UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

Este exemplar corresponde à redação final da tese defendida por José Carmelo Interlando e aprovada pela Comissão Julgadora em 12 / 12 / 94.

Reginaldo Palazzo Júnior
Orientador

UMA CONTRIBUIÇÃO À CONSTRUÇÃO E DECODIFICAÇÃO DE CÓDIGOS LINEARES SOBRE GRUPOS ABELIANOS VIA CONCATENAÇÃO DE CÓDIGOS SOBRE ANÉIS DE INTEIROS RESIDUAIS

Orientado : José Carmelo Interlando

Orientador : Prof. Dr. Reginaldo Palazzo Jr.

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, como parte dos requisitos exigidos para a obtenção do título de *DOUTOR EM ENGENHARIA ELÉTRICA*

Dezembro/1994

Com amor aos meus pais
Neno e Lenil
e ao meu irmão
Luciano
With love and affection to my sweetheart
Patricia,
for her encouragement and understanding.

AGRADECIMENTOS

A Deus, por ter permitido que eu concluísse mais uma etapa desta longa jornada;

Em especial ao Prof. Dr. Reginaldo Palazzo Jr., meu orientador de tese e amigo. Durante todo este período, foram fundamentais: o seu constante estímulo ao trabalho que eu vinha realizando, mesmo nos momentos de maior dificuldade; a valorização de cada pequeno passo conquistado por mim; o seu vasto conhecimento técnico que permitiu que eu sempre trilhasse por caminhos seguros e como não poderia deixar de ser, a sua amizade, paciência e atenção;

Aos professores da banca examinadora: Prof. Dr. Michele Elia (Politécnico di Torino, Itália), Prof. Dr. Valdemar Cardoso da Rocha Jr. (UFPE, Recife), Prof. Dr. Celso de Almeida (FEE-UNICAMP) e Prof. Dr. Jaime Portugheis (FEE-UNICAMP), este último também por valiosas discussões;

Aos primos Orlando e Luzilene P. Fernandes pela cessão do equipamento computacional durante a edição da versão em inglês deste trabalho;

Aos amigos que contribuíram de diversas formas: Antônio de Andrade e Silva, Bartolomeu e Fátima Uchôa, Carlos Eduardo Câmara (Dinho), Cristina Palazzo, João Roberto Gerônimo e Mário Massato Harada;

Ao CNPq, pelo apoio financeiro durante os dois primeiros anos e à CAPES pelo apoio financeiro durante o doutorado "sanduíche" (3º ano) e o último ano;

À Universidade Estadual de Campinas (UNICAMP);

À Universidade de Notre Dame (Notre Dame, Indiana, USA), pela estadia durante o programa de doutorado "sanduíche";

A Elza Aoki, pelo trabalho impecável na digitação dos manuscritos, pela atenção e paciência.

RESUMO

Códigos lineares e sistemáticos sobre grupos não abelianos são assintoticamente ruins, i.e., a razão d^*/n (onde d^* é a distância mínima e n é o comprimento das palavras-código) tende a zero à medida que n aumenta. Com isto, códigos lineares sobre grupos abelianos são investigados em maior profundidade. O desempenho de um código linear e sistemático sobre um grupo abeliano G é limitado pelo desempenho de um subcódigo linear e sistemático definido sobre um subgrupo H de G, onde H é isomorfo ao grupo aditivo de um anel de inteiros residuais Z_q, onde q é uma potência de primo. É feita então uma proposta de construção que consiste em concatenar m códigos sobre anéis do tipo Z_q (onde o inteiro m depende de certas propriedades estruturais de G) para se obter um código linear sobre G. A decodificação é realizada por m decodificadores, sendo um para cada código sobre um anel do tipo Z_q. Devido à forte relação entre códigos sobre grupos abelianos e códigos sobre anéis de inteiros residuais, é feita inicialmente uma revisão geral acerca destes últimos, considerando geração e decodificação. Aplicações da teoria de códigos sobre grupos para a teoria de códigos do espaço Euclidiano são discutidas brevemente.

ABSTRACT

Linear systematic codes over non-abelian groups are asymptotically bad, i.e., the ratio d^*/n (where d^* and n represent the minimum distance and length of the codewords, respectively) cannot be bounded away from zero. Thus, attention is focused on linear codes over abelian groups. The performance (rate and minimum distance) of a linear systematic code over an abelian group G is shown to be bounded by the performance of some linear systematic subcode defined over a subgroup H of G, where H is isomorphic to the additive group of an integer residue ring Z_q, where q is a power of prime. From this, linear codes over abelian groups are obtained via generalized concatenation of m codes over rings (m is an integer depending on certain structural properties of the abelian group). Decoding is made by m decoders, i.e., one decoder for each component code defined over some ring of the type Z_q. Due to the strong relationship between codes over abelian groups and codes over integer residue rings, we first make a review of the latter, considering encoding and decoding. Applications of the theory of codes over groups to the theory of Euclidean space codes are briefly discussed.

ÍNDICE

INTRODUÇÃO

APRESENTAÇÃO E BREVE HISTÓRICO

Uma forma simplificada de um sistema de comunicações digitais está ilustrada na Figura 1:

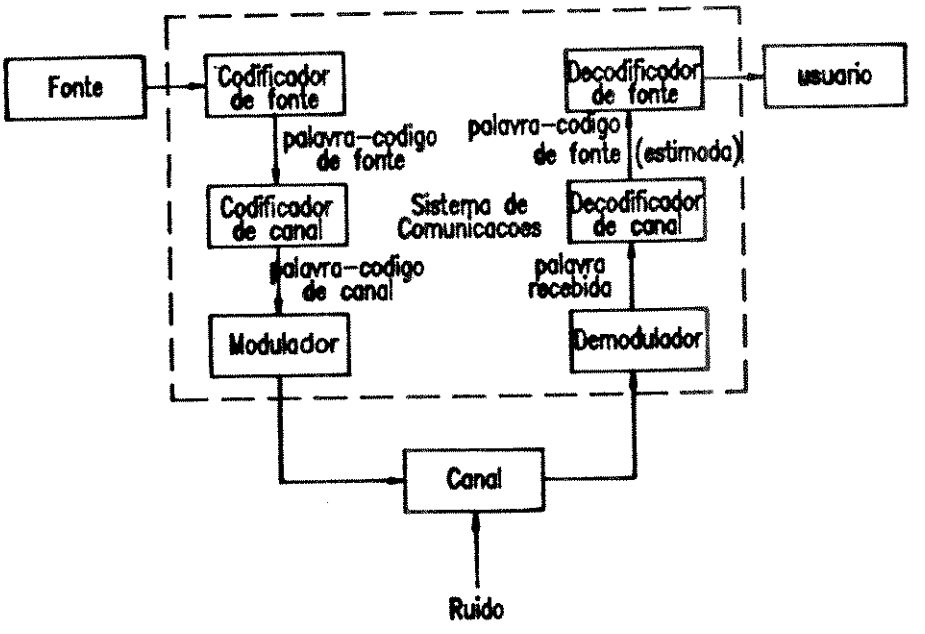

Figura 1 - Diagrama de blocos de um sistema de comunicações digitais.

Iremos descrever brevemente o modelo do sistema de comunicações da Fig. 1 para depois introduzirmos o problema que se pretende analisar neste trabalho. Este sistema de comunicações conecta uma fonte de dados a um usuário através de um canal o qual poderá ser, por exemplo, uma fibra óptica, um cabo coaxial, uma fita ou disco magnético e até mesmo a atmosfera e o espaço.

Os dados que entram neste sistema de comunicações a partir da fonte de dados são primeiramente processados pelo codificador de fonte, o qual tem como objetivo representá-la de forma mais compacta, retirando redundância. A saída desse codificador são seqüências chamadas palavras-código de fonte.

Essas seqüências são então processadas por um codificador de canal o qual introduz redundância transformando-as em outras seqüências denominadas palavras-código de canal. Cada símbolo na palavra-código de canal é representado por bits (dígitos binários) no caso de sinalização binária. Caso se use mais do que dois sinais (e.g., q sinais), não temos bits, mas sim dígitos de um alfabeto q-ário.

A seguir, o modulador converte cada símbolo da palavra-código de canal em um símbolo analógico correspondente o qual é transmitido através do canal.

Freqüentemente o canal fica sujeito a vários tipos de ruído, distorções e interferências e com isso a saída pode diferir da entrada. O demodulador então converte - sempre fazendo a melhor estimativa - cada sinal recebido da saída do canal em um dos possíveis símbolos que compõem as palavras-código de canal.

A seqüência demodulada de símbolos é chamada a palavra recebida. Obviamente, devido ao ruído, nem sempre a palavra recebida corresponde à palavra-código de canal enviada. É aí então que o decodificador de canal se utiliza da redundância contida na palavra-código de canal para corrigir os erros e então produzir uma estimativa da palavra-código de fonte. O decodificador de fonte processa esta última palavra estimada e a transforma numa seqüência de dados a qual é entregue ao usuário.

Neste trabalho iremos focalizar apenas nos blocos codificador e decodificador de canal (Fig. 1), os quais iremos chamar simplesmente codificador e decodificador.

A teoria de códigos corretores de erros teve início com o trabalho de Shannon [1]. Neste trabalho Shannon mostrou que associado a cada canal de comunicações existe um valor $\mathscr{C}$ que possui, em linhas gerais, o seguinte significado: sempre que a taxa de informação $\mathscr{R}$ (expressa em dígitos por segundo) for menor do que $\mathscr{C}$, então é possível se projetar um sistema de comunicações (usando códigos corretores de erros) tal que a probabilidade de erro seja tão pequena quanto se queira.

