2)$
4	$(1, 0, 3, 0, 4, 0)$
5	$(0, 1, 0, 3, 0, 4)$
6	$(2, 0, 1, 0, 3, 0)$
7	$(0, 2, 0, 1, 0, 3)$
8	$(4, 0, 2, 0, 1, 0)$

Exemplo 3.3

Considere um código constacíclico 5*-ário* $(6, 2, 3)$ *gerado por* $g(x) = M_1(x)M_{13}(x) = 4+2x^2+x^4$ *(Vide Tabela 3.1) e que os polinômios-código são reduzidos* $\bmod (x^6 - 3)$. *Como* $g(x)$ *é um polinômio-código, ele pode ser representado pela palavra-código* $\mathbf{g} = (4, 0, 2, 0, 1, 0)$. *A Tabela 3.2 mostra os deslocamentos constacíclicos de* $\mathbf{g}$. *Logo, a ordem constacíclica de* $g(x)$ *é* 8.

Exemplo 3.4

De maneira semelhante ao Exemplo 3.3, pode-se mostrar que o polinômio gerador $g(x) = M_1(x)M_9(x) = 4 + x + x^2 + 2x^3 + x^4 \leftrightarrow \mathbf{g} = (4, 1, 1, 2, 1, 0)$, *o qual gera um código constacíclico* 5*-ário* $(6, 2, 5)$ *com os polinômios-código reduzidos* $\bmod (x^6 - 3)$, *possui ordem constacíclica* 24. □

Definição 3.3 – Ordem constacíclica plena

Uma palavra-código, pertencente a um código constacíclico, com $n = p+1$, *tem* ***ordem constacíclica plena*** *quando o menor valor de* i *tal que* $x^i c(x) = c(x) \bmod (x^n - a)$ *é* $i = p^2 - 1$. □

Neste ponto, pode-se notar que, diferentemente do que ocorre com os códigos cíclicos cuja ordem cíclica é um divisor de n (comprimento do código), a ordem constacíclica pode ser maior que o comprimento, $n = p + 1$, do código constacíclico. De fato, no caso dos códigos cíclicos BCH primitivos, por exemplo, as raízes do polinômio $x^{q^m-1} - 1$ estão em GF(q^m), logo a ordem multiplicativa das raízes do polinômio gerador $g(x)$ de um código BCH primitivo é igual ao comprimento do código $n = q^m - 1$ ou a um dos seus divisores. Portanto, a ordem cíclica não será maior que o comprimento do código. Mas, no caso dos

códigos constacíclicos, as raízes do polinômio $x^{p+1} - a$ estão em GF(p^2), portanto a ordem multiplicativa das raízes do polinômio gerador $g(x)$ de um código constacíclico é igual a $p^2 - 1$ ou a um dos seus divisores. Desta forma, a ordem constacíclica pode ser maior que o comprimento do código. Para $p = 5$, por exemplo, temos $p^2 - 1 = 24$, logo os possíveis valores para a ordem constacíclica das palavras dos códigos $(6, 2, 3)$ e $(6, 2, 5)$ são $1, 2, 3, 4, 6, 8, 12, 24$. Observando os Exemplos 3.3 e 3.4, percebe-se que este resultado é satisfeito.

A relação entre a ordem multiplicativa das raízes do polinômio $x^{p+1} - a$, consequentemente das raízes de $g(x)$, e a ordem constacíclica das palavras de um código constacíclico é explorada na construção destes códigos tal que o maior número possível de palavras-código tenha ordem constacíclica plena. Essa característica é interessante, pois maximiza o número de possíveis palavras de um código constacíclico que podem ser usadas na construção de códigos cíclicos binários como é feito no Capítulo 4.

Voltando ao Exemplo 3.4, o código constacíclico $(6, 2, 5)$ tem dimensão $k = 2$, logo ele possui $p^k = 5^2 = 25$ palavras-código. Como o código é linear, a palavra toda nula $\mathbf{0}$ pertence ao código e, pela Definição 3.2, possui ordem constacíclica igual a 1. Sendo assim, restam 24 palavras-código não-nulas. Conforme o Exemplo 3.4, a palavra-código $\mathbf{g} = (4, 1, 1, 2, 1, 0)$ possui ordem constacíclica 24. Logo, as 24 palavras-código restantes do código $(6, 2, 5)$, gerado por $g(x) = 4 + x + x^2 + 2x^3 + x^4$, correspondem à palavra-código $\mathbf{g}$ e seus deslocamentos constacíclicos. Com esse exemplo, é possível deduzir que o maior número de palavras-código com ordem constacíclica plena que é possível obter para um código constacíclico é igual a $p^k - 1$, pois como estes códigos são lineares, a palavra toda nula sempre pertence ao código. Em contrapartida, o código constacíclico do Exemplo 3.3, que também possui 25 palavras-código, é composto pela palavra-código $\mathbf{0}$ e por outras 24 palavras-código de ordem constacíclica 8. A Tabela 3.3 mostra as palavras não-nulas do código $(6, 2, 3)$ do Exemplo 3.3. Nela pode-se observar como as palavras-código não-nulas podem ser obtidas por intermédio do deslocamento constacíclico das palavras-código $\mathbf{c}_1$, $\mathbf{c}_2$ e $\mathbf{c}_3$.

Conforme mencionado, a relação entre a ordem multiplicativa das raízes do polinômio $x^{p+1} - a$ e a ordem constacíclica das palavras-código é de fundamental importância na construção de códigos constacíclicos com o maior número possível de palavras-código com ordem constacíclica plena. O Teorema 3.1, enunciado a seguir, é uma contribuição desta tese. Ele é equivalente ao Teorema 2.8 para códigos cíclicos e mostra como escolher o polinômio gerador $g(x)$ tal que o código constacíclico gerado por ele tenha $p^k - 1$ palavras-código com ordem

Tabela 3.3: *Palavras não-nulas do código* $(6, 2, 3)$ *gerado por* $g(x) = 4 + 2x^2 + x^4$ *(Exemplo 3.3) . A primeira coluna corresponde à quantidade de deslocamentos constacíclicos para direita.*

i	$\mathbf{c}_1 = (4,0,2,0,1,0)$	$\mathbf{c}_2 = (2,1,1,3,3,4)$	$\mathbf{c}_3 = (3,1,4,3,2,4)$
1	$(0,4,0,2,0,1)$	$(2,2,1,1,3,3)$	$(2,3,1,4,3,2)$
2	$(3,0,4,0,2,0)$	$(4,2,2,1,1,3)$	$(1,2,3,1,4,3)$
3	$(0,3,0,4,0,2)$	$(4,4,2,2,1,1)$	$(4,1,2,3,1,4)$
4	$(1,0,3,0,4,0)$	$(3,4,4,2,2,1)$	$(2,4,1,2,3,1)$
5	$(0,1,0,3,0,4)$	$(3,3,4,4,2,2)$	$(3,2,4,1,2,3)$
6	$(2,0,1,0,3,0)$	$(1,3,3,4,4,2)$	$(4,3,2,4,1,2)$
7	$(0,2,0,1,0,3)$	$(1,1,3,3,4,4)$	$(1,4,3,2,4,1)$

constacíclica plena.

Teorema 3.1

Seja o polinômio $x^{p+1} - a$*, em que* p *é um número primo,* $p > 3$*, e* $a \neq 0$ *é um elemento gerador do grupo multiplicativo de* $\mathrm{GF}(p)$*. Assuma que pelo menos um par de raízes conjugadas de* $x^{p+1} - a$ *tenha ordem multiplicativa* $p^2 - 1$*. Todas as palavras-código não-nulas de um código* $\mathcal{C}$ *constacíclico linear* p*-ário* $(p + 1, k, d_{min})$ *possuem ordem constacíclica plena se, e somente se, todas as raízes do polinômio* $x^{p+1} - a$ *que não têm ordem multiplicativa igual a* $p^2 - 1$ *sao escolhidas como raízes do polinômio gerador* $g(x)$*.* □

Demonstração: Suponha que todas as raízes de $x^{p+1} - a$ que possuem ordem multiplicativa diferente de $p^2 - 1$ estão em $g(x)$. Para que qualquer $c(x) \in \mathcal{C}$ tenha ordem constacíclica plena, o menor valor de i tal que

$$x^i c(x) \quad = \quad c(x) \bmod (x^{p+1} - a)$$

ou, equivalentemente,

$$(x^i - 1)c(x) \quad = \quad 0 \bmod (x^{p+1} - a) \tag{3.1}$$

tem de ser $i = p^2 - 1$. Entretanto, pode-se escrever $c(x) = m(x)g(x)$ e substituir $c(x)$ por $m(x)g(x)$ em (3.1). Logo,

$$(x^i - 1)m(x)g(x) \quad = \quad 0 \bmod (x^{p+1} - a). \tag{3.2}$$

A condição expressa em (3.2) implica que todas as raízes do polinômio $x^{p+1} - a$ estão presentes em $(x^i - 1)m(x)g(x)$. Como grau$[m(x)g(x)] \leq p$, no mínimo uma raiz de $x^{p+1} - a$ é comum a $x^i - 1$. Por hipótese, todas as raízes de $x^{p+1} - a$ que possuem ordem

multiplicativa diferente de p^2-1 estão em $g(x)$. Desta forma, as raízes de $x^{p+1}-a$ comuns a x^i-1 possuem ordem multiplicativa p^2-1. Assim, $i=p^2-1$ é o valor mínimo para que (3.2) seja satisfeita e, portanto, todas as palavras-código não-nulas de $\mathcal{C}$ possuem ordem constacíclica plena.

Por outro lado, suponha que todas as palavras-código não nulas de $\mathcal{C}$ possuem ordem constacíclica plena, ou seja, p^2-1 é o menor valor de i que satisfaz (3.2). Pelos mesmos argumentos já expostos, no mínimo uma raiz de $x^{p+1}-a$ é comum a x^i-1 em (3.2). Se esta raiz comum, denotada por α^{i_1}, tem ordem multiplicativa $i_2 < p^2-1$, tal que $\alpha^{i_1 i_2} = \alpha^{p^2-1} = 1$, então (3.2) é satisfeita para $i = i_2$. Porém, isto implica que pelo menos uma palavra-código possui ordem constacíclica $i_2 < p^2-1$. Como isto não é possível, dada a hipótese assumida que $i = p^2-1$ é o menor valor para o qual (3.2) é satisfeita, então qualquer raiz com ordem multiplicativa $i < p^2-1$ deve estar no conjunto de raízes de $m(x)$ ou de $g(x)$. Uma vez que o polinômio $m(x)$ pode conter ou não raízes em GF(p^2), o único modo de garantir a hipótese assumida é que, em geral, todas as raízes com ordem multiplicativa $i < p^2-1$ devem estar no conjunto de raízes de $g(x)$. ■

É interessante utilizar os Exemplos 3.3 e 3.4 para ilustrar os resultados enunciados no Teorema 3.1. Os códigos daqueles exemplos possuem polinômios geradores que são fatores de x^6-3, que por sua vez, possui quatro raízes com ordem multiplicativa 24, $\{\alpha, \alpha^5, \alpha^{13}, \alpha^{17}\}$ e duas raízes com ordem multiplicativa 8, $\{\alpha^9, \alpha^{21}\}$. No Exemplo 3.3, o código constacíclico $(6,2,3)$ é gerado por $g(x) = M_1(x)M_{13}(x)$ cujas raízes são $\{\alpha, \alpha^5, \alpha^{13}, \alpha^{17}\}$. Observe que as raízes de x^6-3 que não têm ordem multiplicativa 24 não fazem parte de $g(x)$. Logo, este código não satisfaz a condição para $g(x)$ dada no Teorema 3.1. Além do mais, este código possui todas as palavras-código não-nulas com ordem constacíclica 8 (Vide Tabela 3.3). Entretanto, no Exemplo 3.4, o código constacíclico $(6,2,5)$ é gerado por $g'(x) = M_1(x)M_9(x)$ e, neste caso, as raízes de x^6-3 que não têm ordem multiplicativa 24 estão em $g'(x)$. Logo, todas as palavras deste código possuem ordem constacíclica plena (24). Este resultado já era esperado, pois as palavras-código não nulas do código $(6,2,5)$ gerado por $g'(x) = M_1(x)M_9(x)$ são formadas pelos deslocamentos constacíclicos de $g'(x)$ cuja ordem constacíclica é plena. Ainda pelo Teorema 3.1, o código constacíclico gerado por $g''(x) = M_9(x)M_{13}(x)$ (Tabela 3.1) possui todas as palavras-código não-nulas com ordem constacíclica plena.

Embora o Teorema 3.1 garanta uma maneira de construir códigos constacíclicos cujas palavras-código não-nulas tenham ordem constacíclica plena, não se tem a garantia de que

estes códigos também são MDS, já que a escolha do polinômio gerador para garantir que ele seja MDS está relacionada com a escolha das raízes consecutivas de $x^{p+1} - a$. Para construir códigos constacíclicos que sejam MDS e que possuam todas as palavras-código com ordem constacíclica plena, limitam-se as possíveis escolhas para o polinômio gerador desses códigos, visto que se deve selecionar raízes de $x^{p+1} - a$ que sejam consecutivas e que possuam ordem multiplicativa diferente de $p^2 - 1$. Nem sempre é possível escolher polinômios geradores com essas características. Porém, há um caso particular do Teorema 3.1 [32] que garante a construção de códigos constacíclicos MDS cujas palavras-código não-nulas tenham ordem constacíclica plena.

Teorema 3.2

Se $p = 2^m - 1$ é um primo de Mersenne [30], então todas as raízes de $x^{p+1} - a$ possuem ordem multiplicativa igual a $p^2 - 1$. □

Demonstração: As raízes de $x^{p+1} - a$ pertencem a GF(p^2) e podem ser escritas na forma $\alpha^{1+(p-1)i}$, para $0 \leq i \leq p$, em que α é um elemento gerador do grupo multiplicativo de GF(p^2), ou seja, a ordem multiplicativa de α é igual a $\text{ord}(\alpha) = p^2 - 1$. Fazendo $\beta = \alpha^{1+(p-1)i}$, a ordem multiplicativa de β é dada por [10, pág. 35]

$$\begin{aligned} \text{ord}(\beta) &= \frac{\text{ord}(\alpha)}{\text{mdc}\,[\text{ord}(\alpha), 1 + (p-1)i]} \\ &= \frac{p^2 - 1}{\text{mdc}\,[p^2 - 1, 1 + (p-1)i]}. \end{aligned} \tag{3.3}$$

Substituindo $p = 2^m - 1$ no denominador de (3.3)

$$\begin{aligned} \text{mdc}\left[2^{2m} - 2^{m+1}, 1 + (2^m - 2)i\right] &= \text{mdc}\,[2^m(2^m - 2), 1 + (2^m - 2)i] \\ &= 1. \end{aligned}$$

Pois, $1 + (2^m - 2)i$ não possui fatores comuns com 2^m ou $2^m - 2$. Portanto, todas as raízes de $x^{p+1} - a$ possuem ordem multiplicativa $\text{ord}(\beta) = p^2 - 1$ quando p é um primo de Mersenne. ■

Corolário 3.1 – Teorema 3.1 e Teorema 3.2

Se $p = 2^m - 1$ é um primo de Mersenne, então todas as palavras-código não-nulas de um código constacíclico p-ário $(p + 1, k, d_{\min})$, cujo polinômio gerador $g(x)$ é fator de $x^{p+1} - a$, possuem ordem constacíclica plena. □

O Corolário 3.1 garante que se $p = 2^m - 1$ é um primo de Mersenne, não é necessário escolher para o polinômio gerador $g(x)$ de um código constacíclico as raízes de $x^{p+1} - a$ que possuam ordem multiplicativa diferente de $p^2 - 1$, pois todas as raízes, neste caso, são elementos geradores do grupo multiplicativo de GF(p^2). Desta forma, se são selecionadas $2t$ raízes consecutivas de $x^{p+1} - a$ para o polinômio gerador de um código constacíclico, então é obtido um código constacíclico MDS com todas as palavras-código não-nulas com ordem constacíclica plena.