Desde então, pesquisadores vêm procurando encontrar estes bons códigos (previstos pela Teoria de Shannon) e também projetar decodificadores eficientes para os mesmos. Várias teorias sofisticadas surgiram e muitos resultados importantes foram alcançados; contudo, a pesquisa destes tópicos ainda continua de forma intensa nos dias de hoje.

DESCRIÇÃO DO PROBLEMA

Vamos agora descrever mais precisamente o problema que será tratado neste trabalho.

Em geral, estaremos usando símbolos (componentes da palavra-código) que possuam certa estrutura matemática, a fim de que a codificação e a decodificação sejam facilitadas. Será também interessante que o conjunto de sinais usados tenha alguma estrutura matemática a fim de que a sua geração não se torne demasiadamente complexa.

Em última instância, estas características simplificarão de certa forma a análise de desempenho do sistema de comunicações como um todo. O que se espera é que os bons esquemas de codificação e decodificação (previstos pela Teoria de Shannon) possuam também certa regularidade! (isto ainda não foi provado, mas conjectura-se que sim).

No caso de o ruído ser do tipo Gaussiano, branco e aditivo (AWGN, "Additive White Gaussian Noise", do inglês), o espaço métrico correspondente é o espaço Euclidiano N-dimensional. É isto que estaremos supondo durante todo este trabalho.

O trabalho aqui apresentado tem seus fundamentos baseados no artigo [2] o qual reúne e generaliza todas as teorias que até então buscavam o projeto de sinais (códigos) ótimos em termos de desempenho, mas que também possuíssem alguma estrutura regular. A seguir faremos uma breve revisão dos conceitos e definições dados em [2] para que possamos colocar o problema.

Uma ***isometria*** u do espaço Euclidiano N-dimensional é uma transformação $u : \mathbb{R}^N \to \mathbb{R}^N$ que preserva distâncias Euclidianas, i.e., $\|u(x)-u(y)\|^2 = \|x-y\|^2$, para quaisquer $x, y \in \mathbb{R}^N$ e onde $u(x)$ e $u(y)$ denotam as imagens de x e y sob a transformação u.

Seja S um conjunto de pontos (finito ou infinito) do $\mathbb{R}^N$. Uma isometria u que deixa S invariante, $u(S) = S$, é uma ***simetria*** de S. As simetrias de S formam um grupo sob composição, o grupo de simetrias $\Gamma(S)$ de S.

Um conjunto de sinais S ***geometricamente uniforme*** (GU) consiste de um conjunto de pontos em um espaço Euclidiano N-dimensional tendo um grupo de simetria transitivo, i.e., dados quaisquer pontos s_1 e s_2 em S, existe uma isometria que transforma s_1 em s_2, deixando S invariante.

Um ***grupo gerador*** U(S) de S é um subgrupo do grupo de simetrias de S, $\Gamma(S)$, que é minimamente suficiente para gerar S a partir de um ponto inicial $s_0 \in S$. Iremos assumir que o mapeamento $m : U(S) \rightarrow S$ definido por $m(u) = u(s_0)$ é biunívoco.

Uma ***partição geometricamente uniforme*** S/S' é uma partição de um conjunto de sinais GU com grupo gerador U(S) que é induzida por um subgrupo normal U' de U(S). Os elementos da partição são os subconjuntos de S que correspondem às classes laterais de U' em U(S). Seja agora um grupo abstrato G, isomorfo a U(S)/U'. Um ***rotulamento isométrico*** $m : G \rightarrow S/S'$ é um rotulamento de pontos de S por elementos de G, induzido pelo isomorfismo entre G e U(S)/U'.

Considere um grupo (alfabeto) G, um conjunto de índices I, um código C (subgrupo do espaço de rótulos G^I), uma partição geometricamente uniforme S/S' e um rotulamento $\underline{m} : G^I \rightarrow (S/S')^I$ (extensão do rotulamento isométrico $m : G \rightarrow S/S'$). Então um ***código de classes laterais generalizado***, denotado por $\mathbb{C}(S/S'; C)$, é a união disjunta

$$\mathbb{C}(S/S'; C) = \bigcup_{\underline{c} \in C} \underline{m}(\underline{c})$$

do conjunto de seqüências de subconjuntos $\underline{m}(\underline{c}) = \{m(c_k), k \in I\}$, $\underline{c} \in C$; i.e., $\underline{m}(\underline{c})$ é a seqüência de subconjuntos selecionados pela seqüência de rótulos $\underline{c} \in C$ via o mapeamento de rótulos m.

Uma seqüência de sinais $\underline{s}$ é uma seqüência código em $\mathbb{C}(S/S'; C)$ se $\underline{s} \in \underline{m}(\underline{c})$ para algum $\underline{c} \in C$, i.e., se $\{s_k \in m(c_k), k \in I\}$. Portanto, um código de classes laterais generalizado é um subconjunto do espaço de seqüências S^I, o conjunto de todas as seqüências de elementos do conjunto de sinais S.

Em [2], também está mostrado que $\mathbb{C}(S/S'; C)$ é geometricamente uniforme. Portanto, um código $\mathbb{C}$, geometricamente uniforme, pode ser obtido via o mapeamento de palavras-código $\underline{c} \in C$ em seqüências de sinais $s \in S$ de acordo com o mapeamento m estendido componente a componente, i.e.,

$$\mathbb{C} = m(C) = \{s \in (\mathbb{R}^N)^I : s = \underline{m}(\underline{c}), \underline{c} \in C\} \qquad (1)$$

Esta Construção (1) de fato nos indica como obter códigos GU em maiores dimensões a partir de códigos GU em dimensões menores, i.e., códigos geometricamente uniformes "elementares".

> *O principal objetivo é sempre construir códigos GU que tenham seqüências o mais distante possível uma das outras e uma maneira de se atingir isto é construir códigos sobre o grupo $G \cong U(S)/U'$, que tenham também as suas seqüências de símbolos o mais distante possível umas das outras. É este o problema a ser tratado neste presente trabalho.*

O conjunto de índices I pode ser semi-infinito, e.g., $I = Z^+$ (códigos de treliça) ou finitos, e.g., $I = \{1, 2, ..., n\}$ (códigos de bloco). Neste trabalho nos restringiremos somente ao estudo de códigos de bloco.

É possível mostrar que códigos sobre grupos não-abelianos finitos são assintoticamente ruins, i.e., a razão entre a distância mínima de Hamming e o comprimento destes códigos tende a zero à medida que o comprimento aumenta. Devido a isto, direcionaremos nosso estudo para códigos sobre grupos abelianos. A construção que será proposta está baseada no fato de que todo grupo abeliano finito pode ser escrito como um produto direto de grupos cíclicos, sendo cada um destes isomorfo ao grupo aditivo de um anel de inteiros residuais. Como conseqüência, as palavras de um código linear sobre um grupo abeliano serão obtidas através da concatenação de palavras-código pertencentes a códigos sobre anéis de inteiros residuais. É esta a razão pela qual códigos sobre anéis de inteiros têm importância relevante neste presente trabalho. Dessa forma, organizamos este trabalho em cinco capítulos, da seguinte maneira:

Capítulo 1 : Inicialmente faremos uma revisão dos conceitos básicos dos quais depende este trabalho como um todo. São conceitos elementares relacionados a Álgebra Abstrata e a Códigos Corretores de Erros. A seguir serão descritas as principais classes de códigos sobre anéis de inteiros residuais, i.e., códigos cíclicos, de Hamming, Reed-Solomon e BCH;

Capítulo 2 : São apresentados algoritmos para a decodificação de códigos de Hamming sobre anéis de inteiros residuais, Z_q, e decodificação de códigos Reed-Solomon e BCH (também definidos sobre Z_q), este último algoritmo sendo a generalização do Algoritmo de Berlekamp-Massey para anéis. É mostrada ainda a aplicação deste algoritmo para a síntese de registros de deslocamento que geram seqüências prescritas (finitas) em anéis comutativos;

Capítulo 3 : É apresentada uma outra generalização do Algoritmo de Berlekamp-Massey, agora para a geração de múltiplas seqüências e conseqüente aplicação para a decodificação de códigos cíclicos até o limitante de Hartmann-Tzeng;

Capítulo 4 : É feita uma revisão da teoria apresentada em [29] acerca dos códigos sobre grupos. Aprresentamos uma proposta de construção (e decodificação) de códigos sobre grupos abelianos, via concatenação de códigos sobre anéis de inteiros residuais;

Capítulo 5 : Faremos as nossas conclusões e indicaremos alguns tópicos para estudos futuros.

CAPÍTULO 1

CÓDIGOS LINEARES SOBRE ANÉIS DE INTEIROS

Neste capítulo faremos uma descrição das principais classes de códigos corretores de erros definidos sobre anéis de inteiros residuais. Serão abordados os códigos cíclicos, de Hamming, Reed-Solomon e BCH. Aqui serão tratadas a construção destes códigos bem como a análise de parâmetros tais como taxa e distância mínima de Hamming. No Capítulo 2 serão apresentados os respectivos métodos de decodificação.

O estudo de códigos definidos sobre anéis é importante nos seguintes aspectos:

a) tais códigos servirão de base para a construção de códigos definidos sobre grupos abelianos, como mostraremos no Capítulo 4;

b) a grande parte dos códigos conhecidos na literatura estão definidos sobre corpos, estruturas algébricas mais complexas que anéis. Portanto, códigos sobre anéis podem ser mais apropriados a determinadas aplicações;

c) recentemente tem-se mostrado que algumas classes de códigos binários não lineares podem ser obtidos através de códigos lineares sobre anéis, através de transformações apropriadas [3]. Estes códigos não lineares possuem, em geral, desempenho superior ao de códigos lineares, mas têm a desvantagem de serem de difícil geração/decodificação quando comparados com os códigos lineares.