3.1.3 Classes de Equivalência Constacíclica

Definição 3.4 – Classe de equivalência constacíclica

Considere dois polinômios-código $c_1(x)$ e $c_2(x)$ pertencentes a um código constacíclico q-ário C cujos polinômios-código são reduzidos $\mathrm{mod}(x^{q+1} - a)$. *Diz-se que $c_1(x)$ e $c_2(x)$ pertencem à mesma* ***classe de equivalência constacíclica*** *se $x^i c_1(x) = c_2(x) \bmod (x^{q+1} - a)$ para $1 \leq i < q^2 - 1$. Se $c_1(x)$ tem ordem constacíclica igual a j, então a classe de equivalência constacíclica que contém $c_1(x)$ possui j polinômios-código, que correspondem aos deslocamentos constacíclicos de $c_1(x)$, e a classe de equivalência constacíclica, da qual $c_1(x)$ agora é denominado* ***líder****, também tem ordem constacíclica igual a j.* □

Decorre da Definição 3.4 que a palavra-código **0** constitui uma classe de equivalência constacíclica de ordem igual a 1. Além do mais, qualquer palavra-código pertencente a uma mesma classe de equivalência pode ser definida como líder de sua classe.

Pode-se exemplificar o uso da Definição 3.4 utilizando a Tabela 3.3 na qual estão todas as palavras não-nulas do código do Exemplo 3.3. As palavras-código $\mathbf{c}_1$, $\mathbf{c}_2$ e $\mathbf{c}_3$ são líderes de suas respectivas classes de equivalência constacíclica. Essas palavras-código possuem todas ordem constacíclica igual a 8, logo cada uma dá origem, por meio de seus deslocamentos constacíclicos, a uma classe de equivalência com ordem constacíclica igual a 8 e com oito elementos cada. Portanto, o código do Exemplo 3.3 possui uma classe de equivalência constacíclica com ordem constacíclica igual a 1 e três classes de equivalência com ordem constacíclica 8. Para finalizar, o código dado no Exemplo 3.4 possui duas classes de equivalência constacíclica, uma que tem a palavra-código **0** como líder e, portanto, tem ordem constacíclica igual a 1 e outra com ordem constacíclica igual a 24 a qual tem como líder o polinômio gerador do código.

Capítulo 4

Construção de Códigos Ciclicamente Permutáveis

Eu ouço, eu esqueço. Eu vejo, eu lembro. Eu faço, eu aprendo.

— Confúcio

Neste capítulo são utilizadas propriedades algébricas de códigos cíclicos e de códigos constacíclicos para construir novos códigos ciclicamente permutáveis.

4.1 Códigos Ciclicamente Permutáveis

Códigos ciclicamente permutáveis (códigos CP) foram introduzidos por Gilbert [17] em 1963. A definição a seguir é semelhante àquela introduzida em [18]

Definição 4.1

*Um **código ciclicamente permutável** é um código de bloco binário de comprimento N em que cada palavra-código tem ordem cíclica plena e tal que as palavras-código são ciclicamente distintas.* □

A Definição 4.1 garante que dada uma palavra do código, por exemplo, $\mathbf{c}$, nenhuma outra palavra deste código pode ser obtida deslocando-se ciclicamente a palavra-código $\mathbf{c}$. Embora

seja comum na literatura encontrar a definição para códigos CP como um código binário, ela pode ser generalizada diretamente para o caso de códigos q-ários. Em [27], por exemplo, códigos CP não-binários são aplicados em sistemas DS-CDMA (*Direct Sequence Code Division Multiple Access*) com estações base assíncronas.

O conceito de classe de equivalência constacíclica (Definição 3.4), definido para as palavras de um código constacíclico, pode ser aplicado de maneira semelhante às palavras de um código cíclico ou para uma N-upla, em geral. Desta forma, sendo $\mathbf{b}$ uma N-upla l-ária, em que l denota um número inteiro, pode-se definir *classe de equivalência cíclica* como sendo o conjunto de N-uplas cujos elementos correspondem a todos os deslocamentos cíclicos distintos de $\mathbf{b}$. Definindo o operador $\mathbf{S}^i(.)$, o qual aplicado a uma N-upla a desloca ciclicamente de i posições para a direita, então duas N-uplas, $\mathbf{b}$ e $\mathbf{b}'$, pertencem à mesma classe de equivalência cíclica se e só se $\mathbf{S}^i(\mathbf{b}) = \mathbf{b}'$ para algum valor de i, $1 \leq i \leq N-1$. Se $\mathbf{b}$ tem ordem cíclica j, então a classe de equivalência a qual $\mathbf{b}$ pertence tem j N-uplas e, portanto, esta classe tem ordem j. Desta forma, um código CP de comprimento de bloco N pode ser definido, alternativamente, como um código de bloco q-ário tal que suas palavras-código pertencem a diferentes classes de equivalência cíclica, cada uma com ordem igual a N.

Exemplo 4.1

Um código de bloco constituído pelas palavras-código $\{(0,0,0,1);(0,0,1,1);(0,1,1,1)\}$ *é um código CP. Observe que todas as palavras-código tem ordem cíclica plena, igual a 4, e nenhum deslocamento cíclico de uma palavra-código gera outra palavra-código.* □

A *distância mínima cíclica* de um código CP de comprimento de bloco N, denotada por d_c, é definida como a menor distância de Hamming entre uma palavra-código $\mathbf{c}$ e seus deslocamentos cíclicos $\mathbf{S}^i(\mathbf{c})$, $1 \leq i \leq N-1$, ou os deslocamentos cíclicos de uma outra palavra-código $\mathbf{S}^i(\mathbf{c}')$. Por exemplo, para determinar a distância mínima cíclica do código CP do Exemplo 4.1, deve-se construir as classes de equivalência cíclica, geradas por meio de cada uma das palavras deste código CP, e encontrar a menor distância de Hamming entre um par de 4-uplas de um total de doze 4-uplas binárias. Na Tabela 4.1 são exibidas as classes de equivalência cíclica para as palavras do código CP do Exemplo 4.1. Neste caso, $d_c = d\{(0,0,1,1);(0,1,1,1)\} = 1$. Observando atentamente a Tabela 4.1, percebe-se que o conjunto de doze 4-uplas constitui um código cíclico binário, pois, de acordo com a Definição 2.7, os deslocamentos cíclicos de qualquer uma das doze 4-uplas pertence ao código. Isto implica que o procedimento para calcular a distância mínima cíclica d_c do código CP do Exemplo 4.1, realizado anteriormente, é equiva-

Tabela 4.1: *Classes de equivalência cíclica para as palavras do código CP do Exemplo 4.1.*

i	$(0,0,0,1)$	$(0,0,1,1)$	$(0,1,1,1)$
1	$(1,0,0,0)$	$(1,0,0,1)$	$(1,0,1,1)$
2	$(0,1,0,0)$	$(1,1,0,0)$	$(1,1,0,1)$
3	$(0,0,1,0)$	$(0,1,1,0)$	$(1,1,1,0)$

lente a calcular a distância mínima $d_{\min}$ do código cíclico binário composto pelas doze 4-uplas da Tabela 4.1. Portanto, em geral, a distância mínima cíclica d_c é igual a distância mínima $d_{\min}$ do código cíclico binário obtido ao gerar as classes de equivalência cíclica para cada uma das palavras do código CP. Daqui em diante, denota-se um código CP por $\mathrm{CCP}(N, M_c, d_c)$ em que N é o comprimento do bloco, M_c é o número de palavras-código e d_c é a distância mínima cíclica.

Em geral, dado um código cíclico q-ário $(n, k, d_{\min})$, se seu dicionário for particionado em classes de equivalência cíclica e, no máximo, uma palavra-código de cada uma das distintas classes de equivalência cíclica, de ordem n, for selecionada, então obtém-se um $\mathrm{CCP}(N, M_c, d_c)$ com $d_c \geq d_{\min}$ e M_c igual ao número de palavras selecionadas do código cíclico. Isto implica que se pode obter códigos CP por meio dos códigos lineares cíclicos q-ários estudados no Capítulo 2. Um procedimento direto para realizar essa tarefa é denominado de *busca exaustiva* e é descrito na sequência. Inicialmente, escolhe-se, arbitrariamente, uma palavra $\mathbf{c}$ do código para ser uma palavra do código CP. Depois, outra palavra do código $\mathbf{c}'$ é escolhida e comparam-se todos os deslocamentos cíclicos de $\mathbf{c}'$, $\mathbf{S}^i(\mathbf{c}')$, com a palavra-código $\mathbf{c}$. Se $\mathbf{S}^i(\mathbf{c}') = \mathbf{c}$, para algum i tal que $1 \leq i \leq N-1$, então a palavra é descartada, caso contrário, $\mathbf{c}'$ é escolhida como uma palavra do código CP. Este processo continua até que todas as q^k palavras do código sejam testadas. Se o número de palavras do código é elevado, então o processo de seleção por busca exaustiva torna-se ineficiente.

Conforme dito na Seção 1.3, um procedimento que seleciona diretamente as palavras de um código CP, por meio de uma condição matemática, a partir de um código cíclico pode ser qualificado como uma construção. Dado um código de bloco cíclico q-ário com M palavras-código e comprimento de bloco n, se o número de palavras do código CP for igual ao limitante superior M/n, então a construção é ótima neste sentido. Para um código linear cíclico q-ário com parâmetros $(n, k, d_{\min})$, o limitante superior é $(q^k - 1)/n$, uma vez que o código é linear, $M = q^k$ e a palavra toda nula pertence ao código, além de possuir ordem cíclica 1 (Definição 2.8), menor que n. Já para um código linear constacíclico p-ário com comprimento

de bloco $n = p + 1$, o limitante superior é $(p^k - 1)/(p^2 - 1)$, uma vez que o numerador $p^k - 1$ justifica-se pelo fato do código ser linear e, no caso do denominador, $p^2 - 1$ corresponde à ordem constacíclica plena de uma palavra-código para o caso em que $n = p+1$ (Definição 3.3).

De forma resumida, a metodologia utilizada em [18], [21], [27], [29] para gerar códigos CP consiste em, dado um código linear cíclico q-ário, selecionar as palavras-código que possuem ordem cíclica plena e que são ciclicamente distintas. Na sequência são apresentadas as construções de códigos CP propostas nesta tese e que seguem a metodologia apresentada em [18], [21], [27], [29]. Na Seção 4.2, códigos CP são construídos por meio de códigos lineares cíclicos q-ários cujos resultados foram originalmente publicados em [42]. Na Seção 4.3, códigos CP são construídos por meio de códigos lineares constacíclicos q-ários cujos resultados foram originalmente publicados em [33], [53]. Porém, um teorema provado na Seção 4.3 permite selecionar um maior número de classes de equivalência em comparação ao resultado publicado em [33], [53].

4.2 Códigos CP construídos por meio de Códigos Cíclicos

Nesta seção, mostra-se que o procedimento usado nesta tese para selecionar as palavras de um código CP por meio de códigos lineares cíclicos q-ários é direto, logo pode ser qualificado como construção. Além do mais, mostra-se que tal construção é ótima no sentido que o número de classes de equivalência cíclica geradas é precisamente $(q^k - 1)/n$, que é o limitante superior para a classe de códigos em questão.

No Capítulo 2, o Teorema 2.8 estabelece uma condição que permite construir códigos lineares cíclicos q-ários tal que todas as palavras-código não-nulas possuem ordem cíclica plena. Por meio do Teorema 4.1, enunciado na sequência e publicado originalmente em [42], dado um código linear cíclico q-ário $(n, k, d_{\min})$ cujo polinômio gerador $g(x)$ satisfaz o Teorema 2.8, é possível selecionar todas as palavras-código não-nulas que são ciclicamente distintas. Desta forma, obtém-se um código CP q-ário de parâmetros $N = q^m - 1$, $M_c = \frac{q^k-1}{q^m-1}$ e $d_c > d_{\min}$.

Teorema 4.1

Seja $n = q^m - 1$ e seja $\mathcal{C}$ um código linear cíclico q-ário $(n, k, d_{\min})$ cujo polinômio gerador $g(x)$ satisfaz o Teorema 2.8, e seja $h(x)$ o polinômio verificação de paridade, em que $h(x) = \prod_{j=1}^{k/m} s_j(x)$, e cada $s_j(x)$, $1 \leq j \leq k/m$, tem grau m e pertence ao expoente n, i.e., n é o menor número inteiro positivo para o qual $s_j(x)$ divide $x^n - 1$. As palavras não nulas $c(x) \in \mathcal{C}$ são ciclicamente distintas

se elas são selecionadas tal que

$$c(x) = g(x)[1 + s_i(x)m_i(x)] \prod_{j=1}^{i-1} s_j(x) \mod x^n - 1, \tag{4.1}$$

em que $1 \leq i \leq k/m, m_i(x)$ *é um polinômio mensagem de grau no máximo* $k - 1 - im \geq 0$, *para* $1 \leq i \leq k/m - 1$, *e* $m_{k/m}(x) = 1$. *O número de classes de equivalência cíclica gerado por (4.1) é precisamente* $(q^k - 1)/n$. □

Demonstração: O teorema é provado em três partes. Nas duas primeiras partes provas-se que as palavras geradas por (4.1) são ciclicamente distintas e, na terceira e última parte, mostra-se quantas classes de equivalência cíclica são geradas.

Na primeira parte, considere $c(x)$ em (4.1) para $1 \leq i \leq k/m - 1$. Neste caso, sejam $c_1(x) = g(x)[1+s_i(x)m_i(x)] \prod_{j=1}^{i-1} s_j(x)$ e $c_2(x) = g(x)[1+s_l(x)m'_l(x)] \prod_{j=1}^{l-1} s_j(x)$, $1 \leq l \leq k/m - 1$, duas palavras-código distintas em $\mathcal{C}$, em que $s_i(x)m_i(x)$ e $s_l(x)m'_l(x)$ possuem grau no máximo igual a $k - 1$. Por hipótese, se $c_1(x)$ e $c_2(x)$ pertencem a mesma classe de equivalência cíclica, então

$$x^t c_1(x) = c_2(x) \mod x^n - 1 \tag{4.2}$$

é satisfeita para algum valor de t tal que $0 < t < n$.

▷ Para $i = l$ e após algumas simplificações em (4.2), obtém-se

$$x^t - 1 + s_i(x)[x^t m_i(x) - m'_i(x)] = 0 \mod s_i(x). \tag{4.3}$$

Para (4.3) ser satisfeita, $s_i(x)$ deve ser um fator de $x^t - 1$, mas isso não é possível visto que $s_i(x)$ pertence ao expoente n e $0 < t < n$. Assim, a hipótese que $c_1(x)$ e $c_2(x)$ pertencem a mesma classe de equivalência cíclica é falsa para $i = l$ e, portanto, $c_1(x)$ e $c_2(x)$ pertencem a classes de equivalência cíclica distintas.

▷ Para $i \neq l$, com $l > i$, e após algumas simplificações em (4.2), obtém-se

$$x^t [1 + s_i(x)m_i(x)] - [1 + s_l(x)m'_l(x)] \prod_{j=i}^{l-1} s_j(x) = 0 \mod \; s_l(x) \prod_{j=i}^{l-1} s_j(x). \tag{4.4}$$

Para (4.4) ser satisfeita, $\prod_{j=i}^{l-1} s_j(x)$ deve dividir $x^t [1 + s_i(x)m_i(x)]$. Como

$$\gcd \left[x^t, \prod_{j=i}^{l-1} s_j(x) \right] = 1,$$

e uma vez que $\gcd[1+s_i(x)m_i(x), s_i(x)] = 1$, conclui-se que $1+s_i(x)m_i(x)$ não é divisível por $\prod_{j=i}^{l-1} s_j(x)$, pois este último tem $s_i(x)$ como fator. Assim, a hipótese que $c_1(x)$ e $c_2(x)$ pertencem a mesma classe de equivalência cíclica é falsa para $i \neq l$ e, portanto, $c_1(x)$ e $c_2(x)$ pertencem a classes de equivalência cíclica distintas.

Na segunda parte da prova, considere $c(x)$ em (4.1) para $i = k/m$. Sendo $m_{k/m}(x) = 1$, obtém-se

$$c(x) = g(x) \prod_{j=1}^{k/m-1} s_j(x) + g(x) \prod_{j=1}^{k/m} s_j(x) \mod x^n - 1.$$

Entretanto, $h(x) = \prod_{j=1}^{k/m} s_j(x)$ e $g(x)h(x) = 0 \mod x^n - 1$. Logo,

$$c(x) = g(x) \prod_{j=1}^{k/m-1} s_j(x)$$

para $i = k/m$. Observe que o resultado é independente do polinômio $m_{k/m}(x)$, assim, por definição, $m_{k/m}(x) = 1$. De modo direto, pode-se verificar que a classe de equivalência cíclica gerada por $c(x) = g(x)\prod_{j=1}^{k/m-1} s_j(x)$ não foi gerada previamente por $c(x)$ em (4.1) para $1 \leq i \leq k/m - 1$. A classe de equivalência cíclica gerada por $g(x)\prod_{j=1}^{k/m-1} s_j(x)$ é precisamente a classe gerada pela divisão do polinômio $x^n - 1$ pelo polinômio $s_{k/m}(x)$.

Na terceira e última parte da demonstração, considere o número total de classes de equivalência cíclica geradas. Para $1 \leq i \leq k/m - 1$ em (4.1), o grau de $m_i(x)$ é menor ou igual a $k-1-im$. Logo, existem q^{k-im} possíveis escolhas para $m_i(x)$. Consequentemente, o número de classes de equivalência cíclica neste caso é dado por $\sum_{i=1}^{k/m-1} q^{k-im}$. Para $i = k/m$ em (4.1), uma única classe de equivalência cíclica é gerada. Desta forma, o número total de classes de equivalência cíclica distintas é dado por $\sum_{i=1}^{k/m-1} q^{k-im} + 1 = \sum_{i=1}^{k/m} q^{k-im}$. Mas, $\sum_{i=1}^{k/m} q^{k-im}$ é a soma de k/m termos de uma série geométrica finita de razão q^{-m}. Assim, $\sum_{i=1}^{k/m} q^{k-im} = \frac{q^k-1}{q^m-1} = (q^k - 1)/n$. ■

Exemplo 4.2

Considere o código linear cíclico binário $(15, 8, 4)$ *do Exemplo 2.5. Uma vez que o polinômio gerador* $g(x) = M_0(x)M_3(x)M_5(x)$ *satisfaz o Teorema 2.8, pode-se aplicar o Teorema 4.1 ao código. Neste caso* $k/m = 2$ *e, desta forma,* $h(x) = \prod_{j=1}^{2} s_j(x) = s_1(x)s_2(x)$, *em que* $s_1(x) = M_1(x)$ *e* $s_2(x) = M_7(x)$ *por exemplo. Para* $i = 1$,

$$c(x) = g(x)[1 + s_1(x)m_1(x)], \tag{4.5}$$

em que grau$[m_1(x)] \leq 8-1-4=3$. *Assim,* $2^4=16$ *classes de equivalência cíclica são geradas. Para* $i=2$, $m_2(x)=1$ *por definição, assim*

$$\begin{aligned} c(x) &= g(x)[1+s_2(x)]s_1(x) \mod x^{15}-1 \\ &= g(x)s_1(x) + \underbrace{g(x)s_1(x)s_2(x)}_{\equiv 0 \bmod x^{15}-1} \\ &= g(x)s_1(x) \end{aligned} \tag{4.6}$$

e, assim, uma classe de equivalência cíclica é obtida.

Portanto, $(2^8-1)/15=17$ *classes de equivalência cíclica são geradas por meio de (4.5) e (4.6).*□

Em [29], um resultado interessante sobre a geração de códigos CP foi estabelecido. Para códigos lineares cíclicos binários com comprimento de bloco $n=2^m-1$, em que n é um primo de Mersenne, é possível selecionar todas as palavras-código com ordem cíclica plena que são ciclicamente distintas, desde que o polinômio $x-1$ não seja um fator do polinômio gerador $g(x)$. Entretanto, no caso em que $x-1$ é um fator de $g(x)$, pelo menos uma classe de equivalência cíclica não é selecionada usando o procedimento em [29]. Além do mais, o procedimento estabelecido em [29] não inclui muitos casos de interesse prático como, por exemplo, códigos lineares cíclicos binários cujo comprimento de bloco não é um primo de Mersenne, assim como códigos lineares cíclicos não-binários e códigos lineares constacíclicos que podem ser efetivamente mapeados para binário [33]. Desta forma, o Teorema 4.1 amplia o número de códigos lineares cíclicos que podem ser usados para gerar códigos CP e, portanto, trata-se de uma contribuição desta tese. Além do mais, sabe-se que para um código linear cíclico q-ário $\mathcal{C}$ com comprimento de bloco n e cujo polinômio gerador $g(x)$ satisfaz o Teorema 2.8, o número de classes de equivalência cíclica distintas é igual a $(q^k-1)/n$. Consequentemente, se $\mathcal{C}$ possui comprimento de bloco $n=q^m-1$, então todas as classes de equivalência cíclica são especificadas de maneira única por meio de (4.1). Como $(q^k-1)/n$ é o limitante superior para o número de classes de equivalência cíclica neste caso, a construção proposta é ótima neste sentido, pois atinge precisamente o limitante.

4.3 Códigos CP construídos por meio de Códigos Constacíclicos

Nesta seção, mostra-se como construir códigos CP por meio dos códigos constacíclicos abordados no Capítulo 3. O objetivo é mostrar que o Teorema 4.1, demonstrado na Seção 4.2 e usado para construir códigos CP por meio de códigos lineares cíclicos q-ários, pode ser

aplicado para os códigos lineares constacíclicos p-ários. Desta forma, assim como no caso dos códigos lineares cíclicos, o procedimento usado para selecionar as palavras do código CP é direto e é ótimo, no sentido que o número de classes de equivalência constacíclica geradas é precisamente $(p^k-1)/(p^2-1)$, que é o limitante superior para a classe de códigos lineares constacíclicos.

No Capítulo 2, o Teorema 3.1 estabelece uma condição que permite construir códigos lineares constacíclicos p-ários tal que todas as palavras-código não-nulas possuem ordem constacíclica plena. Por meio do Teorema 4.2, enunciado na sequência, dado um código linear constacíclico p-ário $(n, k, d_{\min})$ cujo polinômio gerador $g(x)$ satisfaz o Teorema 3.1, é possível selecionar todas as palavras-código não-nulas que são constaciclicamente distintas.

Teorema 4.2

Seja $n = p+1$ e seja $\mathcal{C}$ um código linear constacíclico p-ário $(n, k, d_{\min})$ cujo polinômio gerador $g(x)$ satisfaz o Teorema 3.1, e seja $h(x)$ o polinômio verificação de paridade, em que $h(x) = \prod_{j=1}^{k/2} s_j(x)$, e cada $s_j(x)$, $1 \leq j \leq k/2$, tem grau 2 e pertence ao expoente p^2-1, i.e., p^2-1 é o menor número inteiro positivo n para o qual $s_j(x)$ divide x^n-1. As palavras não nulas $c(x) \in \mathcal{C}$ são constaciclicamente distintas se elas são selecionadas tal que

$$c(x) = g(x)[1 + s_i(x)m_i(x)] \prod_{j=1}^{i-1} s_j(x) \mod x^n - a, \tag{4.7}$$

em que, a é um elemento gerador de $\mathrm{GF}(p)$, $1 \leq i \leq k/2, m_i(x)$ é um polinômio mensagem de grau no máximo $k-1-2i \geq 0$, para $1 \leq i \leq k/2-1$, e $m_{k/2}(x) = 1$. O número de classes de equivalência constacíclica gerado por (4.7) é precisamente $(p^k-1)/(p^2-1)$. □

Demonstração: O teorema é provado em três partes. Nas duas primeiras partes provas-se que as palavras geradas por (4.7) são constaciclicamente distintas e, na terceira e última parte, mostra-se quantas classes de equivalência constacíclica são geradas.

Na primeira parte, considere $c(x)$ em (4.7) para $1 \leq i \leq k/2-1$. Neste caso, sejam $c_1(x) = g(x)[1 + s_i(x)m_i(x)] \prod_{j=1}^{i-1} s_j(x)$ e $c_2(x) = g(x)[1 + s_l(x)m'_l(x)] \prod_{j=1}^{l-1} s_j(x)$, $1 \leq l \leq k/2-1$, duas palavras-código distintas em $\mathcal{C}$, em que $s_i(x)m_i(x)$ e $s_l(x)m'_l(x)$ possuem grau no máximo igual a $k-1$. Por hipótese, se $c_1(x)$ e $c_2(x)$ pertencem à mesma classe de equivalência constacíclica, então

$$x^t c_1(x) = c_2(x) \mod x^n - a \tag{4.8}$$

é satisfeita para algum valor de t tal que $0 < t < p^2-1$.

▷ Para $i = l$, após algumas simplificações em (4.8), obtém-se

$$x^t - 1 + s_i(x)[x^t m_i(x) - m_i'(x)] = 0 \bmod s_i(x). \tag{4.9}$$

Para (4.9) ser satisfeita, $s_i(x)$ deve ser um fator de $x^t - 1$, mas isso não é possível visto que $s_i(x)$ pertence ao expoente $p^2 - 1$ e $0 < t < p^2 - 1$. Assim, a hipótese que $c_1(x)$ e $c_2(x)$ pertencem a mesma classe de equivalência cíclica é falsa para $i = l$ e, portanto, $c_1(x)$ e $c_2(x)$ pertencem a classes de equivalência cíclica distintas.

▷ Para $i \neq l$, com $l > i$, após algums simplificações em (4.8), obtém-se

$$x^t\,[1 + s_i(x)m_i(x)] - [1 + s_l(x)m_l'(x)] \prod_{j=i}^{l-1} s_j(x) = 0 \bmod \; s_l(x) \prod_{j=i}^{l-1} s_j(x). \tag{4.10}$$

Para (4.10) ser satisfeita, $\prod_{j=i}^{l-1} s_j(x)$ deve dividir $x^t\,[1 + s_i(x)m_i(x)]$. Como

$$\gcd\left[x^t, \prod_{j=i}^{l-1} s_j(x)\right] = 1,$$

e uma vez que $\gcd[1 + s_i(x)m_i(x), s_i(x)] = 1$, conclui-se que $1 + s_i(x)m_i(x)$ não é divisível por $\prod_{j=i}^{l-1} s_j(x)$, pois este último tem $s_i(x)$ como fator. Assim, a hipótese que $c_1(x)$ e $c_2(x)$ pertencem à mesma classe de equivalência constacíclica é falsa para $i \neq l$ e, portanto, $c_1(x)$ e $c_2(x)$ pertencem a classes de equivalência constacíclica distintas.

Na segunda parte da prova, considere $c(x)$ em (4.7) para $i = k/2$. Sendo $m_{k/2}(x) = 1$, obtém-se

$$c(x) = g(x) \prod_{j=1}^{k/2-1} s_j(x) + g(x) \prod_{j=1}^{k/2} s_j(x) \mod x^n - a.$$

Entretanto, $h(x) = \prod_{j=1}^{k/2} s_j(x)$ e $g(x)h(x) = 0 \mod x^{p^2-1} - 1$. Logo,

$$c(x) = g(x) \prod_{j=1}^{k/2-1} s_j(x)$$

para $i = k/2$. Observe que o resultado é independente do polinômio $m_{k/2}(x)$, assim, por definição, $m_{k/2}(x) = 1$. De modo direto, pode-se verificar que a classe de equivalência cíclica gerada por $c(x) = g(x) \prod_{j=1}^{k/2-1} s_j(x)$ não foi gerada previamente por $c(x)$ em (4.7) para $1 \leq i \leq k/2 - 1$. A classe de equivalência cíclica gerada por $g(x) \prod_{j=1}^{k/2-1} s_j(x)$ é precisamente a sequência-m constacíclica gerada pela divisão do polinômio $x^{p^2-1} - a$ pelo polinômio $s_{k/2}(x)$.