Iniciaremos o capítulo com a Seção 1.1, voltada para a descrição de conceitos e propriedades da Álgebra Abstrata e Códigos Corretores de Erros, os quais se farão necessários para o entendimento deste e dos demais capítulos.

1.1. Conceitos Preliminares de Álgebra Abstrata e Códigos Corretores de Erros

Esta seção tem a finalidade de familiarizar o leitor com as ferramentas básicas e essenciais das quais faremos uso ao longo do texto. Essas ferramentas compreendem conceitos da Álgebra Abstrata e Códigos Corretores de Erros (C.C.E.) e não requerem conhecimentos específicos prévios em cada uma das áreas.

Como os conceitos de C.C.E. são baseados em definições e resultados da Álgebra Abstrata, será conveniente começarmos pelo estudo desta última. A referência adotada é [4].

Definição 1.1 : Uma **operação binária** $*$ sobre um conjunto S é uma regra que associa algum elemento de S a cada par ordenado (a, b) de elementos de S. ($a * b$ denotará o elemento associado a (a, b) através de $*$).

Uma operação binária sobre S deve associar a cada par ordenado (a, b) um elemento que está também em S. Este requerimento de que o elemento deva estar em S também é conhecido como a ***condição de fechamento***; i.e., exige-se que S seja ***fechado*** sob a operação binária. Note também que um único elemento é associado a cada par ordenado de S.

Exemplo 1.1 : A operação de adição usual é uma operação binária sobre $\mathbb{R}$ (o conjunto dos números reais). Por outro lado ela não é uma operação binária sobre $\mathbb{R}^*$ (o conjunto dos números reais não nulos) já que, por exemplo, 2+ (-2) não pertence a $\mathbb{R}^*$.

Definição 1.2 : Uma operação binária sobre um conjunto S é **comutativa** se $a * b = b * a$ para todo $a, b \in S$.

Definição 1.3 : Uma operação sobre um conjunto S é **associativa** se $(a * b) * c = a * (b * c)$, para quaisquer que sejam $a, b, c \in S$.

Definição 1.4 : Um **grupo** $< G, * >$ é um conjunto G, junto com uma operação binária sobre G, tal que as seguintes propriedades são satisfeitas:

G1) A operação binária é associativa;
G2) Existe um elemento e em G tal que $e * x = x * e = x$ para todo $x \in G$. (este elemento é um elemento identidade para * sobre G);
G3) Para cada a em G, existe um elemento a' em G com a propriedade de que $a' * a = a * a' = e$. (O elemento a' é um inverso de a com respeito a operação *).

Obs.: É possível mostrar que em um grupo G com operação binária *, o elemento identidade e é único. Também o inverso de cada elemento é único.

Definição 1.5 : Um grupo G é **abeliano** se a sua operação binária for comutativa.

Exemplo 1.2 : O conjunto Z^+ (os inteiros positivos) com operação + não é um grupo, já que não existe o elemento identidade para + em Z^+.

Exemplo 1.3 : O conjunto Z com operação + é um grupo. Todas as propriedades da Definição 1.4 são satisfeitas. Além disso, o grupo é abeliano.

Definição 1.6 : Se um subconjunto H de um grupo G é fechado sob a operação binária de G e se H forma um grupo, então H é um **subgrupo** de G.

Definição 1.7 : Se G é um grupo, então o subgrupo que consiste do próprio G é o **subgrupo impróprio** de G. Todos os outros subgrupos são **subgrupos próprios**. O subgrupo $\{e\}$ é o **subgrupo trivial** de G. Todos os outros subgrupos são **não-triviais**.

Definição 1.8 : Seja H um subgrupo de um grupo G. Diz-se que H é **normal** em G, ou H é um **subgrupo normal** de um grupo G, se qualquer uma das seguintes condições equivalentes ocorrer:

(i) $gH = Hg$ para todo $g \in G$;
(ii) $g^{-1}Hg = H$ para todo $g \in G$;
(iii) $g^{-1}Hg \subset H$ para todo $g \in G$;
(iv) $g^{-1}hg \in H$ para todo $g \in G$ e $h \in H$.

Teorema 1.1 : Seja G um grupo e seja $a \in G$. Então

$$H = \{a^n \mid n \in Z\}$$

é um subgrupo de G e é o menor subgrupo de G que contém a, isto é, todo subgrupo contendo a, contém H. Este grupo H é o **subgrupo cíclico** $< a >$ de G gerado por a.

Definição 1.9 : Dados um grupo G e um elemento $a \in G$, se ocorrer que

$$G = \{a^n \mid n \in Z\},$$

então a é um **gerador** de G e o grupo $G = < a >$ é **cíclico**.

Definição 1.10 : Seja n um inteiro positivo fixo e sejam h e k quaisquer inteiros. O resto r quando $h + k$ é dividido por n é a **soma de h e k módulo n**. Analogamente definimos o **produto de h e k módulo n** como sendo o resto da divisão de $h \,.\, k$ por n.

Exemplo 1.4 : O conjunto $\{0, 1, 2, ..., n\text{-}1\}$ é um grupo cíclico sob a operação de soma módulo n.

Definição 1.11 : A função (ou mapeamento) $\phi : G \rightarrow H$ é um **homomorfismo** de G em H se para quaisquer x, y em G, tem-se que $\phi(xy) = \phi(x).\phi(y)$. (Note que o produto xy é realizado em G, enquanto que o produto $\phi(x).\phi(y)$ é realizado em H).

Definição 1.12 : Diz-se que dois grupos G e H são **isomorfos** (denota-se isto por $G \cong H$) se existe um mapeamento bijetor ϕ de G em H tal que $\phi(xy) = \phi(x).\phi(y)$, para quaisquer x, y em G.

Definição 1.13 : Sejam $G_1, G_2, ..., G_n$ grupos. Seja $G_1 \times G_2 \times ... \times G_n$ o produto Cartesiano de $G_1, G_2, ..., G_n$. Defina uma operação binária em $G_1 \times G_2 \times ... \times G_n$ por: $(g_1, g_2, ..., g_n).(g'_1, g'_2, ..., g'_n) = (g_1g'_1, g_2g'_2, ..., g_ng'_n)$. Então mostra-se que $\Pi_{i=1}^{n} G_i = G_1 x G_2 x ... x G_n$ sob este "produto" é um grupo, chamado o **produto direto** de $G_1, G_2, ..., G_n$.

Obs.: Quando a operação binária em cada G_i é comutativa, é comum usarmos a notação aditiva em $\Pi_{i=1}^{n} G_i$ e referirmos a $\Pi_{i=1}^{n} G_i$ como a **soma direta** dos grupos G_i, a qual é denotada por $\oplus_{i=1}^{n} G_i$.

Teorema 1.2 : Sejam $H_1, H_2, ..., H_n$ subgrupos de um grupo G. Então G é isomorfo ao produto direto $\Pi_{i=1}^{n} H_i$ se, e somente se, cada um dos seguintes conjuntos (A) e (B) de condições equivalentes ocorrer:

(A)
- (i) Todos os H_i's são normais em G;
- (ii) $G = H_1 H_2 ... H_n$;
- (iii) $H_1 H_2 ... H_{i-1} H_{i+1} H_{i+2} ... H_n \cap H_i = \{e\}$.

(B)
- (i) Todo elemento de G é expresso como um produto $g = h_1 h_2 h_3 ... h_n$, $h_i \in H_i$;
- (ii) Os fatores h_i são univocamente determinados por g;
- (iii) Todo elemento de H_i comuta com todo elemento de H_j, quando $i \neq j$.

Definição 1.14 : Um **anel** $< R, +, \cdot >$ é um conjunto R junto com duas operações binárias + e $\cdot$ (as quais chamamos adição e multiplicação) tal que as seguintes propriedades são satisfeitas:

R1) $< R, + >$ é um grupo abeliano;

R2) A multiplicação é associativa;

R3) Para todos $a, b, c \in R$, valem a lei distributiva à esquerda, $a(b + c) = (ab) + (ac)$ e a lei distributiva a direita, $(a + b)c = (ac) + (bc)$.

Exemplo 1.5 : O conjunto $\{0, 1, ..., n-1\}$ forma um anel sob as operações de soma e produto módulo n.

Definição 1.15 : Dizemos que N é um **subanel** de um anel R se $N \subseteq R$ e N também forma um anel com as operações + e · herdadas de R.

Definição 1.16 : Um subanel N de um anel R é um **ideal à direita (à esquerda)** em R se $Nb \subseteq N$ ($bN \subseteq N$) para todo $b \in R$. Se N é simultaneamente um ideal à direita e à esquerda em R, dizemos que N é um **ideal** em R.

Sejam R um anel e N um ideal de R. Então, N determina uma relação de equivalência em R dada por:

$$x \sim x' \Leftrightarrow x - x' \in N$$

Estas classes de equivalência são os conjuntos

$$\bar{x} = x + N = \{x + n \mid n \in N\},$$

com $x \in R$, e são chamadas ***classes laterais*** (aditivas) de N em R. Todo elemento de R está contido em exatamente uma classe lateral $\bar{x}$. Denotaremos por R/N o conjunto dessas classes laterais. Define-se em R/N duas operações, a partir das operações de adição e multiplicação em R, da seguinte forma:

$$\bar{x} + \bar{y} = (x + N) + (y + N) = \overline{x + y} = (x + y) + N$$

e

$$\bar{x}.\bar{y} = (x + N).(y + N) = \overline{xy} = x.y + N$$

Estas operações são a "soma" e a "multiplicação" em R/N, respectivamente. Pode-se verificar que estas definições das operações não dependem da escolha de representantes em

$\bar{x}$ e $\bar{y}$. Mais ainda, mostra-se que R/N é um anel em relação às operações introduzidas, conhecido como ***anel quociente de*** R ***módulo*** N.