Na terceira e última parte da demonstração, considere o número total de classes de equivalência cíclica geradas. Para $1 \leq i \leq k/2-1$ em (4.7), o grau de $m_i(x)$ é menor ou igual a $k-1-2i$. Logo, existem p^{k-2i} possíveis escolhas para $m_i(x)$. Consequentemente, o número de classes de equivalência cíclica neste caso é dado por $\sum_{i=1}^{k/2-1} p^{k-2i}$. Para $i = k/2$ em (4.7), uma única classe de equivalência cíclica é gerada. Desta forma, o número total de classes de equivalência cíclica distintas é dado por $\sum_{i=1}^{k/2-1} p^{k-2i} + 1 = \sum_{i=1}^{k/2} p^{k-2i}$. Mas, $\sum_{i=1}^{k/2} p^{k-2i}$ é a soma de $k/2$ termos de uma série geométrica finita de razão p^{-2}. Assim, $\sum_{i=1}^{k/2} p^{k-2i} = \frac{p^k-1}{p^2-1} = (p^k-1)/(p^2-1)$. ■

Exemplo 4.3

De acordo com a Tabela 3.1, $x^6 - 3 = M_1(x)M_9(x)M_{13}(x)$. *Os polinômios* $M_1(x)$ *e* $M_{13}(x)$ *pertencem ao expoente* 24, *enquanto o polinômio* $M_9(x)$ *pertence ao expoente* 8. *Assim, considere o código* $\mathcal{C}$ *gerado por* $g(x) = M_9(x) = 3 + x^2$. *Como* $n = 6$ *e* $n - k = 2$ *(grau de* $g(x)$*), então* $k = 4$. *Uma vez que* $g(x)$ *satisfaz o Teorema 3.1, pode-se aplicar o Teorema 4.2 a* $\mathcal{C}$. *Neste caso* $k/2 = 2$ *e, desta forma,* $h(x) = \prod_{j=1}^{2} s_j(x) = s_1(x)s_2(x)$, *em que* $s_1(x) = M_1(x)$ *e* $s_2(x) = M_{13}(x)$ *por exemplo. Para* $i = 1$,

$$c(x) = g(x)[1 + s_1(x)m_1(x)], \tag{4.11}$$

em que grau$[m_1(x)] \leq 4 - 1 - 2 = 1$. *Assim,* $5^2 = 25$ *classes de equivalência constacíclica são geradas. Para* $i = 2$, $m_2(x) = 1$ *por definição, assim*

$$\begin{aligned} c(x) &= g(x)[1 + s_2(x)]s_1(x) \mod x^6 - 3 \\ &= g(x)s_1(x) + \underbrace{g(x)s_1(x)s_2(x)}_{\equiv 0 \bmod x^6-3} \\ &= g(x)s_1(x) \end{aligned} \tag{4.12}$$

e, assim, uma classe de equivalência constacíclica é obtida.

Portanto, $(5^4 - 1)/(5^2 - 1) = 26$ *classes de equivalência constacíclica são geradas por meio de (4.11) e (4.12).* □

Sabe-se que para um código linear constacíclico p-ário $\mathcal{C}$ com comprimento de bloco $n = p+1$ e cujo polinômio gerador $g(x)$ satisfaz o Teorema 3.1, o número de classes de equivalência constacíclica distintas é igual a $(p^k - 1)/(p^2 - 1)$. Então, todas as classes de equivalência constacíclica são especificadas de maneira única por meio de (4.7). Como $(p^k - 1)/(p^2 - 1)$

é o limitante superior para o número de classes de equivalência constacíclica, neste caso, a construção proposta é ótima neste sentido, pois atinge precisamente o limitante.

Em [33], [53], mostra-se que códigos lineares constacíclicos p-ários, $p \neq 2$, podem ser mapeados com sucesso para binário, desta forma, é possível obter códigos CP binários. O procedimento utilizado em [33], [53] para gerar códigos CP por meio de códigos lineares constacíclicos p-ários seleciona p^{k-2} palavras-código, constaciclicamente distintas e com ordem constacíclica plena. Conforme dito no parágrafo anterior, o procedimento descrito nesta tese seleciona $(p^k - 1)/(p^2 - 1)$ palavras-código. Portanto, aplicando o mapeamento para binário usado em [33], [53] e o procedimento utilizado nesta tese (Teorema 4.2) seleciona-se um número maior de palavras-código em relação ao procedimento utilizado em [33], [53], uma vez que $(p^k - 1)/(p^2 - 1)$ é o limitante superior e, portanto, $(p^k - 1)/(p^2 - 1) > p^{k-2}$ para $k > 2$.

Na sequência, é descrito o procedimento utilizado em [33], [53] para mapear as palavras de um código linear constacíclico p-ário para binário e, consequentemente, como se obter o respectivo código CP utilizando as palavras-código selecionadas pelo Teorema 4.2.

4.3.1 Mapeamento de códigos constacíclicos p-ários para binário

Neste ponto, mostra-se como é feito o mapeamento para binário dos códigos constacíclicos p-ários. O texto apresentado a seguir segue aquele publicado originalmente em [33], [34], [53].

Arranjos Bidimensionais e N-uplas

O objetivo é estabelecer uma correspondência um-a-um entre arranjos bidimensionais e N-uplas. Os arranjos bidimensionais considerados são semelhantes ao arranjo $A_{m\times n}$ mostrado em (4.13), cujos elementos $a(i,j)$, $i = \{0,1,\ldots,m-1\}$ e $j = \{0,1,\ldots,n-1\}$, pertencem a um alfabeto arbitrário.

$$A = \begin{bmatrix} a(0,0) & a(0,1) & \ldots & a(0,n-1) \\ a(1,0) & a(1,1) & \ldots & a(1,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ a(m-1,0) & a(m-1,1) & \ldots & a(m-1,n-1) \end{bmatrix}_{m\times n} . \tag{4.13}$$

Em [18] foi utilizada uma correspondência um-a-um entre arranjos bidimensionais e N-uplas, com $N = mn$, que é garantida pelo *teorema chinês do resto* [30]. Nesta situação, é

necessário que m e n sejam primos entre si, isto é, mdc$(m,n) = 1$. Nesta tese, utiliza-se uma correspondência entre arranjos bidimensionais e N-uplas que foi proposta em [54] e apresentada de uma forma mais simples em [32], [53]. Tal correspondência é distinta da proposta utilizada em [18] e, além disso, não necessita que m e n sejam primos entre si.

De acordo com [32], [53], [54], para m e n números inteiros a relação que estabelece uma correspondência um-a-um entre o arranjo $A_{m\times n}$ em (4.13) e N-uplas da forma $\mathbf{b} = (b_0, b_1, \ldots, b_{mn-1})$, ambos com elementos pertencentes a um mesmo alfabeto, é dada por

$$b_{in+j} = a(i,j),\ 0 \leq i \leq m-1 \text{ e } 0 \leq j \leq n-1. \tag{4.14}$$

Exemplo 4.4

Considere o arranjo $A_{3\times 3}$ a seguir

$$A = \begin{bmatrix} a(0,0) & a(0,1) & a(0,2) \\ a(1,0) & a(1,1) & a(1,2) \\ a(2,0) & a(2,1) & a(2,2) \end{bmatrix}_{3\times 3} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}_{3\times 3}. \tag{4.15}$$

Pela relação dada em (4.14), com $n = 3$, a 9-upla correspondente ao arranjo dado em (4.15) é $\mathbf{b} = (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8) = (1,2,3,4,5,6,7,8,9)$. Veja os cálculos na Tabela 4.2. □

Definição 4.2 – Operador **DB(.)**

O operador $\mathbf{DB}(.)$ *atua sobre um arranjo bidimensional $A_{m\times n}$, produzindo o arranjo $A''_{m\times n}$, da seguinte forma:*

1. *O operador* $\mathbf{DB}(.)$, *inicialmente, desloca ciclicamente todas as colunas do arranjo $A_{m\times n}$ uma posição para a direita produzindo um novo arranjo $A'_{m\times n}$;*
2. *depois, o operador* $\mathbf{DB}(.)$ *desloca ciclicamente uma posição para baixo a coluna mais à esquerda do arranjo $A'_{m\times n}$ produzindo o arranjo $A''_{m\times n}$.* □

Exemplo 4.5

Aplicando o operador $\mathbf{DB}(.)$ *da Definição 4.2 ao arranjo $A_{3\times 3}$ do Exemplo 4.4, obtém-se*

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}_{3\times 3} \xrightarrow{\mathbf{DB}(A)} A'' = \begin{bmatrix} 9 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}_{3\times 3}. \tag{4.16}$$

Tabela 4.2: *Correspondência entre os elementos do arranjo $A_{3\times3}$ e os elementos da 9-upla* **b**.

(i,j)	$in+j$	$b_{in+j}=a(i,j)$
$(0,0)$	0	$b_0=a(0,0)=1$
$(0,1)$	1	$b_1=a(0,1)=2$
$(0,2)$	2	$b_2=a(0,2)=3$
$(1,0)$	3	$b_3=a(1,0)=4$
$(1,1)$	4	$b_4=a(1,1)=5$
$(1,2)$	5	$b_5=a(1,2)=6$
$(2,0)$	6	$b_6=a(2,0)=7$
$(2,1)$	7	$b_7=a(2,1)=8$
$(2,2)$	8	$b_8=a(2,2)=9$

□

A 9-upla $\mathbf{b}''=(9,1,2,3,4,5,6,7,8)$ é obtida aplicando a relação dada em (4.14) ao arranjo A'' do Exemplo 4.5. Nota-se que $\mathbf{b}''$ corresponde a um deslocamento cíclico para direita da 9-upla $\mathbf{b}=(1,2,3,4,5,6,7,8,9)$ do Exemplo 4.4. Em termos do operador $\mathbf{S}^i(.)$, a relação entre as 9-uplas $\mathbf{b}$ e $\mathbf{b}''$ pode ser expressa por intermédio deste operador como $\mathbf{S}(\mathbf{b})=\mathbf{b}''$. Vale lembar que a ordem cíclica, Definição 2.8, em termos do operador $\mathbf{S}^i(.)$ é o menor valor de i para o qual $\mathbf{S}^i(\mathbf{b})=\mathbf{b}$; e os possíveis valores de i são os divisores de N, em que N é o comprimento de $\mathbf{b}$. O teorema a seguir faz uso dos operadores $\mathbf{DB}(.)$ e $\mathbf{S}^i(.)$ para estabelecer um resultado importante relacionando um conjunto de arranjos bidimensionais $m\times n$ e o conjunto de mn-uplas derivado deste conjunto de arranjos por meio da relação dada em (4.14).

Teorema 4.3 – [33]

Considere um conjunto $\mathcal{X}$ formado por arranjos bidimensionais $m\times n$ cujos elementos pertencem a um alfabeto arbitrário. O conjunto $\mathcal{X}$ será fechado em relação à operação realizada por $\mathbf{DB}(.)$ se e somente se o conjunto correspondente de mn-uplas for fechado em relação à operação realizada por $\mathbf{S}^i(.)$. □

Demonstração: Seja $A_{m\times n}$ um arranjo bidimensional pertencente ao conjunto $\mathcal{X}$ e seja $\mathbf{b}$ a mn-upla binária correspondente ao arranjo $A_{m\times n}$ de acordo com a Relação (4.14). Seja $A''_{m\times n}$ um arranjo bidimensional tal que $\mathbf{DB}(A_{m\times n})=A''_{m\times n}$. A relação entre os

elementos dos arranjos $A_{m\times n}$ e $A''_{m\times n}$ é dada por

$$a''(i,j) = a(i \bmod m, j-1 \bmod n), \text{ para } 0 \le i \le m-1,\ 1 \le j \le n-1, \text{ e} \quad (4.17)$$

$$a''(i,0) = a(i-1 \bmod m, n-1), \text{ para } 0 \le i \le m-1 \text{ e } j = 0, \quad (4.18)$$

em que $l \bmod y$ denota o resto da divisão quando l é dividido por y. Sendo $\mathbf{S}(\mathbf{b}) = \mathbf{b}'$, a relação entre os elementos das mn-uplas $\mathbf{b}$ e $\mathbf{b}'$ é tal que

$$b'_{in+j \bmod mn} = b_{in+j-1 \bmod mn}, \text{ para } 0 \le i \le m-1 \text{ e } 0 \le j \le n-1. \quad (4.19)$$

A mn-upla $\mathbf{b}''$ é obtida aplicando-se a Relação (4.14) ao arranjo $A''_{m\times n}$. Logo, usando as Relações (4.17) e (4.18) e para $i = \{0, 1, 2, \ldots, m-1\}$ obtém-se

$$b''_{in+j \bmod mn} = b_{in+j-1 \bmod mn},\ 1 \le j \le n-1, \text{ e} \quad (4.20)$$

$$b''_{in+j \bmod mn} = b_{in-1 \bmod mn},\ j = 0. \quad (4.21)$$

Comparando as Relações (4.20) e (4.21) com a Relação (4.19), conclui-se que $\mathbf{b}' = \mathbf{b}''$ e, assim, $\mathbf{S}(\mathbf{b}) = \mathbf{b}''$. Portanto, uma condição suficiente para o conjunto $\mathcal{X}$ ser fechado com relação à operação realizada por $\mathbf{DB}(.)$ é o conjunto de mn-uplas ser fechado com relação à operação realizada por $\mathbf{S}^i(.)$. De maneira análoga, pode-se mostrar que uma condição necessária para o conjunto $\mathcal{X}$ ser fechado com relação à operação realizada por $\mathbf{DB}(.)$ é o conjunto de mn-uplas ser fechado em relação à operação realizada por $\mathbf{S}^i(.)$. ■

Uma Representação Cíclica para os Elementos de GF(p)

Neste ponto, o objetivo é representar os elementos $\{0, 1, 2, \ldots, p-1\}$ de GF(p) por meio de N-uplas binárias. Sendo p um número primo ímpar, $p-1$ sempre será um número par. Conforme mencionado anteriormente, a ordem cíclica de uma N-upla é igual a N ou a um de seus divisores. Logo, para $N = p-1$, sempre existe uma $(p-1)$-upla binária de ordem cíclica igual a 2 que corresponde a $(p-1)$-upla de peso $w = (p-1)/2$ cujas coordenadas assumem valores alternados 0 ou 1, isto é, $(1, 0, 1, 0, \ldots, 1, 0)$ ou $(0, 1, 0, 1, \ldots, 0, 1)$. Além do mais, sempre existe pelo menos uma $(p-1)$-upla binária de ordem cíclica $p-1$ que corresponde à $(p-1)$-upla de peso unitário. Porém, é possível que existam outras $(p-1)$-uplas com ordem cíclica igual a $p-1$, além da que foi citada, e que não tenham peso igual a 1.

Exemplo 4.6

Para $p = 7$, *as* 6*-uplas binárias* $\mathbf{v}_1 = (1, 0, 0, 0, 0, 0)$, $\mathbf{v}_2 = (1, 1, 1, 0, 0, 0)$ *e* $\mathbf{v}_3 = (1, 1, 0, 1, 0, 0)$

possuem ordem cíclica igual a 6*, enquanto que a* 6*-upla binária* $\mathbf{v}_4 = (1,0,1,0,1,0)$ *possui ordem cíclica igual a* 2*.* □

A definição a seguir estabelece uma representação para os elementos de GF(p) por meio de $(p-1)$-uplas binárias.