Definição 1.17 : Um anel no qual a multiplicação é comutativa é um **anel comutativo**. Um anel R com uma identidade multiplicativa 1 tal que $1.x = x.1 = x$ para todo $x \in R$ é um **anel com unidade**. Uma identidade multiplicativa em um anel é uma **unidade**.

Definição 1.18 : Seja R um anel:

a) Um elemento não nulo a de R é chamado um **divisor de zero** se existe um elemento não nulo b em R tal que $a.b = 0$ ou $b.a = 0$;

b) Considere que R seja um anel com unidade. Um elemento a de R é chamado **inversível** ("unit", em inglês) em R se existe a^{-1} em R tal que $a.a^{-1} = a^{-1}.a = 1$.

Definição 1.19 : Um **corpo** é um anel comutativo com uma unidade e tal que todo elemento não-nulo é inversível.

Definição 1.20 : Seja F um corpo. Um **espaço vetorial** sobre F consiste de um grupo abeliano V sob adição junto com uma operação de multiplicação por escalar de cada elemento de V por cada elemento de F à esquerda, tal que para todo a, $b \in F$ e α, $\beta \in V$, valem as seguintes propriedades:

V1) $a\alpha \in V$;
V2) $a(b\alpha) = (ab)\alpha$;
V3) $(a + b)\alpha = (a\alpha) + (b\alpha)$;
V4) $a(\alpha + \beta) = (a\alpha) + (a\beta)$;
V5) $1\alpha = \alpha$, onde 1 é a unidade multiplicativa de F.

Os elementos de V são **vetores** e os elementos de F são **escalares**.

Definição 1.21 : Seja V um espaço vetorial sobre F. Os vetores em um subconjunto $S = \{\alpha_i \mid i \in I\}$ (onde I é um conjunto de índices) de V **geram** V se para todo $\beta \in V$ tem-se que:

$$\beta = a_1.\alpha_{i_1} + a_2.\alpha_{i_2} + \ldots + a_n.\alpha_{i_n}$$

para algum conjunto de $a_j \in F$ e $\alpha_{i_j} \in S, j = 1, ..., n$. Um vetor

$$\sum_{j=1}^{n} a_j . \alpha_{i_j}$$

é chamado de **combinação linear** dos α_{i_j}.

Definição 1.22 : Um espaço vetorial sobre um corpo F tem **dimensão finita** se existe um subconjunto finito de V cujos vetores geram V.

Definição 1.23 : Os vetores em um subconjunto $S = \{\alpha_i \mid i \in I\}$ de um espaço vetorial V sobre um corpo F são **linearmente independentes** sobre F se

$$\sum_{j=1}^{n} a_j . \alpha_{i_j} = 0$$

implica que $a_j = 0$ para $j = 1, ..., n$.

Se os vetores não são linearmente independentes sobre F, dizemos que eles são **linearmente dependentes** sobre F.

Definição 1.24 : Se V é um espaço vetorial sobre um corpo F, os vetores em um subconjunto $B = \{\beta_i \mid i \in I\}$ de V formam uma **base** para V sobre F se eles geram V e são linearmente independentes. O número de elementos de B é conhecido como a **dimensão** de V sobre F. (É possível mostrar que este número independe da escolha da base B).

Definição 1.25 : Seja R um anel. Um **R-módulo** (à esquerda) consiste de um grupo abeliano M junto com uma operação de multiplicação externa de cada elemento de M por cada elemento de R (à esquerda) tal que para todo $\alpha, \beta \in M$ e $r, s \in R$, as seguintes condições são satisfeitas:

1) $(r\alpha) \in M$
2) $r(\alpha + \beta) = r\alpha + r\beta$
3) $(r + s)\alpha = r\alpha + s\alpha$
4) $(rs)\alpha = r(s\alpha)$

Um R-módulo assemelha-se com um espaço vetorial exceto que os escalares precisam somente formar um anel. Se R é um anel com unidade e $1\alpha = \alpha$ para todo $\alpha \in M$, então M é um R-módulo unitário.

Definição 1.26 : Um R-módulo M é **cíclico** se existe $\alpha \in M$ tal que $M = \{r\alpha \mid r \in R\}$.

Portanto, um R-módulo cíclico é gerado por um único elemento. A idéia de um conjunto de geradores para um R-módulo é a generalização natural da idéia de um conjunto de vetores geradores de um espaço vetorial.

Definição 1.27 : Uma **álgebra** consiste de um espaço vetorial V sobre um corpo F, junto com uma operação binária de multiplicação sobre o conjunto V de vetores, tal que para todo $a \in F$ e $\alpha, \beta, \gamma \in V$, as seguintes condições são satisfeitas:

1) $(a\alpha)\beta = a(\alpha\beta) = \alpha(a\beta)$
2) $(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$
3) $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$

V será uma **álgebra associativa** sobre F se além das três condições acima,

4) $(\alpha\beta)\gamma = \alpha(\beta\gamma)$, para todo $\alpha, \beta, \gamma \in V$.

Teorema 1.2 : As classes residuais de polinômios módulo um polinômio $f(x)$ de grau n formam uma álgebra de dimensão n sobre o corpo dos coeficientes.

Teorema 1.3 : Seja $p(x)$ um polinômio com coeficientes em um corpo F. Se $p(x)$ for irredutível em F, i.e., se $p(x)$ não possuir fatores com coeficientes em F, então a álgebra de polinômios sobre F módulo $p(x)$ será um corpo.

O corpo formado tomando-se polinômios sobre um corpo F módulo um polinômio irredutível $p(x)$ de grau m é chamado de ***corpo de extensão*** de grau m sobre F.

No exemplo 1.5, vimos que as classes residuais módulo um dado inteiro n formam

um anel sob adição e multiplicação módulo n, denotado por Z_n. Ainda é possível mostrar que quando n é um primo p, então estas classes residuais formam um corpo de p elementos, chamado corpo de Galois e denotado por GF(p).

Um resultado da álgebra nos diz que o anel de polinômios sobre qualquer corpo finito tem pelo menos um polinômio irredutível de todo grau. O corpo de polinômios sobre GF(p) módulo um polinômio irredutível de grau m é chamado corpo de Galois de ordem p^m e é denotado por GF(p^m). Com isto concluímos que é sempre possível encontrar um corpo de $q = p^m$ elementos, onde p é um primo. Pelo Teorema 1.2, o corpo GF(p^m) é um espaço vetorial de dimensão m sobre GF(p) e, portanto, tem p^m elementos. O seguinte teorema será útil quando formos construir tal corpo.

Teorema 1.4 : Seja F* o conjunto dos q - 1 elementos não-nulos de GF(q), onde $q = p^m$. Então, F* é um grupo cíclico multiplicativo de ordem p^m - 1.

A unicidade de GF(p^m) é garantida pelo seguinte:

Teorema 1.5 : Todos os corpos finitos de ordem p^m são isomorfos (diz-se que dois corpos F e G são isomorfos se existe uma bijeção de F em G a qual preserva adição e multiplicação).

Exemplo 1.6 : Vamos construir o corpo de Galois de 16 elementos, GF(2^4), o qual será formado a partir das classes residuais de polinômios (sobre GF(2)) módulo $x^4 + x + 1$ (polinômio irredutível sobre GF(2)). Este polinômio, $x^4 + x + 1$, módulo ele próprio é zero, isto é, $x^4 + x + 1 = 0$. Seja α o elemento pertencente ao corpo de extensão tal que $\alpha^4 + \alpha + 1 = 0$, i.e., $\alpha^4 = \alpha + 1$. Este identidade será usada repetidamente para formarmos representações polinomiais de GF(2^4). Por exemplo,

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

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

E assim prosseguimos até $\alpha^{14} = 1 + \alpha^3$. Lembre que $\alpha^{15} = 1$ pois o grupo multiplicativo possui p^m-1 elementos, ou seja $2^4 - 1 = 15$ elementos.

Agora iremos introduzir os conceitos e resultados clássicos a respeito de códigos corretores de erros, dos quais faremos uso ao longo do texto (ver [7] e [9]).

Definição 1.28 : Um **espaço de seqüências** A^I é o conjunto de todas as seqüências $c = \{c_i \mid i \in I\}$ de elementos c_i sobre algum alfabeto A, onde I é um conjunto de índices.

Definição 1.29 : Um **código** C sobre um alfabeto A é qualquer subconjunto não-vazio do espaço de seqüências A^I.

Ao longo de todo o texto, estaremos trabalhando somente com alfabetos finitos. A princípio, o alfabeto A pode ser qualquer conjunto de símbolos. Entretanto, muitas vezes é conveniente que o mesmo seja "estruturado" a fim de que a codificação e a decodificação sejam simplificadas. Por alfabetos "estruturados", entendemos aqueles que formam alguma estrutura algébrica, tal como corpo, anel ou grupo.

Definição 1.30 : Um **código de bloco** C de **comprimento** n sobre um alfabeto A é qualquer subconjunto não-vazio do conjunto A^n de todas as seqüências (que agora chamamos "palavras") $c = \{c_i \mid 1 \leq i \leq n\}$.