Definição 4.3 – Representação-**V**

Seja $\mathbf{v}$ *uma* $(p-1)$*-upla binária cuja ordem cíclica é igual a* $p-1$*. Define-se a* ***representação*-V***, como uma representação para os elementos de* GF(p) *por intermédio de* $(p-1)$*-uplas binárias tal que os elementos não-nulos* a^i*,* $i = 0,1,2,\ldots,p-2$*, são representados pelas* $(p-1)$*-uplas binárias* $\mathbf{S}^i(\mathbf{v})$ *em que* a *é um elemento gerador do grupo multiplicativo de* GF(p) *e* $\mathbf{S}^i(.)$ *é o operador que desloca ciclicamente de* i *posições para a direita a* $(p-1)$*-upla* $\mathbf{v}$*. Além disso, o elemento* 0 *pode ser representado por uma* $(p-1)$*-upla binária não-nula* $\mathbf{v}'$ *e seus deslocamentos cíclicos tal que* $\mathbf{v}' \neq \mathbf{S}^i(\mathbf{v})$ *para* $0 \leq i \leq p-2$*. Em particular,* $\mathbf{v}'$ *pode ser escolhida como a* $(p-1)$*-upla toda nula denotada por* $\mathbf{0}$*.* □

Exemplo 4.7 – Continuação do Exemplo 4.6

Considere $p = 7$*,* $a = 3$*,* $\mathbf{v}' = \mathbf{v}_4 = (1,0,1,0,1,0)$ *e* $\mathbf{v} = \mathbf{v}_3 = (1,1,0,1,0,0)$*. A representação-***V** *resultante para* GF(7) *é dada na Tabela 4.3.* □

A representação-**V** dada na Definição 4.3 nada mais é que um conjunto de $(p-1)$-uplas binárias, logo é possível interpretá-la como um código de bloco cíclico não-linear. Assim, pode-se associar uma distância mínima, denotada por $d(\mathbf{v})$, cuja definição é a mesma dada na Definição 2.5. Baseando-se nesta afirmação, se existir uma representação-**V** em que a distância de Hamming entre qualquer par de $(p-1)$-uplas deste conjunto for igual a $d(\mathbf{v})$, então esta representação é dita ser *equidistante*. Claramente, a representação-**V** dada na Tabela 4.3 não é uma representação equidistante. Se a representação-**V** for limitada aos elementos do grupo multiplicativo de GF(p), e a denotando por representação-$\mathbf{V}^*$, então, por exemplo, para $\mathbf{v}^* = (1,0,0,\ldots,0)$ ou $\mathbf{v}^* = (0,1,1,\ldots,1)$ a representação-$\mathbf{V}^*$ é equidistante com $d(\mathbf{v}^*) = 2$.

Mapeamento para binário

Uma vez que foram definidos a relação entre arranjos bidimensionais e N-uplas e a representação cíclica para os elementos de GF(p), o procedimento para mapear as palavras

Tabela 4.3: *representação-***V** *para o elementos de* GF(7).

a^i	6-upla
0	$(1,0,1,0,1,0)$
0	$(0,1,0,1,0,1)$
3^0	$(1,1,0,1,0,0)$
3^1	$(0,1,1,0,1,0)$
3^2	$(0,0,1,1,0,1)$
3^3	$(1,0,0,1,1,0)$
3^4	$(0,1,0,0,1,1)$
3^5	$(1,0,1,0,0,1)$

de um código linear constacíclico p-ário para binário é descrito a seguir. Primeiro, cada palavra p-ária $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$, pertencente ao código constacíclico, é mapeada em um arranjo bidimensional cujas colunas são as transpostas das $(p-1)$-uplas binárias, dadas pela representação-**V**, para cada coordenada c_i, $0 \leq i \leq n-1$, da palavra-código $\mathbf{c}$. Depois, os arranjos bidimensionais são convertidos em N-uplas binárias por meio de (4.14). O exemplo a seguir ilustra o procedimento descrito.

Exemplo 4.8

Considere as 6*-uplas* 5*-árias* $(4,0,2,0,1,0)$ *e* $(2,1,1,3,3,4)$*, palavras do código constacíclico do Exemplo 3.3, e a representação-***V** *em que* $\mathbf{v} = (1,1,0,0)$ *e* $\mathbf{v}' = (1,0,1,0)$*. Sendo assim, a representação-***V** *para os elementos* $\{0,1,2,3,4\}$ *de* GF(5) *é:* $0 \leftrightarrow (1,0,1,0)$ *ou* $0 \leftrightarrow (0,1,0,1)$*,* $3^0 = 1 \leftrightarrow (1,1,0,0)$*,* $3^1 = 3 \leftrightarrow (0,1,1,0)$*,* $3^2 = 4 \leftrightarrow (0,0,1,1)$ *e* $3^3 = 2 \leftrightarrow (1,0,0,1)$*. Logo, os arranjos bidimensionais* $A_{4\times 6}$ *correspondentes às duas palavras-código dadas são:*

$$(4,0,2,0,1,0) \Leftrightarrow \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}_{4\times 6} \quad \text{e} \quad (2,1,1,3,3,4) \Leftrightarrow \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{4\times 6}.$$

As respectivas 24*-uplas são*

$$(4,0,2,0,1,0) \Leftrightarrow (0,1,1,1,1,1,0,0,0,0,1,0,1,1,0,1,0,1,1,0,1,0,0,0) \text{ e}$$
$$(2,1,1,3,3,4) \Leftrightarrow (1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1). \qquad \square$$

Com o auxílio do Exemplo 4.8, é possível perceber que na representação binária, as palavras de um código constacíclico p-ário podem ser de peso constante ou não. Tal característica

depende da escolha feita para as $(p-1)$-uplas $\mathbf{v}$ e $\mathbf{v}'$ da representação-$\mathbf{V}$. Se $w(\mathbf{v}) = w(\mathbf{v}')$, então a representação binária é de peso constante com $w = nw(\mathbf{v})$, em que n é o comprimento do bloco do código constacíclico. Caso contrário, se $w(\mathbf{v}) \neq w(\mathbf{v}')$, então a representação binária não é de peso constante.

Antes de enunciar o Teorema 4.4, vale ressaltar que na Definição 4.3 para a representação-$\mathbf{V}$, o elemento 0 é representado por uma $(p-1)$-upla $\mathbf{v}'$ e seus deslocamentos cíclicos, logo uma palavra-código p-ária $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ que tenha uma ou mais coordenadas nulas, $c_i = 0$ para $0 \leq i \leq n-1$, pode ser associada a mais de um arranjo bidimensional e, consequentemente, a mais de uma N-upla binária. Sendo assim, deve-se ter um cuidado especial para que as p^k palavras do código constacíclico representem exatamente p^k N-uplas binárias correspondendo as palavras do código cíclico binário. Para isto, as palavras do código constacíclico devem ser separadas em classes de equivalência constacíclica conforme a Definição 3.4. Depois disso, seleciona-se, arbitrariamente, uma palavra-código $\mathbf{c}$ para ser líder de sua respectiva classe de equivalência constacíclica e a ela associa-se um arranjo bidimensional $A_{(p-1)\times n}$. Se $\mathbf{c}$ possui todas as coordenadas não-nulas, o mapeamento de $\mathbf{c}$ para $A_{(p-1)\times n}$ é um-a-um e, portanto, não há problemas. Entretanto, se $\mathbf{c}$ possui uma ou mais coordenadas nulas, o mapeamento de $\mathbf{c}$ para $A_{(p-1)\times n}$ é feito escolhendo-se, inicialmente, uma $(p-1)$-upla $\mathbf{v}'$ arbitrária para representar o elemento 0 e mantendo fixa esta escolha. Os arranjos associados às palavras-código que pertencem à mesma classe de equivalência constacíclica de $\mathbf{c}$, são obtidos aplicando-se o operador $\mathbf{DB}(.)$ ao arranjo $A_{(p-1)\times n}$ de modo que a palavra-código $\mathbf{c}'$, correspondente ao i-ésimo deslocamento constacíclico de $\mathbf{c}$, é representada pelo arranjo bidimensional $Z_{(p-1)\times n}$ obtido ao aplicar o operador $\mathbf{DB}(.)$ i vezes ao arranjo $A_{(p-1)\times n}$. Daqui em diante, faz-se referência a esse processo como *geração biunívoca de arranjos*.

Teorema 4.4 – Códigos cíclicos binários [33]

*Seja p um número primo, $p > 3$, e $\mathcal{C}$ um código constacíclico linear p-ário de parâmetros (n, k, d). Considere que cada palavra-código, $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$, líder de classe de equivalência constacíclica determine um arranjo bidimensional $A_{(p-1)\times n}$ de modo que a i-ésima coluna de $A_{(p-1)\times n}$ seja a transposta de uma $(p-1)$-upla binária que corresponde à representação-$\mathbf{V}$ da i-ésima componente de $\mathbf{c}$ e defina $\mathbf{b}$ como sendo a N-upla, com $N = (p-1)n$, que corresponde ao arranjo $A_{(p-1)\times n}$ por meio de (4.14). Além do mais, considere que as demais palavras-código são mapeadas em N-uplas binárias com o auxílio do processo de **geração biunívoca de arranjos**. Então, o conjunto de p^k N-uplas binárias correspondentes às p^k palavras-código de $\mathcal{C}$ formam um código*

cíclico binário de distância mínima $d_{\min} \geq dd(\mathbf{v})$ *com igualdade se a representação-*$\mathbf{V}$ *de* $\mathrm{GF}(p)$ *for equidistante.* □

Demonstração: Seja $\mathbf{c} \in \mathcal{C}$ uma palavra-código líder de classe de equivalência constacíclica e seja $A_{(p-1)\times n}$ o arranjo bidimensional correspondente a $\mathbf{c}$. Uma vez que $\mathcal{C}$ é um código linear constacíclico, deslocar constaciclicamente para a direita a palavra-código $\mathbf{c}$ produz uma palavra-código $\mathbf{c}' \in \mathcal{C}$ cujo arranjo bidimensional, denotado por $A'_{(p-1)\times n}$, é tal que $\mathbf{DB}(A_{(p-1)\times n}) = A'_{(p-1)\times n}$. Sendo assim, os p^k arranjos bidimensionais, correspondentes às palavras-código de $\mathcal{C}$, formam um conjunto $\mathcal{Y}$ fechado em relação à operação realizada pelo operador $\mathbf{DB}(.)$. Segue do Teorema 4.3 que o conjunto de p^k N-uplas binárias $\mathbf{b}$, com $N = (p-1)n$, obtidas ao se aplicar a relação dada em (4.14) aos arranjos bidimensionais do conjunto $\mathcal{Y}$, é um conjunto fechado em relação à operação realizada pelo operador $\mathbf{S}^i(.)$ e, portanto, é um código cíclico binário.

Para concluir a demonstração, resta deduzir o limitante inferior dado para $d_{\min}$. Como o código $\mathcal{C}$ tem distância mínima d, duas palavras-código distintas $\mathbf{c}_1$ e $\mathbf{c}_2$ diferem em d coordenadas no mínimo, isto é, a distância de Hamming entre elas satisfaz $d(\mathbf{c}_1, \mathbf{c}_2) \geq d$. Sendo assim, as N-uplas binárias $\mathbf{b}_1$ e $\mathbf{b}_2$, correspondendo a $\mathbf{c}_1$ e $\mathbf{c}_2$, respectivamente, diferem em $dd(\mathbf{v})$ coordenadas no mínimo, em que $d(\mathbf{v})$ é a distância mínima da representação-$\mathbf{V}$. Uma vez que $d(\mathbf{c}_1, \mathbf{c}_2) \geq d$ é satisfeita com igualdade para algumas escolhas de $\mathbf{c}_1$ e $\mathbf{c}_2$, conclui-se que $d_{\min} \geq dd(\mathbf{v})$ e ela é satisfeita com igualdade caso a representação-$\mathbf{V}$ seja equidistante. ■

O Teorema 4.4 mostra que é possível obter o código cíclico binário $\mathcal{C}_1$ a partir do código linear constacíclico p-ário $\mathcal{C}_2$. Desta forma, se é possível selecionar as palavras-código de $\mathcal{C}_1$ que possuem ordem cíclica plena e são ciclicamente distintas, então um código CP binário é obtido. Entretanto, admitindo que o polinômio gerador de $\mathcal{C}_2$ satisfaz o Teorema 3.1, é possível selecionar todas as palavras-código de $\mathcal{C}_2$ que são constaciclicamente distintas por meio do Teorema 4.2. Assim, a importância do Teorema 4.4 vem do fato que, mapeando as palavras-código de $\mathcal{C}_2$ para binário, conforme o procedimento descrito anteriormente, tem-se a garantia de que as palavras-código de $\mathcal{C}_2$ em binário possuem ordem cíclica plena e são ciclicamente distintas, desta forma, obtendo-se um código CP binário. Portanto, é possível enunciar a construção a seguir.

Construção 4.1

Seja p um número primo, $p > 3$, $n = p + 1$ e k um número par tal que $4 \leq k \leq p - 1$. Escolha uma representação-$\mathbf{V}$ com $\mathbf{v} = (1, 0, 0, \ldots, 0)$ e $w(\mathbf{v}') \geq 3$, e um código C constacíclico linear p-ário $(p + 1, k, p - k + 2)$ (MDS) cujo polinômio gerador $g(x)$ satisfaz o Teorema 3.1. Mapeando para binário cada palavra-código $c(x)$ selecionada de acordo com o Teorema 4.2, obtém-se um $CCP(N, M_c, d_c)$ com $N = p^2 - 1$, $M_c = (p^k - 1)/(p^2 - 1)$ e com distância mínima cíclica $d_c \geq (p - k + 2)d(\mathbf{v})$. □

A Construção 4.1 é uma contribuição desta tese e é usada no Capítulo 5 para demonstrar a aplicação de códigos CP como sequências de protocolo para o canal de colisão sem realimentação. Assim, define-se $w(\mathbf{v}') \geq 3$ de modo que $d(\mathbf{v}) \geq 2$, pois na aplicação como sequência de protocolo é desejável ter valores elevados de d_c.

Neste ponto, também vale lembrar uma análise feita no Capítulo 3 sobre códigos lineares constacíclicos que são MDS e cujo polinômio gerador satisfaz o Teorema 3.1. Embora o Teorema 3.1 garanta uma maneira de construir códigos constacíclicos cujas palavras-código não-nulas tenham ordem constacíclica plena, não se tem a garantia de que estes códigos também são MDS, já que a escolha do polinômio gerador para garantir que ele seja MDS está relacionada com a escolha das raízes consecutivas de $x^{p+1} - a$. Para construir códigos constacíclicos que sejam MDS e que possuam todas as palavras-código com ordem constacíclica plena, limitam-se as possíveis escolhas para o polinômio gerador desses códigos, visto que duas condições devem ser satisfeitas de modo simultâneo, ou seja, deve-se selecionar raízes de $x^{p+1} - a$ que sejam consecutivas e que possuam ordem multiplicativa diferente de $p^2 - 1$. Para os casos em que $p = 2^m - 1$, número primo de Mersenne, as duas condições são satisfeitas de forma direta e simultânea haja vista os resultados do Lema **??** e do Corolário 3.1. Para outros casos, não existe uma prova, que seja do conhecimento do autor, tal que para um dado p e $x^{p+1} - a$, sempre há um polinômio gerador que satisfaça as duas condições de modo simultâneo. Porém, o código constacíclico do Exemplo 3.4 mostra que tal situação é possível.

Capítulo 5

Aplicação: Sequências de Protocolo para o Canal de Colisão sem Realimentação

Ninguém é tão sábio que nada tenha para aprender, nem tão tolo que nada tenha para ensinar.

— Blaise Pascal

Neste capítulo, o objetivo é mostrar a aplicação de códigos CP como sequências de protocolo para o canal de colisão sem realimentação [18], [22], [23]. O desempenho das sequências de protocolo propostas é analisado e é feita uma comparação com as sequências de protocolo, também construídas por meio de códigos CP, propostas em [18], [21], [36], [53].

5.1 O canal de Colisão sem Realimentação (CCsR)

No CCsR [22], [23], de forma compartilhada, os usuários emitem informação na forma de pacotes cujos valores são elementos de GF(q), e geralmente um valor elevado de q é usado.

Cada pacote possui uma duração fixa de T segundos. Os usuários particionam o tempo dos seus respectivos relógios em intervalos de tempo com duração de T segundos e os seus pacotes devem ser transmitidos alinhados com estes intervalos.

No CCsR, devido à ausência de um elo de realimentação e da defasagem entre os relógios dos usuários, não é possível, por exemplo, compartilhar o canal em um modo de transmissão por divisão de tempo (TDMA). Além do mais, a ausência de um elo de realimentação não permite que os usuários tenham alguma informação sobre as mensagens enviadas em intervalos de tempo anteriores. A saída do CCsR corresponde a uma das três situações possíveis: *silêncio*, *colisão* e *mensagem*. O silêncio indica que, num dado intervalo de tempo, não há emissão de pacotes por parte dos usuários, ou seja, não há nenhum usuário ativo. A colisão indica que mais de um usuário está ativo emitindo pacotes no mesmo intervalo de tempo. E, por último, a mensagem indica que, num dado intervalo de tempo, um único usuário está ativo emitindo pacotes.

No CCsR, cada usuário possui uma sequência de protocolo binária de comprimento N. Para o usuário i, a sequência de protocolo é denotada por $\mathbf{s}_i = \{s_{i_l}\}_{l=1}^{N}$. A sequência de protocolo determina quando é permitido ao usuário i utilizar o canal e ela é independente dos pacotes enviados. A sequência de protocolo controla a emissão de pacotes do usuário i conforme explicado a seguir. No j-ésimo intervalo de tempo, se $s_{i_j} = 1$, $1 \leq j \leq N$, então é permitido que o usuário use o canal. Caso contrário, $s_{i_j} = 0$, o usuário não tem permissão para usar o canal e, desta forma, permanece em silêncio durante o j-ésimo intervalo de tempo. O usuário i continua a usar periodicamente sua sequência de protocolo, $\mathbf{s}_i$, até que não tenha mais pacotes para emitir. Após esse tempo de atividade, o usuário i deve permanecer inativo por, no mínimo, $N - 1$ intervalos de tempo para poder voltar a emitir pacotes. Para o CCsR, um *quadro* corresponde ao período de uma sequência de protocolo que é igual a N intervalos de tempo. Assim, se $\mathbf{s}_i$ tem peso de Hamming w, então o usuário i é capaz de enviar w pacotes por quadro. Sob certas restrições de uso do canal, o usuário i codifica os seus próprios pacotes, assim transmitindo pacotes redundantes, de tal forma que alguns dos seus pacotes perdidos em colisões possam, em condições específicas, ser recuperados no receptor.

5.1.1 Um caso particular

Em [18] demonstra-se que códigos CP constituem uma solução natural para o caso particular de acesso múltiplo no CCsR em que M usuários, de um total de U, estão ativos em um

dado quadro. Nesta situação, cada usuário recebe uma palavra do código CP e a utiliza como sequência de protocolo para controlar suas transmissões. Desta forma, as palavras do código CP constituem um conjunto (U, M, N, σ) de sequências de protocolo, em que U denota o total de usuários que compartilham o canal, M denota o número de usuários ativos por quadro, cujo comprimento é denotado por N, e σ denota o número mínimo de pacotes por quadro que podem ser recebidos livres de colisão. A taxa total de informação máxima obtida nesta situação é dada por [18]

$$R_{\text{sum}} = \frac{M\sigma}{N} \text{ (pacotes/intervalo de tempo).} \tag{5.1}$$

5.2 Sequências de Protocolo

Em [18], as sequências de protocolo propostas são obtidas, exclusivamente, por meio de códigos CP de peso constante, ou seja, todas a palavras do código CP possuem o mesmo peso de Hamming. Uma abordagem complementar ao trabalho apresentado em [18] utiliza códigos ciclicamente permutáveis de peso não-constante [33], [53] para obter sequências de protocolo. Códigos CP de peso não-constante permitem que usuários distintos usem sequências de protocolo com diferentes *fatores de trabalho*. O fator de trabalho λ_i do usuário i, $1 \leq i \leq U$, é definido como a fração de tempo em que a sua sequência de protocolo assume o valor 1 [23]. Então, para sequências de protocolo provenientes das palavras de um código CP de comprimento N, pode-se, alternativamente, definir o fator de trabalho λ_i, para o usuário i, como a razão entre o peso w_i da palavra-código, correspondente à sequência de protocolo, e o comprimento N das palavras-código, logo $\lambda_i = w_i/N$. Se o código CP for de peso constante, então todos os usuários possuem o mesmo fator de trabalho dado por $\lambda = w/N$.

O Teorema 4 em [18] estabelece condições para códigos CP de peso constante serem usados como sequências de protocolo para o CCsR. Além do mais, permite calcular os valores de M e σ neste caso. Entretanto, para o caso de códigos CP de peso não-constante, que não é abordada em [18], é necessário estabelecer um novo resultado. Com esse intuito, o Teorema 5.1 [36], enunciado na sequência, estabelece um resultado para o cálculo de M e σ quando as palavras de um código CP, de peso constante ou não, são usadas como sequências de protocolo. Se o código CP for de peso constante, então o Teorema 5.1 resulta equivalente ao Teorema 4 em [18].

Definição 5.1

*A **correlação** entre duas N-uplas binárias é definida como o número de coordenadas em que ambas possuem o valor* 1. □

Lema 5.1 – [36]

Em um código ciclicamente permutável de peso não-constante, CCP(N, M_c, d_c), a correlação, denotada por ρ, entre qualquer palavra-código e seus deslocamentos cíclicos ou entre quaisquer deslocamentos cíclicos de duas palavras-código distintas satisfaz $\rho \leq w_{\max} - d_c/2$, em que $w_{\max}$ denota o maior peso de Hamming dentre as palavras do código. □

Demonstração: Para duas N-uplas binárias quaisquer, cuja distância de Hamming seja d, e que possuam pesos de Hamming w_i e w_j, respectivamente, o número de 1's em que elas coincidem é exatamente $(w_i + w_j)/2 - d/2$. Sendo as N-uplas binárias palavras de um código CP, sendo $w_{\max}$ o maior peso de Hamming dentre as palavras-código e sendo d_c a distância mínima cíclica, então o valor máximo para a correlação entre duas palavras do código é obtido quando ambas possuem peso $w_{\max}$. Assim, $\rho = w_{\max} - d_c/2$. Portanto, a correlação entre quaisquer deslocamentos cíclicos de duas palavras-código distintas satisfaz $\rho \leq w_{\max} - d_c/2$. ■

Teorema 5.1 – [36]

Seja CCP$(N, M_c = U, d_c)$ um código CP de peso não-constante, de valor mínimo $w_{\min}$ e valor máximo $w_{\max}$. Para um número inteiro σ, $1 \leq \sigma \leq w_{\max}$, um CCP$(N, M_c = U, d_c)$ é um conjunto de sequências de protocolo, representadas por (U, M, N, σ), para

$$M \geq \min\left\{U, \left\lfloor \frac{w_{\min} - 1}{w_{\max} - d_c/2} \right\rfloor, \left\lfloor \frac{w_{\min} - \sigma}{w_{\max} - d_c/2} \right\rfloor + 1\right\}, \tag{5.2}$$

em que $\lfloor x \rfloor$ é o maior número inteiro positivo tal que $\lfloor x \rfloor \leq x$. □

Demonstração: Inicialmente, considere a estratégia pela qual o receptor é capaz de identificar os usuários cujos pacotes foram recebidos com sucesso. Para isto, considere o conjunto $\mathcal{W}$ cujos elementos são os pesos de Hamming das sequências de protocolo dos U usuários do canal e considere um quadro arbitrário de comprimento N que é processado pelo receptor em um instante de tempo também arbitrário. Seja $\tau = [\tau_1, \tau_2, \ldots, \tau_N]$ a N-upla binária que representa o *vetor atividade de transmissão*, em que τ_j, $1 \leq j \leq N$, assume valores 0 ou 1 e é recebido no j-ésimo intervalo de tempo desse quadro. Se $\tau_j = 0$,

houve "silêncio" no j-ésimo intervalo de tempo, caso contrário, se $\tau_j = 1$, houve uma "mensagem" ou uma "colisão". O receptor decide se o usuário i está ativo no quadro recebido se e somente se os valores de j para os quais $\tau_j = 1$ coincidem com os valores de l para os quais $s_{i_l} = 1,\ 1 \leq l \leq N$, em que $\mathbf{s}_i = \{s_{i_l}\}_{l=1}^{N}$, $1 \leq i \leq U$, denota a sequência de protocolo do usuário i. Se o usuário i, de fato, estiver ativo no quadro, então a regra de decisão descrita estará sempre correta. No entanto, se o usuário i não estiver ativo no quadro, então a regra de decisão utilizada falhará. Para deduzir uma condição suficiente assegurando que o usuário i não está ativo em um quadro arbitrário, considere um número M de usuários ativos cujas sequências de protocolo possuem peso arbitrário, não necessariamente iguais, mas que pertencem a $\mathcal{W}$, e considere, ainda, que a sequência de protocolo do usuário i tem peso igual a $w_i \in \mathcal{W}$. Portanto, se o usuário i não estiver ativo no quadro, quando no máximo M usuários estão ativos, e ρ denota o número máximo de 1's em que as sequências de protocolo dos M usuários coincidem, uma por vez, com os 1's em $\mathbf{s}_i$, então $M\rho < w_{\min}$ é uma condição suficiente para identificar corretamente que o usuário i não está ativo, qualquer que seja o $w_i \in \mathcal{W}$. Porém, do Lema 5.1 tem-se $\rho \leq w_{\max} - d_c/2$, porque a sequência de protocolo de cada usuário pode estar deslocada ciclicamente. Assim resulta que $M(w_{\max} - d_c/2) < w_{\min}$ ou, equivalentemente, $M(w_{\max} - d_c/2) \leq w_{\min} - 1$ é uma condição suficiente para identificar corretamente os usuários ativos por quadro. Nessa condição, o número de usuários ativos M é dado por

$$M = \left\lfloor \frac{w_{\min} - 1}{w_{\max} - d_c/2} \right\rfloor. \tag{5.3}$$

No próximo passo é estabelecida uma condição suficiente para que cada um dos M usuários ativos, por quadro, possa enviar no mínimo σ pacotes, que são recebidos livres de colisão. Para isto, suponha que o usuário i está ativo. Como os pacotes dos demais $M-1$ usuários ativos podem colidir com, no máximo, $w_{\max} - d_c/2$ pacotes enviados pelo usuário i, o usuário i tem a garantia de que $w_{\min} - (M-1)(w_{\max} - d_c/2)$ dos seus pacotes chegam ao receptor sem sofrer colisão, qualquer que seja o peso $w_i \in \mathcal{W}$ da sequência de protocolo do usuário i. Logo, $\sigma \geq w_{\min} - (M-1)(w_{\max} - d_c/2)$ ou, equivalentemente,

$$M = \left\lfloor \frac{w_{\min} - \sigma}{w_{\max} - d_c/2} \right\rfloor + 1. \tag{5.4}$$

Por fim, é trivial que a condição $M \leq U$ seja satisfeita e, portanto, se o valor de M é o mínimo entre U e os valores inteiros dados em (5.3) e (5.4), então o receptor é capaz de identificar corretamente os usuários ativos por quadro e cada um deles tem a garantia de poder enviar σ pacotes que são recebidos livres de colisão. Porém, as expressões (5.3)

e (5.4) são deduzidas considerando o pior caso, pois é possível situações em que só há usuários ativos que possuem sequências de protocolo com peso $w_{\min}$ e, então, o número de usuários ativos é maior que o valor calculado em (5.3) e (5.4). Logo, justifica-se a desigualdade em (5.2) e a condição de igualdade ocorre quando o código CP é de peso constante. ■

5.2.1 Sequências-BCH e Sequências-RS

Segundo [21], os parâmetros (U, M, N, σ) das sequências-BCH são $U = p^{(k-2)r}, N = p(p^r - 1)$ e M em função de σ é dado por

$$M = \min\left\{U, \left\lfloor \frac{w-1}{w-d_c/2} \right\rfloor, \left\lfloor \frac{w-\sigma}{w-d_c/2} \right\rfloor + 1 \right\}, \tag{5.5}$$

em que p é um número primo tal que $p \geq 5$, e r e k são números inteiros tais que $r \geq 1$ e $3 \leq k \leq p-1$. As sequências de protocolo são palavras de um código CP de peso constante com $w = p^r - 1$ e $d_c \geq 2(p^r - 1 - (k-1)p^{r-1})$. É demonstrado em [21] que, para $r = 1$, as Sequências-BCH equivalem às Sequências-RS [18], considerando os códigos Reed-Solomon com comprimento do bloco máximo igual a $p-1$.

Segundo [55], o limitante superior para o valor de M é deduzido considerando o número máximo de usuários ativos que podem transmitir, no mínimo, um pacote que seja recebido livre de colisão em um quadro de comprimento N. Assim, para $\sigma = 1$, o segundo termo do lado direito em (5.5) é o menor. Logo, substituindo w por $p^r - 1$ e d_c por $2(p^r - 1 - (k-1)p^{r-1})$ resulta $M \geq \left\lfloor \frac{(p^r-1)-1}{(p^r-1)-[p^r-1-(k-1)p^{r-1}]} \right\rfloor = \left\lfloor \frac{p^r-2}{(k-1)p^{r-1}} \right\rfloor$. Para valores elevados de p, o valor de M é aproximadamente igual a $\lfloor p/(k-1) \rfloor$. Se este resultado for utilizado para avaliar o número de usuários ativos, então, no máximo, $\lfloor p/2 \rfloor = (p-1)/2$ podem estar ativos, uma vez que $3 \leq k \leq p-1$.

Em [21], para deduzir o limitante inferior para o valor de R_{sum}, o primeiro passo é avaliar para quais valores de σ, no lado direito de (5.5), o terceiro termo é o menor. Para isto, o segundo e o terceiro termos do lado direito de (5.5) podem ser reescritos, respectivamente, como $m_1 \leq \frac{w-1}{w-d_c/2}$ e $m_2 \leq \frac{w-\sigma}{w-d_c/2} + 1 = \frac{2w-\sigma-d_c/2}{w-d_c/2}$. Para que ocorra $m_2 \leq m_1$, é suficiente que $(2w - \sigma - d_c/2) \leq (w-1)$, o que implica em $\sigma \geq w + 1 - d_c/2$. Logo, para $\sigma \geq (k-1)p^{r-1} + 1$, o terceiro termo é o menor. Portanto, $M \geq \left\lfloor \frac{w-\sigma}{(k-1)p^{r-1}} \right\rfloor + 1 > \frac{w-\sigma}{(k-1)p^{r-1}}$, e quando substituído em (5.1) resulta em $R_{\text{sum}} \geq \frac{\sigma(w-\sigma)}{N(k-1)p^{r-1}}$, cujo valor é máximo para $\sigma = w/2$, desde que $(w/2) \geq (k-1)p^{r-1} + 1$. Logo, sendo $N = p(p^r - 1) = pw$, resulta

$R_{\text{sum}} \geq \frac{(w/2)(w-w/2)}{pw(k-1)p^{r-1}} = \frac{p^r-1}{4(k-1)p^r}$, que para valores elevados de p pode ser aproximada para $\frac{1}{4(k-1)}$.