Quando o conjunto de índices é semi-infinito, e.g., I = {0, 1, 2, 3, ...} dizemos que o código é "de árvore". Durante todo o texto, estaremos focalizando somente em códigos de bloco (onde o conjunto I é finito), salvo menção explícita em contrário.

- **<u>Parâmetros de um código de bloco</u>**

Na definição de um código de bloco, implicitamente foi também definido o parâmetro n, que é o comprimento do código. Iremos agora definir mais três parâmetros importantes: a dimensão, a taxa e a distância mínima de Hamming.

Definição 1.31 : A **dimensão** de um código C é dada por $k = log_{|A|} |C|$ símbolos por bloco, onde $|\cdot|$ denota a cardinalidade do conjunto.

Definição 1.32 : A **taxa** de um código C é dada por $r = k/n$, onde k é a dimensão e n é o comprimento do código.

Definição 1.33 : A **distância de Hamming** $d_H(\underline{c}, \underline{c}')$ entre duas palavras $\underline{c}$ e $\underline{c}' \in A^n$ é o número de componentes nas quais elas diferem.

Repare que as três propriedades de métrica estão sendo satisfeitas:

D1) $d_H(\underline{c}, \underline{c}') \geq 0$ e $d_H(\underline{c}, \underline{c}') = 0 \Leftrightarrow \underline{c} = \underline{c}'$
D2) $d_H(\underline{c}, \underline{c}') = d_H(\underline{c}', \underline{c})$
D3) $d_H(\underline{c}, \underline{c}') + d_H(\underline{c}', \underline{c}'') \geq d_H(\underline{c}, \underline{c}'')$

Definição 1.34 : Seja um código C de comprimento n e tal que $|C| \geq 2$. A **distância mínima de Hamming** de C, denotada por $d_{min}(C)$ é dada por:

$$d_{min}(C) = \min_{\substack{\underline{c},\underline{c}' \in C \\ \underline{c} \neq \underline{c}'}} d_H(\underline{c},\underline{c}')$$

Note que $1 \leq d_{min}(C) \leq n$. Se $|C| = 1$, adota-se $d_{min}(C) = \infty$, por convenção.

Um código de bloco C de comprimento n, dimensão k e distância mínima de Hamming $d = d_{min}(C)$ é chamado de um **(n, k, d)-código**. O seguinte teorema nos dá um limitante superior para a distância mínima em função dos parâmetros n e k.

Teorema 1.6: Para qualquer (n, k, d)-código, vale a seguinte relação:

$$d \leq n - k + 1$$

Ela é conhecida como "Limitante de Singleton".

Outras distâncias ainda podem ser definidas tais como a distância de Lee ou a distância Euclidiana, esta última quando estamos associando uma modulação (um conjunto de pontos do $\mathbb{R}^n$) ao código. Entretanto, durante todo o texto estaremos usando a distância de Hamming, salvo menção explícita em contrário.

Estaremos também trabalhando basicamente com alfabetos que formam alguma das estruturas algébricas: grupo, anel ou corpo. O objetivo final, como sugere o título deste

trabalho, é caracterizar códigos sobre grupos. Entretanto, códigos sobre corpos e anéis que serão estudados neste e no capítulo seguinte terão um papel fundamental.

Faremos agora uma breve revisão de códigos sobre corpos finitos.

Definição 1.35 : Se o alfabeto A for um corpo finito, $F_q = GF(q)$, dizemos que um código C sobre F_q é linear se ele formar um subespaço vetorial do espaço vetorial F_q^n.

Considerando então um código linear como um subespaço vetorial (finito), temos que ele admite uma base, i.e., um conjunto mínimo de vetores (palavras código) linearmente independentes que são capazes de gerá-lo através de combinação linear. Os coeficientes desta combinação linear pertencem a $GF(q)$. Do ponto de vista prático, isto é bastante vantajoso já que não se necessita armazenar todas as palavras-código, pois qualquer uma delas é obtida por combinação linear de apenas algumas.

Se a base for formada por k vetores, então a dimensão do código tanto no sentido de subespaço vetorial como no sentido da Definição 1.25, será k. Isto pode ser visto da seguinte forma: a partir da combinação linear de k vetores linearmente independentes é possível se gerar q^k vetores distintos (o número total de palavras-código). Daí vem que a dimensão do código é dada por $log_{|A|}|C| = log_q q^k = k$. Portanto, k vetores linearmente independentes de F_q^n geram um *(n, k)*-código linear de bloco sobre F_q, onde q é uma potência de primo.

Sejam $\underline{g}_1, \underline{g}_2, \ldots, \underline{g}_k$ vetores de F_q^n que geram um *(n, k)*-código linear C. Então qualquer palavra $\underline{v}$ pode ser expressa na forma $\underline{v} = \underline{u} \cdot G$, onde

$$G = \begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \\ \vdots \\ \underline{g}_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}.$$

Esta matriz G é conhecida como a ***matriz geradora*** de C. O vetor $\underline{u}$, formado pelos k coeficientes da combinação linear que gera $\underline{v}$, é chamado de vetor mensagem

associado à palavra-código $\underline{v}$. Quando as k primeiras componentes de $\underline{v}$ formam o próprio vetor $\underline{u}$, dizemos que o código é ***sistemático***. Isto ocorre sempre que G possuir a forma:

$$G = [\, I_k \mid P_{k \times n-k} \,]$$

onde I_k é a matriz identidade k x k e P é uma matriz k x n-k. Neste caso, dizemos que G está na forma escada (ou sistemática).

Repare que como G tem posto k, é sempre possível reduzi-la à forma escada através de operações elementares com as suas linhas. Qualquer uma das operações abaixo é considerada uma operação elementar:

a) troca de duas linhas;
b) multiplicação de uma linha por um elemento não-nulo do corpo F_q;
c) adição de qualquer múltiplo de uma linha à outra.

Quando a matriz geradora de um código pode ser obtida através de operações elementares com a matriz geradora de outro código, dizemos que os códigos são equivalentes, i.e., os respectivos conjuntos de palavras-código coincidem.

Iremos agora descrever de maneira alternativa (ainda através do uso de matrizes) um *(n, k)*-código linear sobre F_q.

Define-se o produto de duas n-uplas $\underline{v} = (v_1, v_2, \ldots, v_n)$ e $\underline{w} = (w_1, w_2, \ldots, w_n) \in F_q^n$ como $\underline{v} \cdot \underline{w} = v_1w_1 + v_2w_2 + \ldots + v_nw_n$ onde as operações entre as componentes são as operações do corpo base. Diz-se que $\underline{v}$ e $\underline{w}$ são ortogonais se $\underline{v} \cdot \underline{w} = 0$ e que $\underline{v}$ é ortogonal a um subespaço W se $\underline{v}$ for ortogonal a todo $\underline{w} \in W$.

Mostra-se facilmente que o conjunto de todas as n-uplas ortogonais a um subespaço vetorial V de n-uplas, formam também um subespaço W de n-uplas, chamado o espaço-nulo de V. Mais ainda, se um vetor $\underline{w}$ for ortogonal a todo vetor da base de um espaço V, então $\underline{w}$ pertence ao espaço-nulo de V.

Um outro resultado ainda afirma que se a dimensão de um subespaço de n-uplas for k, então a dimensão do espaço-nulo será n-k, i.e., a base do subespaço nulo é formada por n-k vetores. Isto implica que associada a uma matriz geradora G k x n (de posto k) de um *(n, k)*-código linear, existe uma matriz H do tipo *(n-k)* x n (de posto n-k) tal que

$$G \cdot H^t = 0$$

Portanto, um *(n, k)*-código linear C sobre F_q pode ser descrito como sendo o conjunto de todas as *n*-uplas $\underline{v}$ tais que

$$\underline{v} \cdot H^t = 0$$

para alguma matriz H *(n-k)* x *n* de posto *k*. Esta matriz H é conhecida como a ***matriz verificação de paridade*** de C.

A matriz H pode ainda ser vista como a matriz geradora de um *(n, n-k)*-código linear $C^{\perp}$ sobre F_q, conhecido como o ***código ortogonal*** ou ***dual de C***.

Definição 1.36 : O **Peso de Hamming** de um vetor $\underline{v} \in F_q^n$, denotado por $w_H(\underline{v})$ é o número de componentes não-nulas de $\underline{v}$.

Como a distância de Hamming entre dois vetores $\underline{v}_1$ e $\underline{v}_2$ é o número de posições nas quais eles diferem, então a distância entre $\underline{v}_1$ e $\underline{v}_2$ é dada por $w_H(\underline{v}_1 - \underline{v}_2)$. Se $\underline{v}_1$ e $\underline{v}_2$ são palavras de um código linear de bloco, então $\underline{v}_1 - \underline{v}_2$ também o é. Portanto, a distância entre quaisquer duas palavras é igual ao peso de uma outra palavra (pertencente também ao código) e daí então a distância mínima de um código linear ser igual ao peso mínimo de suas palavras não-nulas. Esta propriedade, apesar de simples, será bastante usada.

O teorema a seguir, juntamente com o seu corolário, terão papel fundamental quando formos determinar a distância mínima de um código linear sobre F_q.

Teorema 1.7 [9] : Se H for a matriz verificação de paridade de um código de bloco sobre F_q, então para cada palavra-código com peso de Hamming *w*, existe uma relação de dependência linear entre *w* colunas de H. Por outro lado, para cada relação de dependência linear de *w* colunas de H, existe uma palavra-código de peso *w*.

Corolário 1.1 : Se H for a matriz verificação de paridade de um código C, então o mesmo possui distância mínima *d* se, e somente se, todo conjunto de *d* - 1 colunas de H for linearmente independente e existir um conjunto de *d* colunas linearmente dependente.

Comentaremos agora a importância prática dos parâmetros distância mínima, capacidade de correção e taxa de um código corretor de erro. Analisaremos as possibilidades de construção de "bons códigos", i.e., aqueles que corrijam até um determinado número de erros t (que o canal possa introduzir) e que transmitam informação a uma taxa específica r.

Seja o sistema de comunicações como mostrado na Fig. 1.1:

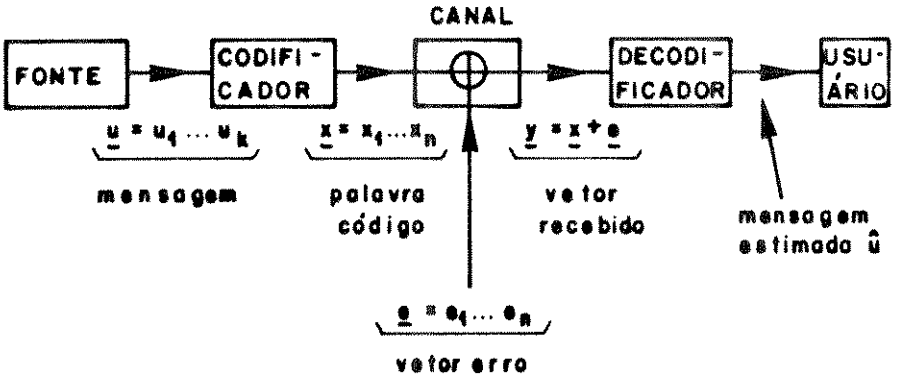

Figura 1.1 - Diagrama de blocos de um sistema de comunicação geral.

Suponha agora que a mensagem $\underline{u} = (u_1, ..., u_k)$ corresponda à palavra código $\underline{x} = (x_1, ..., x_n)$ a qual é enviada através do canal. Devido ao ruído introduzido pelo canal, o vetor recebido $\underline{y} = (y_1, ..., y_n)$ pode ser diferente de $\underline{x}$. Define-se o vetor erro por:

$$\underline{e} = \underline{y} - \underline{x} = (e_1, ..., e_n)$$

O decodificador deve decidir a partir de $\underline{y}$ qual foi a palavra-código transmitida. Por causa da natureza aleatória do ruído, o decodificador não é capaz de determinar com certeza absoluta qual foi o vetor erro que realmente ocorreu.

Diante disto ele escolherá o vetor erro que tenha ocorrido com maior probabilidade, dado que $\underline{y}$ foi recebido. Isto minimizará a probabilidade de erro de palavra. Supondo que todas as palavras sejam igualmente prováveis, esta estratégia é ótima no sentido em que ela minimiza a probabilidade de que o decodificador cometa um erro e é conhecida como ***decodificação de máxima verossimilhança***.

Diz-se que não ocorreram erros durante a transmissão em uma determinada palavra-código quando o vetor erro for todo nulo, i.e., $e_i = 0$, $1 \leq i \leq n$. Seja agora X uma variável

aleatória que representa o número de erros ocorridos em uma transmissão, i.e., X é o número de componentes não-nulas de $\underline{e}$. Em geral, se os erros forem independentes, tem-se que:

$$P(X = 0) > P(X = 1) > P(X = 2) \ldots > P(X = n)$$

Portanto, a estratégia de decodificação de máxima verossimilhança será tomar o vetor erro de menor peso de Hamming, i.e., recebida uma palavra, o decodificador a decodificará como a palavra-código mais próxima, no espaço de Hamming. Em particular, este tipo de decodificação é conhecida por ***decodificação pelo vizinho mais próximo***.

Uma maneira de implementá-la é a seguinte: recebida uma palavra, compare-a com todas as palavras-código e tome a mais próxima. Entretanto, em geral, o número de palavras-código é muito grande e este método torna-se altamente complexo e impraticável. Um dos objetivos da teoria de códigos corretores de erros é o de se propor métodos mais eficientes de decodificação.

Dizemos que um decodificador ***detecta*** erros quando ele determina que a palavra recebida não é uma palavra-código. Se, além disso, o decodificador ainda for capaz de determinar corretamente a palavra-código transmitida, dizemos que ele ***corrige*** erros. Um decodificador que apenas detecta erros é usado em sistemas de comunicações onde é possível a retransmissão de dados, enquanto que um decodificador que corrige erros é usado mais comumente em sistemas de comunicações onde não é possível essa retransmissão.

Os teoremas a seguir fornecem a capacidade de correção e deteção de erros associada a um código de bloco (linear ou não linear) com distância mínima de Hamming d.

Teorema 1.8 [9] : Um código de bloco com distância mínima d, se usado somente para deteção, é capaz de detectar a presença de erros se ocorrerem até d - 1 erros. Se usado somente para correção, ele é capaz de corrigir até $\left\lfloor \frac{d-1}{2} \right\rfloor$ erros, onde $\lfloor x \rfloor$ indica o maior inteiro menor ou igual a x.

Teorema 1.9 [9] : Um código de bloco com distância mínima d é capaz de detectar até λ erros e corrigir até μ erros, simultaneamente, se $\lambda + \mu + 1 \leq d$ e $\mu \leq \lambda$.

Portanto, quanto mais erros desejarmos corrigir e/ou detectar, maior deve ser a distância mínima do código.

Um teste comum realizado para se detectar a presença de erros em códigos lineares sobre F_q é o cálculo do vetor síndrome. Lembre que um vetor $\underline{v} \in F_q^n$ é palavra-código se, e somente se,

$$\underline{v} \cdot H^t = \underline{0}$$

onde $H_{n-k \times n}$ é a matriz verificação de paridade de um *(n, k)*-código linear C sobre F_q.

Define-se a ***síndrome*** $\underline{s}$ de um vetor recebido $\underline{r}$ através da relação:

$$\underline{s} = \underline{r} \cdot H^t$$

Note que $\underline{s} = 0$ se, e somente se, $\underline{r} \in C$, i.e., $\underline{r}$ é uma palavra-código. Portanto, se não ocorrerem erros durante a transmissão, a síndrome do vetor recebido é o vetor todo nulo, mas entretanto o contrário não é verdade já que dependendo do vetor erro, o vetor recebido poderá ser uma outra palavra-código, diferente da enviada.

Definição 1.37 : A **probabilidade de erro de palavra**, P_{err}, para um determinado esquema de decodificação é a probabilidade de que a saída do decodificador seja uma palavra-código diferente da enviada.

Se o código corretor de erro (linear ou não) possuir M palavras $\underline{v}^{(1)}, \underline{v}^{(2)}, \ldots, \underline{v}^{(M)}$ as quais são usadas com igual probabilidade, temos que:

$$P_{err} = \frac{1}{M} \cdot \sum_{i=1}^{M} \text{Prob}\{\text{saída do decodificador} \neq \underline{v}^{(i)} \mid \underline{v}^{(i)} \text{ foi enviada}\}$$

O princípio básico da teoria de códigos corretores de erros é a introdução de redundância nas mensagens enviadas. Quanto mais redundância for introduzida nas palavras de um código, maior serão a sua distância mínima e a sua capacidade de correção de erros e com isto a probabilidade de erro diminui.

Entretanto, a taxa do código, i.e., o número de símbolos efetivamente transmitidos por bloco diminui à medida que a redundância aumenta. O que se deseja na prática é construir códigos que apresentem boas taxas e baixas probabilidades de erro. Podemos perceber que esses objetivos são conflitantes, mas ainda assim a teoria de Shannon [1] garante a existência de bons códigos.

Em [1], Shannon define um parâmetro chamado ***capacidade de canal*** (denotada por $\mathcal{C}$) o qual depende das características físicas do mesmo e mostra que é possivel transmitir informação a qualquer taxa menor do que a capacidade de canal com uma probabilidade de erro arbitrariamente pequena. Em outras palavras, para qualquer ***$\varepsilon > 0$, se $R < \mathcal{C}$, existe um (n,k)-código de taxa $r = k/n \leq R$ com probabilidade de erro $P_{err} < \varepsilon$, para n suficientemente grande.***

O que acabamos de enunciar é o Teorema de Codificação de Canal de Shannon e ele pode ser provado para qualquer que seja o alfabeto A usado. Os códigos que o satisfazem podem ser não-lineares. O nosso objetivo aqui será o de propor esquemas de codificação (juntamente com os respectivos métodos de decodificação) quando o alfabeto A é um grupo. Além disso estaremos focalizando em códigos lineares, i.e., aqueles códigos que formam um subgrupo de A^n.

Antes de chegarmos a isto, faz-se necessário o estudo de códigos sobre anéis, o que será feito neste e no próximo capítulo.

As definições dos parâmetros de um código de bloco (comprimento, taxa, dimensão, distância mínima e probabilidade de erro) continuam sempre as mesmas, não importando o alfabeto sobre o qual se está trabalhando (corpo, anel ou grupo).

Entretanto, as noções de linearidade são modificadas. Sabemos que se R é um anel, nem todos os R-módulos possuem uma base, no sentido de espaços vetoriais. Mais especificamente, podemos ter vetores gerando um R-módulo e que, entretanto, não são linearmente independentes. Mais ainda, nem todo conjunto linearmente independente de um R-módulo pode ser ampliado a uma base. Aqueles R-módulos que são gerados por um conjunto de vetores linearmente independentes são chamados de ***módulos livres***.

Definição 1.38 : Um **código linear C de comprimento *n* sobre um anel R** é um conjunto de n-uplas que formam um R-módulo.