5.2.2 Sequências-Constacíclicas

As Sequências-Constacíclicas podem ser obtidas por meio de códigos CP de peso constante ou não. A seguir, códigos CP de peso não-constante são usados para gerar as Sequências-Constacíclicas tipo-I e as Sequências-Constacíclicas baseadas nos códigos CP da Construção 4.1. As Sequências-Constacíclicas tipo-II são obtidas por meio de códigos CP de peso constante.

Sequências-Constacíclicas tipo-I

Segundo [33], [53], os parâmetros (U, M, N, σ) das sequências baseadas em códigos CP de peso não-constante são $U = p^{k-2}, N = p^2 - 1$ e M em função de σ é dado por

$$M \geq \min\left\{U, \left\lfloor \frac{w_{\min} - 1}{w_{\max} - d_c/2} \right\rfloor, \left\lfloor \frac{w_{\min} - \sigma}{w_{\max} - d_c/2} \right\rfloor + 1 \right\}, \tag{5.6}$$

em que p é um número primo tal que $p \geq 5$ e k é um número inteiro par tal que $4 \leq k \leq p-1$. As sequências de protocolo são palavras de um código CP, de peso não-constante, com $w_{\min} = p + 1$, $w_{\max} = (p - k + 2) + (k - 1)w(\mathbf{v}')$ e $d_c \geq (p - k + 2)d(\mathbf{v})$, em que $w(\mathbf{v}')$, $w(\mathbf{v}') \geq 3$, denota o peso da $(p-1)$-upla que representa o elemento 0 na representação-$\mathbf{V}$ e $d(\mathbf{v})$ denota sua distância mínima.

Para obter-se o limitante superior para M, segue-se o mesmo procedimento aplicado às Sequências-BCH com a hipótese adicional de que todos os usuários ativos, num determinado quadro, possuam sequências de protocolo que correspondem a palavras do código CP com peso igual a $w_{\min}$. Tal hipótese corresponde a substituir $w_{\max}$ por $w_{\min}$ em (5.6) que, nesse caso, é satisfeita com igualdade. Além do mais, para palavras do código CP com peso igual a $w_{\min}$, $d(\mathbf{v}) = 2$. Assim, para $\sigma = 1$, $w_{\max} = w_{\min}$ e $d(\mathbf{v}) = 2$, obtém-se $M \geq \left\lfloor \frac{(p+1)-1}{(p+1)-(p-k+2)} \right\rfloor = \left\lfloor \frac{p}{(k-1)} \right\rfloor$. Logo, se este resultado for utilizado para avaliar o número de usuários ativos, então, no máximo, $\lfloor p/3 \rfloor$ podem estar ativos, uma vez que $4 \leq k \leq p-1$.

Na dedução do limitante inferior para R_{sum}, o primeiro passo é avaliar para quais valores de σ, no lado direito em (5.6), o terceiro termo é o menor. Seguindo o mesmo procedimento utilizado para as Sequências-BCH, obtém-se $\sigma \geq w_{\max} + 1 - d_c/2$. Como $d(\mathbf{v}) \geq 2$ para $w(\mathbf{v}') \geq 3$, resulta $\sigma \geq (k-1)w(\mathbf{v}')+1$ e o terceiro termo é o menor. Assim $M \geq \left\lfloor \frac{w_{\min}-\sigma}{(k-1)w(\mathbf{v}')} \right\rfloor + 1 > \frac{w_{\min}-\sigma}{(k-1)w(\mathbf{v}')}$, que quando substituído em (5.1) resulta em $R_{\text{sum}} \geq \frac{\sigma(w_{\min}-\sigma)}{N(k-1)w(\mathbf{v}')}$, cujo valor é

máximo para $\sigma = w_{\min}/2$, desde que $(w_{\min}/2) \geq (k-1)w(\mathbf{v}')+1$. Logo, sendo $N = p^2-1 = (p+1)(p-1) = w_{\min}(p-1)$, resulta $R_{\text{sum}} \geq \frac{(w_{\min}/2)(w_{\min}-w_{\min}/2)}{w_{\min}(p-1)(k-1)w(\mathbf{v}')} = \frac{(p+1)}{4(p-1)(k-1)w(\mathbf{v}')}$, que para valores elevados de p pode ser aproximada para $\frac{1}{4(k-1)w(\mathbf{v}')}$.

Sequências-Constacíclicas tipo-II

Em [33], [53], os parâmetros (U, M, N, σ) das sequências baseadas em códigos CP de peso constante são $U = A_{p+1}/N, N = p^2-1$ e M em função de σ é dado por

$$M = \min\left\{U, \left\lfloor \frac{w-1}{w-d_c/2} \right\rfloor, \left\lfloor \frac{w-\sigma}{w-d_c/2} \right\rfloor + 1\right\}, \tag{5.7}$$

em que p é um número primo tal que $p \geq 5$, k é um número inteiro par tal que $4 \leq k \leq p-1$ e o valor de A_{p+1} é dado em (2.19). As sequências de protocolo são palavras de um código CP de peso constante com $w = p+1$ e $d_c = 2(p-k+2)$. O limitante superior para o valor de M é o mesmo deduzido para as Sequências-Constacíclicas tipo-I, pois a hipótese assumida, naquele ponto, de que todos os usuários ativos possuem sequências de protocolo que são as palavras do código CP com peso igual a $w_{\min}$, corresponde às Sequências-Constacíclicas tipo-II. Logo, $M \leq \lfloor p/3 \rfloor$.

A dedução do limitante inferior para o valor de R_{sum} segue o procedimento já apresentado anteriormente para as outras sequências. Dessa forma, $\sigma \geq w+1-d_c/2$. Logo, para $\sigma \geq k$, o terceiro termo, do lado direito de (5.7) é o menor. Assim resulta $M \geq \left\lfloor \frac{w-\sigma}{(k-1)} \right\rfloor + 1 > \frac{w-\sigma}{k-1}$, que quando substituído em (5.1), resulta em $R_{\text{sum}} \geq \frac{\sigma(w-\sigma)}{N(k-1)}$, cujo valor é máximo para $\sigma = w/2$, desde que $(w/2) \geq k$. Logo, sendo $N = p^2-1 = (p+1)(p-1) = w(p-1)$, resulta $R_{\text{sum}} \geq \frac{(w/2)(w-w/2)}{w(p-1)(k-1)} = \frac{(p+1)}{4(p-1)(k-1)}$, que para valores elevados de p pode ser aproximada para $\frac{1}{4(k-1)}$.

Sequências-Constacíclicas baseadas na Construção 4.1

De acordo com a Construção 4.1, os parâmetros (U, M, N, σ) das sequências baseadas em códigos CP de peso não-constante são $U = (p^k-1)/(p^2-1), N = p^2-1$ e M em função de σ é dado por

$$M \geq \min\left\{U, \left\lfloor \frac{w_{\min}-1}{w_{\max}-d_c/2} \right\rfloor, \left\lfloor \frac{w_{\min}-\sigma}{w_{\max}-d_c/2} \right\rfloor + 1\right\}, \tag{5.8}$$

em que p é um número primo tal que $p \geq 5$ e k é um número inteiro par tal que $4 \leq k \leq p-1$. As sequências de protocolo são palavras de um código CP, de peso não-constante, com $w_{\min} = p+1$, $w_{\max} = (p-k+2)+(k-1)w(\mathbf{v}')$ e $d_c \geq (p-k+2)d(\mathbf{v})$, em que $w(\mathbf{v}')$,

Tabela 5.1: *Parâmetros de comparação para as sequências de protocolo. Sequências-Constacíclicas com* $p \geq 5$, $4 \leq k \leq p-1$ *e* $w(\mathbf{v}') \geq 3$. *Sequências-RS e Sequências-BCH com* $p \geq 5$, $3 \leq k \leq p-1$ *e* $r > 1$.

Critérios	**Sequências**				
	Construção 4.1	tipo-I [33]	tipo-II [33]	RS [18]	BCH [21]
Limitante inferior para R_{sum}	$\frac{1}{4(k-1)w(\mathbf{v}')}$	$\frac{1}{4(k-1)w(\mathbf{v}')}$	$\frac{1}{4(k-1)}$	$\frac{1}{4(k-1)}$	$\frac{1}{4(k-1)}$
Limitante superior para M	$\lfloor p/3 \rfloor$	$\lfloor p/3 \rfloor$	$\lfloor p/3 \rfloor$	$\lfloor p/2 \rfloor$	$\lfloor p/2 \rfloor$
Nº de sequências geradas (U)	$\frac{p^k-1}{p^2-1}$	p^{k-2}	$\frac{A_{p+1}}{N}$	p^{k-2}	$p^{(k-2)r}$
Comprimento do quadro (N)	p^2-1	p^2-1	p^2-1	p^2-p	$p(p^r-1)$
Diferentes fatores de trabalho	*sim*	*sim*	*não*	*não*	*não*

$w(\mathbf{v}') \geq 3$, denota o peso da $(p-1)$-upla que representa o elemento 0 na representação-**V** e $d(\mathbf{v})$ denota sua distância mínima.

Observe que as Sequências-constacíclicas baseadas na Construção 4.1 e as Sequências-constacíclicas do tipo-I diferem, apenas, com relação ao parâmetro U. Assim, o limitante inferior para R_{sum} e o limitante superior para o valor de M são os mesmos.

5.3 Comparação das Sequências de Protocolo

De acordo com [55], a avaliação de sequências de protocolo não é simples e o resultado depende, em geral, da natureza da aplicação pretendida. No entanto, os seguintes parâmetros são comumente considerados.

a. O número de usuários, M, ativos por quadro;

b. A taxa total de informação transmitida (R_{sum});

c. O número máximo de sequências distintas;

d. O comprimento do quadro, N, utilizado pelos usuários;

e. Suporte a usuários com diferentes fatores de trabalho;

f. Uso de cabeçalhos de identificação dos usuários.

5.3.1 Análise dos Parâmetros das Sequências

A Tabela 5.1 resume os parâmetros para comparação das sequências apresentadas e que são discutidos a seguir. Como todas as sequências de protocolo analisadas nesta tese são obtidas por meio de códigos CP, isto implica que quando os pacotes chegam ao receptor num dado quadro de transmissão, é possível distinguir os usuários ativos sem a necessidade de cabeçalhos de identificação [55]. A sequência de protocolo de cada usuário pode ser identificada,

mesmo que seja recebida com algum deslocamento cíclico, uma vez que a sequência resultante de um deslocamento cíclico é diferente dela mesma e da sequência de protocolo de cada um dos outros usuários. É também desejável que as sequências de protocolo deem suporte a usuários com diferentes fatores de trabalho [55], pois diferentes sensores ou estações de trabalho, podem necessitar de usuários com diferentes taxas de transmissão. Dentre as sequências apresentadas, as sequências baseadas na Construção 4.1 e as Sequências-Constacíclicas tipo-I possuem tal característica, pois o código CP utilizado não é de peso constante.

O comprimento N do quadro utilizado nas transmissões também é um parâmetro importante porque quanto maior o comprimento do quadro, maior a complexidade de decodificação por intervalo de tempo [21]. As Sequências-Constacíclicas possuem comprimento $N = p^2 - 1$ que é aproximadamente o mesmo valor do comprimento das Sequências-RS, $N = p^2 - p$. Comparando com as Sequencias-BCH, cujo comprimento é $N = p(p^r - 1)$, as Sequências-Constacíclicas possuem comprimento bem inferior, principalmente, à medida que o valor de r aumenta. Por exemplo, para $p = 13$, $k = 4$ e $r = 2$, as Sequências-BCH têm $N = 2184$, enquanto que as Sequências-Constacíclicas e as Sequências-RS, para os mesmos valores de p e k, possuem $N = 168$ e $N = 156$, respectivamente.

Comparando o valor de U das Sequências-Constacíclicas tipo-I com o valor de U das Sequências-RS e Sequências-BCH, conclui-se que ele é igual ao valor da primeira e inferior ao valor da segunda, $U = p^{(k-2)r}$, principalmente para valores elevados de r. As sequências baseadas na Construção 4.1, por sua vez, possuem um valor de U maior do que Sequências-Constacíclicas tipo-I e as Sequências-RS, pois $(p^k - 1)/(p^2 - 1) > p^{k-2}$ para $k > 2$. Por fim, o valor de U das Sequências-Constacíclicas tipo-II, é sempre inferior quando comparado com os valores de U das demais sequências na Tabela 5.1.

Sequências-Constacíclicas tipo-I, tipo-II e as sequências baseadas na Construção 4.1 possuem o mesmo limitante, $M \leq \lfloor p/3 \rfloor$, que é menor que o limitante superior para as Sequências-RS e Sequências-BCH dado por $M \leq \lfloor p/2 \rfloor$. Porém, a diferença entre os valores dos limitantes é cada vez menor à medida que o valor de p aumenta.

Pela Tabela 5.1, as Sequências-Constacíclicas tipo-II possuem o mesmo limitante inferior das Sequências-RS e das Sequências-BCH para R_{sum}. Já as Sequências-Constacíclicas tipo-I e as sequências baseadas na Construção 4.1 possuem um limitante inferior que é menor que o correspondente das demais sequências por um fator de $\frac{1}{w(\mathbf{v}')}$, $w(\mathbf{v}') \geq 3$. Esta diminuição é devida ao fato dos códigos CP usados serem de peso não-constante e o valor de $w(\mathbf{v}')$ influ-

enciar diretamente no peso das palavras-código. Como há usuários com variados fatores de trabalho, o número de usuários ativos por quadro pode diminuir. Porém, como mencionado anteriormente, as Sequências-Constacíclicas tipo-I e as sequências baseadas na Construção 4.1 são as únicas na literatura, obtidas por meio de códigos CP, que comportam usuários com diferentes fatores de trabalho.

Capítulo 6

Considerações Finais e Contribuições

Não me envergonho de mudar de opinião, porque não tenho vergonha de pensar.

— Blaise Pascal

Neste capítulo são apresentados alguns tópicos para futuras investigações. Além do mais, são apresentados os trabalhos publicados em anais de conferências e em revistas especializadas relacionados à pesquisa desta tese, assim como trabalhos publicados em atividades paralelas de pesquisa na área de Comunicações, especificamente em códigos corretores de erro.

6.1 Sugestões para Trabalhos Futuros

A seguir, apresenta-se algumas sugestões para trabalhos futuros que possam ser realizados a partir dos resultados expostos nesta tese.