A dimensão do código C é dada por $k = log_{|R|} |C|$. Caso o código forme um módulo livre, a dimensão é dada pelo número de vetores (n-uplas) que o geram. Na maioria das vezes estaremos estudando códigos que possuem uma base e que, portanto, podem ser descritos por matrizes geradoras e verificação de paridade.

Proposição 1.1 : Se C é um código linear sobre Z_{ab}, com $mdc(a, b) = 1$, então $C = C_a \oplus C_b$, i.e., C é a soma direta de dois subcódigos lineares definidos sobre Z_a e Z_b, respectivamente. Como conseqüências imediatas, temos que:

i) $d_{min}(C) = \min\{d_{min}(C_a), d_{min}(C_b)\}$;

ii) $|C| = |C_a| . |C_b|$, implicando, então, que a taxa de C é a soma das taxas de C_a e C_b, i.e., $R_C = R_{C_a} + R_{C_b}$.

Prova : Sendo (a,b) = 1, temos que $aZ_{ab} \cong Z_b$ e $bZ_{ab} \cong Z_a$. Então, $C_a \triangleq b.C$ e $C_b \triangleq a.C$ são subcódigos lineares de C sobre Z_a e Z_b, respectivamente. Além disso, $C_a \cap C_b = \underline{0}$ (a palavra toda-nula). Mais ainda, pela algoritmo de Euclides, podemos afirmar que existem elementos r e s pertencentes a Z_{ab} tais que:

$$1 = ra + sb$$

Repare que nesta expressão, as operações + e . são realizadas módulo ab. A partir dela, podemos concluir que toda palavra-código $\underline{c} \in C$ pode ser escrita na forma:

$$\underline{c} = a(r\underline{c}) + b(s\underline{c})$$

Note que $\underline{c}_b = a(r\underline{c}) \in C_b$ e $\underline{c}_a = b(s\underline{c}) \in C_a$. Portanto, cada palavra $\underline{c} \in C$ pode ser escrita como

$$\underline{c} = \underline{c}_a + \underline{c}_b$$

onde $\underline{c}_a \in C_a$ e $\underline{c}_b \in C_b$.

Finalmente, para demonstrarmos que a soma é direta, resta mostrar que a equação

$$\underline{0} = \underline{c}_a + \underline{c}_b$$

tem a única solução $\underline{c}_a = \underline{c}_b = \underline{0}$. Isto é imediato a partir do fato de que o inverso aditivo (em Z_{ab}) de um elemento não-nulo pertencente a Z_a não pode ser um elemento pertencente a Z_b. ■

Obs. :

1) Embora sempre estejamos supondo a distância de Hamming, note que esta Proposição 1 ainda continua válida para outras distâncias, tais como a Euclidiana ou a de Lee;

2) Se C é um código definido sobre Z_m, onde $m = p_1^{r_1} p_2^{r_2} \cdots p_k^{r_k}$ então $C = \bigoplus_{i=1}^{k} C_{p_i^{r_i}}$ onde $C_{p_i^{r_i}}$ é um código sobre $Z_{p_i^{r_i}}$. E, portanto :

   i) $d_{min}(C) = \min_i \left\{ d_{min}(C_{p_i^{r_i}}) \right\}$

   ii) $R_C = \sum_{i=1}^{k} R_{C_{p_i^{r_i}}}$

Com isto, nos restringiremos ao estudo de códigos sobre anéis da forma Z_q, onde q agora é uma potência de primo.

1.2. Códigos Cíclicos sobre Anéis de Inteiros Residuais

Nesta seção serão apresentadas as definições e teoremas acerca de códigos cíclicos sobre anéis Z_m ($m \geq 2$ e inteiro), as quais servirão de base para a construção de códigos BCH (Seção 1.5).

Definição 1.39 : Um **(n, k)-código linear sobre Z_m** é definido como um módulo livre de dimensão ***k*** no espaço das n-uplas Z_m^n.

Definição 1.40 : Um (n, k)-código linear C sobre Z_m é **cíclico** se sempre quando $\underline{x} = (a_0\ a_1\ a_2\ \ldots\ a_{n-1}) \in C$, então $\underline{x}^{(1)} \triangleq (a_{n-1}\ a_0\ a_1\ a_2 \cdots a_{n-2}) \in C$, com $a_i \in Z_m$, $0 \leq i \leq n-1$.

Seja a palavra código $\underline{v} = (v_0\ v_1\ \ldots\ v_{n-1})$ de um código cíclico C. Será útil

representá-la pelo polinômio

$$v(x) = v_0 + v_1 x + \ldots + v_{n-1} x^{n-1}$$

Se fizermos o produto x.v(x) (mod $x^n - 1$), o resultado será:

$$v'(x) = v_{r-1} + v_0 x + v_1 x^2 + \ldots + v_{n-2} x^{n-1}$$

o qual corresponde à palavra-código

$$v' = (v_{n-1}\ v_0\ v_1 \ldots\ v_{n-2}) = v^{(1)}$$

Portanto, $v^{(1)}(x)$ é obtida calculando-se o produto x.v(x) no anel quociente $R_n = Z_m[x] / < x^n - 1 >$. A soma de duas palavras código é realizada em $Z_m[x]$.

Teorema 1.10 : Um conjunto S de elementos em R_n corresponde a um código cíclico se, e somente se, S é um ideal em R_n.

Prova : Se S corresponde a um código cíclico linear então se $v_1(x)$ e $v_2(x) \in S$, temos que $v_1(x) \pm v_2(x) \in S$ também. Se $v(x) \in S$, então $x.v(x) \in S$ (pela definição de código cíclico). Logo se $w(x) = w_0 + w_1 x + \ldots + w_{n-1} x^{n-1} \in R_n$ e $v(x) \in S$ então $w(x).v(x) = w_0 v(x) + w_1 v(x).x + \ldots + w_{n-1} v(x) x^{n-1} \in S$ já que cada termo isoladamente pertence a S. Isto caracteriza S como sendo um ideal de R_n.

Por outro lado, se S é um ideal em R_n, então:

a) a soma de dois elementos em S é um elemento de S;

b) se $v(x) \in S$, então $x.v(x) \in S$.

Por a) e b) concluímos que S é um código cíclico. ■

Proposição 1.2 : Seja C um ideal (i.e., um código cíclico de comprimento *n*) em $R_n = Z_m[x]/ < x^n - 1 >$. Se o polinômio de menor grau em C, g(x), é mônico então g(x) é univocamente determinado e $C = < g(x) >$ (i.e., C é um ***ideal principal***).

Prova : Suponha então que g(x) (o polinômio de menor grau em C) seja mônico. Seja agora v(x) um polinômio em C. Então:

$$v(x) = g(x).b(x) + r(x), \quad \text{com } \partial r < \partial g$$

onde ∂p denota o grau do polinômio p(x). Pela definição de ideal, $r(x) \in C$. Isto contradiz a escolha de g(x), a menos que $r(x) \equiv 0$. Portanto, $v(x) = g(x).b(x)$. Ou seja, todo polinômio em C é múltiplo de g(x). Vamos agora provar a unicidade de g(x): suponha h(x) um polinômio de menor grau em C e mônico. Então $k(x) = g(x) - h(x)$ é um polinômio em C de grau menor que o de g(x) e h(x), o que é um absurdo. Portanto, g(x) é único. ■

Teorema 1.11 : Seja C um ideal em $R_n = Z_m[x] / < x^n - 1 >$, i.e., um código cíclico de comprimento *n* e g(x) um polinômio mônico de menor grau em C. Então $C = < g(x) >$, e o código C consiste de todos os múltiplos de g(x) (i.e., C é um ideal principal).

Prova : Seja v(x) um polinômio em C. Pelo algoritmo de divisão (para anéis comutativos) existem e são únicos os polinômios b(x) e r(x) tais que:

$$v(x) = g(x) \,.\, b(x) + r(x) \qquad \text{com} \qquad \partial r < \partial g,$$

onde ∂p = grau do polinômio p(x). Pela definição de ideal, $r(x) \in C$. Isto contradiz a escolha de g(x), a menos que $r(x) \equiv 0$. Portanto, $v(x) = g(x)b(x)$, ou seja, todo polinômio em C é múltiplo de g(x). ■

Teorema 1.12 : Seja C um ideal principal em R_n. Se o polinômio de menor grau em C, g(x), for mônico, então g(x) divide $x^n - 1$.

Prova : Suponha que g(x) seja o polinômio de menor grau em C. Então existem e são únicos os polinômios a(x) e r(x) tais que:

$$x^n - 1 = g(x) \,.\, a(x) + r(x), \quad \text{com} \quad \partial r < \partial g,$$

Disto segue que:

$$-g(x) \,.\, a(x) = -(x^n - 1) + r(x)$$

Portanto, r(x) está em $< g(x) >$ e tem grau menor do que g(x). Isto é uma contradição (já que o polinômio de menor grau em C é g(x)) a menos que $r(x) \equiv 0$. Isto implica que $g(x) \mid (x^n - 1)$. ■

Teorema 1.13: Se $g(x) \in C$ e $g(x) \mid (x^n - 1)$ então g(x) tem o menor grau em $< g(x) >$.