- Para códigos lineares cíclicos q-ários cujo comprimento de bloco n é um divisor de $q^m - 1$ e cujo polinômio gerador satisfaz o Teorema 2.8, ainda não há uma expressão, como aquela

enunciada no Teorema 4.1, que permita particionar o dicionário de palavras do código em classes de equivalência cíclica;

- ▷ O Teorema 2.8 permite gerar um código linear cíclico q-ário com a garantia de que todas as palavras-código não nulas possuem ordem cíclica plena. Entretanto, dado um código linear cíclico q-ário de comprimento de bloco n que não satisfaça o Teorema 2.8, há palavras-código com ordem cíclica n e outras palavras-código com ordem cíclica igual a um divisor de n. Portanto, encontrar uma expressão similar a do Teorema 4.1, que permita particionar o conjunto de palavras de um código linear cíclico q-ário em classes de equivalência cíclica de ordens distintas, ainda é um problema a ser investigado. Análise similar é válida para o caso dos códigos lineares constacíclicos p-ários;
- ▷ Códigos CP possuem diversas aplicações [23], [25]–[27]. Portanto, a depender da aplicação, investigar códigos CP obtidos por meio das construções propostas nesta tese, que atendam às propriedades solicitadas, por exemplo, peso das palavras-código, distância mínima, função de autocorrelação, etc;
- ▷ Analisar as construções de códigos CP apresentadas nesta tese do ponto de vista da teoria de designs [56].

6.2 Publicações

i. Publicações relacionadas ao trabalho de pesquisa do doutorado:

a) J. S. de Lemos-Neto e V. C. da Rocha , "Códigos ciclicamente permutáveis derivados de códigos constacíclicos", *Anais do XXIX Simpósio Brasileiro de Telecomunicações (XXIX SBrT)*, Curitiba-PR, Brasil, pp. 1–5, Outubro 2011.

b) V. C. da Rocha and J. S. de Lemos-Neto, "New cyclically permutable codes", *IEEE Information Theory Workshop (ITW)*, Rio de Janeiro, Brazil, pp. 693–697, October 2011.

c) J. S. de Lemos-Neto e V. C. da Rocha , "Sequências de protocolo para o canal de colisão sem realimentação", *Anais do XXXI Simpósio Brasileiro de Telecomunicações (XXXI SBrT)*, Fortaleza-CE, Brasil, pp. 1–5, Setembro 2013.

d) J. S. de Lemos-Neto e V. C. da Rocha , "Cyclically permutable codes specified by roots of generator polynomial", *Electronics Letters*, v. 50, n. 17., pp. 1202–1204, August 2014.

ii. Publicações em atividades paralelas de pesquisa:

a) P. R. Freitas, V. C. da Rocha Jr. and J. S. de Lemos-Neto, "On the iterative decoding of binary product codes over the binary erasure channel", *Proceedings of the 8th International Symposium on Wireless Communication Systems (ISWCS 2011)*, Aachen, Germany, pp. 126–130, 2011.

b) W. P. S. Guimarães, J. S. de Lemos-Neto and V. C. da Rocha Jr., "A hybrid iterative decoder for LDPC codes", *Proceedings of the 9th International Symposium on Wireless Communication Systems (ISWCS 2012)*, Paris, France, pp. 979–983, 2012.

c) D. P. B. A. Camara, J. S. de Lemos-Neto and V. C. da Rocha Jr., "Multi-instance Based Cryptographic Key Regeneration System", *Anais do XXX Simpósio Brasileiro de Telecomunicações*, Brasília-DF, Brasil, pp. 1–5, Setembro 2012.

d) W. P. S. Guimarães, J. S. de Lemos-Neto and V. C. da Rocha Jr., "Efficient use of a hybrid decoding technique for LDPC code". *EURASIP Journal on Wireless Communications and Networking*, v. 2014, pp. 32–41, 2014.

e) D. P. B. A. Camara, J. S. de Lemos-Neto and V. C. da Rocha Jr., "Multi-instance Based Cryptographic Key Regeneration System", *Journal of Communication and Information Systems*, v. 29, pp. 46–55, 2014.

REFERÊNCIAS

[1] B. P. Lathi, **Modern Digital and Analog Communication Systems**, 3ª ed. Oxford University Press, 1998.

[2] C. E. Shannon, A mathematical theory of communication, *Bell System Technical Journal*, v. 27, n. 3 and 4, pp. 379–423 and 623–656, July and October 1948.

[3] ——, Communication theory of secrecy systems, *Bell System Technical Journal*, v. 28, n. 4, pp. 656–715, October 1949.

[4] R. W. Hamming, Error detecting and error correcting codes, *Bell System Technical Journal*, v. 29, n. 2, pp. 147–160, Abril 1950.

[5] J. L. Massey, Applied Digital Information Theory – part I, Swiss Federal Institute of Technology-Zurich, Notas de Aula, 1980, *Information and Signal Processing Laboratory*.

[6] N. Abramson, **Information Theory and Coding**. New York: McGraw-Hill, 1963.

[7] T. M. Cover & J. A. Thomas, **Elements of Information Theory**, 2ª ed. John Wiley and Sons, 2006.

[8] J. L. Massey, Applied Digital Information Theory – part II, Swiss Federal Institute of Technology-Zurich, Notas de Aula, 1981, *Information and Signal Processing Laboratory*.

[9] S. Lin & D. J. Costello, **Error Control Coding**, 2ª ed. Prentice Hall, 2004.

[10] S. B. Wicker, **Error Control Systems for Digital Communication and Storage**. Prentice Hall, 1995.

[11] F. J. MacWilliams & N. J. A. Sloane, **The Theory of Error-Correcting Codes**. North-Holland, 1977.

[12] E. R. Berlekamp, **Algebraic Coding Theory**. McGraw-Hill, 1968.

[13] R. E. Blahut, **Algebraic Codes for Data Transmission**. Cambridge University Press, 2003.

[14] T. K. Moon, **Error correction coding : mathematical methods and algorithms**. John Wiley and Sons, 2005.

[15] C. Pimentel, **Comunicação Digital**. Brasport, Rio de Janeiro, 2007.

[16] J. G. Proakis & M. Salehi, **Digital Communications**, 5ª ed. McGraw-Hill Science/Engineering/Math, New York, 2007.

[17] E. N. Gilbert, Cyclically permutable error-correcting codes, *IEEE Transactions on Information Theory*, v. IT-9, n. 3, pp. 175–182, July 1963.

[18] N. Q. A, L. Györfi, & J. L. Massey, Constructions of binary constant-weight cyclic codes and cyclically permutable codes, *IEEE Transactions on Information Theory*, v. IT-38, n. 3, pp. 940–948, May 1985.

[19] E. Berlekamp & J. Justesen, Some long cyclic linear binary codes are not so bad, *IEEE Transactions on Information Theory*, v. IT-20, n. 3, pp. 351–356, May 1974.

[20] R. J. McEliece, **Finite fields for computer scientists and engineers**, 0ª ed., ser. The Kluwer International Series in Engineering and Computer Science; 23. Kluwer Academic Publishers, 1987.

[21] L. Györfi & I. Vajda, Constructions of protocol sequences for multiple access collision channel without feedback, *IEEE Transactions on Information Theory*, v. IT-39, n. 5, pp. 1762–1765, September 1993.

[22] J. L. Massey, The capacity of the collision channel without feedback, *Abstracts of Papers, IEEE Int. Symp. Inform. Theory*, pp. 101, June 1982.

[23] J. L. Massey & P. Mathys, The collision channel without feedback, *IEEE Transactions on Information Theory*, v. IT-31, n. 2, pp. 192–204, March 1985.

[24] S. Bitan & T. Etzion, Constructions for optimal constant weight cyclically permutable codes and difference families, *IEEE Transactions on Information Theory*, v. IT-41, n. 1, pp. 77–87, January 1995.

[25] F. Chung, J. Salehi, & V. Wei, Optical orthogonal codes: design, analysis and applications, *IEEE Transactions on Information Theory*, v. IT-35, n. 3, pp. 595–604, May 1989.

[26] S. W. Golomb, **Digital Communication with Space Application**, 2ª ed. Englewood Cliffs, NJ: Prentice-Hall. Los Altos, CA: Peninsula Publishing, 1982, 1964.

[27] S. Sriram & S. Hosur, Cyclically permutable codes for rapid acquisition in DS-CDMA systems with asynchronous base stations, *IEEE Journal on Selected Areas in Communications*, v. 19, n. 1, pp. 83–94, January 2001.

[28] S. Xia & F. Fu, Nonperiodic cyclic equivalence classes of cyclic codes and algebraic constructions of cyclically permutable codes, In: **Proceedings of the 12th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes**, ser. AAECC-12. London, UK: Springer-Verlag, June 1997, pp. 341–352.

[29] M. Kuribayashi & H. Tanaka, How to generate cyclically permutable codes from cyclic codes, *IEEE Transactions on Information Theory*, v. IT-52, n. 10, pp. 4660–4663, October 2006.

[30] D. M. Burton, **Elementary Number Theory**, 7ª ed. McGraw-Hill, 2010.

[31] S. Katzenbeisser & F. A. P. Petitcolas, **Information Hiding Techniques for Steganography and Digital Watermarking**, 2ª ed. Norwood, MA: Artech House, 2000.

[32] V. C. da Rocha Jr. & J. S. de Lemos-Neto, Nonlinear binary codes derived from constacyclic codes, In: **Proceedings of the 7th International Telecommunications Symposium**. Manaus, Brazil: ITS, September 2010.

[33] ——, New cyclically permutable codes, In: **Information Theory Workshop (ITW), 2011 IEEE**, Paraty-RJ, Brazil, October 2011, pp. 693–697.

[34] J. S. de Lemos-Neto & V. C. da Rocha Jr., Códigos ciclicamente permutáveis derivados de códigos constacíclicos, In: **Anais do XXIX Simpósio Brasileiro de Telecomunicações (XXIX SBrT)**, Curitiba-PR, Brasil, Outubro 2011, pp. 1–5.

[35] D. H. Smith & S. Perkins, Cyclically permutable representations of cyclic codes, *Discrete Applied Mathematics*, v. 156, n. 1, pp. 76 – 81, 2008. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0166218X07003538.

[36] J. S. de Lemos-Neto & V. C. da Rocha Jr., Sequências de protocolo para o canal de colisão sem realimentação, In: **Anais do XXXI Simpósio Brasileiro de Telecomunicações** (**XXXI SBrT**), Fortaleza-CE, Brasil, Setembro 2013, pp. 1–5.

[37] P. Neumann, On a class of cyclically permutable error-correcting codes, *IEEE Transactions on Information Theory*, v. IT-10, n. 1, pp. 75–78, January 1964.

[38] D. Maracle & C. Wolverton, Generating cyclically permutable codes, *IEEE Transactions on Information Theory*, v. IT-20, n. 4, pp. 554–555, July 1974.

[39] G. Redinbo & J. Wolcott, Systematic construction of cyclically permutable code words, *IEEE Transactions on Communications*, v. IT-23, n. 7, pp. 786–789, July 1975.

[40] A. Lundqvist, On the construction of constant weight cyclically permutable codes using cyclic codes, In: **Proceedings of IEEE International Symposium on Information Theory**, June 1994, pp. 285.

[41] O. Moreno, Z. Zhang, P. Kumar, & V. Zinoviev, New constructions of optimal cyclically permutable constant weight codes, *IEEE Transactions on Information Theory*, v. IT-41, n. 2, pp. 448–455, March 1995.

[42] J. S. de Lemos-Neto & V. C. da Rocha Jr., Cyclically permutable codes specified by roots of generator polynomial, *Electronics Letters*, v. 50, n. 17, pp. 1202–1204, August 2014.

[43] A. Hocquenghem, Codes correcteurs d'erreurs, *Chiffres*, v. 2, pp. 147–156, 1959.

[44] R. C. Bose & D. K. Ray-Chaudhuri, On a class of error correcting binary group codes, *Information and Control*, v. 3, pp. 68–79, March 1960.

[45] ——, Further results on error correcting group codes, *Information and Control*, v. 3, pp. 279–290, September 1960.

[46] W. W. Peterson, Encoding and error-correction procedures for de Bose-Chaudhuri codes, *IRE Transactions on Information Theory*, v. IT-6, pp. 459–470, September 1960.

[47] I. S. Reed & G. Solomon, Polynomial codes over certain finite fields, *Journal of the Society for Industrial and Applied Mathematics*, v. 8, n. 2, pp. 300–304, June 1960.

[48] S. B. Wicker & V. K. Bhargava, **Reed-Solomon Codes and Their Applications**. IEEE Press, 1994.

[49] V. C. da Rocha Jr., Maximum distance separable multilevel codes, *IEEE Transactions on Information Theory*, v. IT-30, n. 3, pp. 547–548, May 1984.

[50] ——, Algebraic decoding of a class of multilevel pseudocyclic codes, *IEE Electronics Letters*, v. 25, n. 5, pp. 341–342, March 1989.

[51] V. C. da Rocha Jr., R. M. C. de Souza, & P. G. Farrell, Multilevel pseudocyclic codes, *Journal of Information and Optimization Sciences*, v. 11, n. 1, pp. 101–106, January 1990.

[52] A. Krishna & D. Sarwate, Pseudocyclic maximum-distance-separable codes, *IEEE Transactions on Information Theory*, v. IT-36, n. 4, pp. 880–884, May 1990.

[53] J. S. de Lemos-Neto, Construção de sequências de protocolo para o canal de colisão sem realimentação, Dissertação (Mestrado em Eng. Elétrica), Depto. de Eletrônica e Sistemas, UFPE, Recife, 2011.

[54] J. M. Jensen, Cyclic concatenated codes with constacyclic outer codes, *IEEE Transactions on Information Theory*, v. IT-38, n. 3, pp. 950–959, May 1992.

[55] W. S. Wong, New protocol sequences for random-access channels without feedback, *IEEE Transactions on Information Theory*, v. IT-53, n. 6, pp. 2060–2071, June 2007.

[56] T. Beth, D. Jungnickel, & H. Lenz, **Design Theory – Volume I**, 2ª ed. Cambridge University Press, 1999.

Sobre o Autor

O autor nasceu em Bezerros, Pernambuco, no dia 27 de Novembro de 1980. Formou-se em Engenharia Elétrica, modalidade Eletrônica, pela Universidade Federal de Pernambuco em 2004. É membro da *Sociedade Brasileira de Telecomunicações*. Seus interesses de pesquisa incluem Teoria da Informação, Códigos Corretores de Erro, Sistemas de Comunicação Digital, Processamento Digital de Sinais e Matemática Aplicada.

Endereço: Av. Benedita de Andrade, 92
55660 – 000 São Sebastião
Bezerros – PE
Brasil

e-mail: netosam@msn.com

Esta tese foi diagramada usando LaTeX 2_ε[1] pelo autor.

[1] LaTeX 2_ε é uma extensão do LaTeX. LaTeX é uma coleção de macros criadas por Leslie Lamport para o sistema TeX, que foi desenvolvido por Donald E. Knuth. TeX é uma marca registrada da Sociedade Americana de Matemática ($\mathcal{AMS}$). O estilo usado na formatação desta tese foi escrito por Dinesh Das, Universidade do Texas. Modificado por Renato José de Sobral Cintra (2001) e por André Leite Wanderley (2005), ambos da Universidade Federal de Pernambuco. Sua última modificação ocorreu em 2015 realizada por José Sampaio de Lemos Neto, também da Universidade Federal de Pernambuco.