Prova: Suponha a existência de um polinômio b(x) pertencente a $< g(x) >$ tal que $\partial b < \partial g$. Como $b(x) \in < g(x) >$, então:

$$a(x) \cdot g(x) = (x^n - 1)d(x) + b(x)$$

Disto segue que:

$$b(x) = a(x)g(x) - (x^n - 1)d(x)$$

Como $g(x) \mid (x^n - 1)$, então $g(x) \mid (a(x)g(x) - (x^n - 1)d(x))$, o que implica que $g(x) \mid b(x)$. Isto é uma contradição, pois já tínhamos assumido que $\partial g > \partial b$. Portanto, se $g(x) \mid (x^n - 1)$, g(x) é o polinômio de menor grau em $< g(x) >$. ■

A relevância dos Teoremas 1.12 e 1.13 reside no fato de que eles nos fornecem um método de se construir códigos cíclicos sobre anéis de inteiros residuais, que é exatamente análogo ao da construção dos mesmos códigos sobre corpos finitos, isto é, fatorando-se o polinômio $(x^n - 1)$ no anel de interesse e, então, tomando-se um fator como o polinômio gerador. Mais adiante (na construção de códigos BCH), veremos como realizar esta fatoração, quando $n = p^r - 1$, onde p é primo e $r \geq 1$.

Teorema 1.14: Se $g(x) \mid (x^n - 1)$ e $\partial g = n - k$, então a dimensão de $C = < g(x) >$ é k. Se $g(x) = g_0 + g_1 x + \ldots + x^{n-k}$, então uma matriz geradora de C é dada por:

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & 1 & 0 & \cdots & 0 \\ 0 & 0 & g_0 & \cdots & g_{n-k-2} & g_{n-k-1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & g_0 & g_1 & & \cdots & 1 \end{bmatrix}$$

Prova : Os vetores g(x), x.g(x), x^2.g(x), ..., x^{k-1}.g(x) são linearmente independentes. Do contrário, existiriam elementos a_i, $0 \le i \le k-1$, não todos nulos, pertencentes a Z_m tais que:

$$a_0 \cdot g(x) + a_1.x \cdot g(x) + \ldots + a_{k-1} \cdot x^{k-1} \cdot g(x) = (a_0 + a_1.x + \ldots + a_{k-1} \cdot x^{k-1})g(x) = 0$$

Como g(x) é mônico e este produto tem grau menor que n, esta última igualdade só se verifica se $a_0 = a_1 = a_2 = \ldots = a_{k-1} = 0$. Portanto, estes vetores são linearmente independentes. Vamos agora mostrar que eles geram C. Seja então um polinômio v(x) em C. Sabemos que v(x) = c(x)g(x), onde $\partial c \le k-1$. Daí:

$$c(x)g(x) = (c_0 + c_1x + \ldots + c_{k-1}\, x^{k-1})g(x) = c_0\, g(x) + c_1\, g(x)x + \ldots + c_{k-1}\, g(x)x^{k-1}$$

Portanto, as linhas de G geram o (n, k) código cíclico sobre o anel Z_m. ■

Iremos agora descrever a forma sistemática para a matriz geradora G do código cíclico de forma análoga a de códigos cíclicos sobre corpos finitos.

O primeiro passo é dividir x^{n-k+i} $(0 \le i \le k-1)$ por g(x), o polinômio gerador de C:

$$x^{n-k+i} = a_i(x)g(x) + r_i(x)$$

$$r_i(x) = r_{i0} + r_{i1} \cdot x + \ldots + r_{i,n-k+1} \cdot x^{n-k-1}$$

Disso segue que o termo $(-r_i(x) + x^{n-k+i})$ é múltiplo de g(x) e, portanto, pertence a C. A matriz geradora fica:

$$G = \left[\begin{array}{c|cccc} -r_0(x) & 1 & 0 & \cdots & 0 \\ -r_1(x) & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -r_{k-1}(x) & 0 & 0 & \cdots & 1 \end{array}\right]$$

Proposição 1.3 : Se C é um código cíclico sobre Z_m, onde $m = p_1^{r_1}\, p_2^{r_2} \cdots p_m^{r_m}$, então

$$C = \bigoplus_{i=1}^{m} C_i,$$

onde C_i é um código cíclico sobre $Z_{p_i^{r_i}}$, $1 \le i \le m$.

Prova : Segue diretamente da Proposição 1.1.

1.3. Códigos de Hamming sobre Anéis

Iremos nesta seção descrever a generalização dos códigos de Hamming para anéis de inteiros residuais [11]. Também aqui a construção será feita de forma a garantir que estes códigos tenham distância mínima igual a três e, portanto, capacidade de correção de um erro aleatório. Devido à Proposição1.1, iremos focalizar em anéis da forma Z_q, onde $q = p^r (r \geq 1)$, com p primo. Como no caso de Códigos de Hamming sobre corpos, a matriz verificação de paridade será construída a partir de um parâmetro m (inteiro maior ou igual a dois) que representará o seu número de linhas. Os parâmetros do código tais como comprimento e dimensão serão dados também em função de m.

1.3.1. Construção da matriz verificação de paridade

Seja então o anel Z_q, onde $q = p^r$, com p primo. O conjunto dos divisores de zero deste anel é formado pelos elementos:

$$0,\ p,\ 2.p,\ 3.p,\ \ldots,\ (p^{r-1} - 1).p$$

Como podemos observar, este conjunto possui p^{r-1} elementos. Seja agora Z o conjunto dos $p^{(r-1)m}$ elementos em Z_q^m cujas componentes são elementos divisores de zero. Dizemos que a e b pertencentes a $(Z_q^m - Z)$ são equivalentes se eles forem linearmente dependentes (LD), i.e., se $\exists\ \alpha$ e β (não nulos) pertencentes a Z_q tais que:

$$\alpha \underline{a} + \beta \underline{b} = 0$$

A fim de construirmos a matriz verificação de paridade H de um código que tenha distância 3 (capacidade de correção de 1 erro), não podemos permitir que:

a) uma coluna de H pertença a Z, i.e., todas as suas componentes são elementos divisores de zero;

b) em H existam colunas equivalentes.

Por a), estamos evitando que o código tenha distância 1 e por b) que ele tenha distância 2. O teorema a seguir nos diz quantos elementos podemos escolher em Z_q^m, satisfazendo estas condições.

Teorema 1.15 [11] : O número de classes de equivalência em $(Z_q^m - Z)$ é dado por $(p^m - 1) / (p - 1)$.

Repare que este número depende somente de p e m e não de r. A conclusão deste Teorema 1.15 é que o número máximo de colunas na matriz verificação de paridade de um código sobre Z_q com distância 3 é dado por $n = (p^m - 1) / (p - 1)$ onde m representa o número de linhas de H. Isto implica que dado um parâmetro m ($m \geq 2$), podemos sempre construir um código sobre Z_q o qual possui os seguintes parâmetros:

1) comprimento $n = (p^m - 1) / (p - 1)$;

2) dimensão $k = n - m = (p^m - 1) / (p - 1) - m$;

3) distância mínima $d_{min} = 3$.

O teorema a seguir ajuda-nos a escolher quais elementos devemos escolher em Z_q^m que satisfaçam as condições **a)** e **b)**.

Teorema 1.16 : Sejam $\underline{a} = (a_1,..., a_m)$ e $\underline{b} = (b_1,..., b_m)$ elementos de Z_q^m e $\underline{a}' = (a_1',...,a_m')$ e $\underline{b}' = (b_1',...,b_m')$, com $a_i' = a_i \pmod p$ e $b_i' = b_i \pmod p$, $1 \leq i \leq m$, e onde (mod p) representa redução módulo p. Então, $\underline{a}$ e $\underline{b}$ são LD sobre Z_q se, e somente se, $\underline{a}'$ e $\underline{b}'$ são LD sobre Z_p ou GF(p), o corpo de Galois de ordem p.

Prova : I) Suponha inicialmente que $\underline{a} = (a_1, ..., a_m)$ e $\underline{b} = (b_1, ..., b_m)$ sejam LD sobre Z_q^m.

Disto segue que $\exists\ \alpha$ e $\beta \in Z_q$ (não ambos nulos) tais que:

$\alpha.a_i + \beta.b_i = 0$, para $1 \le i \le m$

Reduzindo módulo p esta última expressão, temos que:

$\bar{\alpha}.a'_i + \bar{\beta}.b'_i = 0$, para $1 \le i \le m$

onde $\bar{\alpha} \equiv \alpha \pmod p$ e $\bar{\beta} \equiv \beta \pmod p$ e, portanto, $\bar{\alpha}$ e $\bar{\beta} \in Z_p$. Disto podemos concluir que se $\underline{a}$ e $\underline{b}$ são LD sobre Z_q, então $\underline{a}'$ e $\underline{b}'$ são LD sobre Z_p.

II) Por outro lado, vamos agora supor que $\underline{a}'$ e $\underline{b}'$ sejam LD sobre Z_p, i.e., $\exists\ \alpha$ e $\beta \in Z_p$ tais que:

$\alpha.a'_i + \beta.b'_i = 0$, para $1 \le i \le m$.

Como $a_i \equiv a'_i \pmod p$ e $b_i \equiv b'_i \pmod p$, para $1 \le i \le m$, então:

$a_i = \ell_i.p + a'_i$ e $b_i = m_i.p + b'_i$

para algum conjunto de ℓ_i's e m_i's $(1 \le i \le m)$. Daí:

$\alpha(a_i - \ell_i.p) + \beta(b_i - m_i.p) = 0 \quad (1 \le i \le m)$

Multiplicando ambos os membros por p^{r-1}, vem que:

$\alpha'.a_i + \beta'.b_i = 0$, para $1 \le i \le m$

onde $\alpha' = \alpha.p^{r-1}$ e $\beta' = \beta.p^{r-1} \in Z_q$ (lembre que $\alpha.\ell_i.p^r$ e $\beta.m_i.p^r$ valem zero, quando reduzidos módulo $q = p^r$).

Juntando as partes I) e II), concluímos que $\underline{a}$ e $\underline{b}$ são LD sobre Z_q se, e somente se, $\underline{a}'$ e $\underline{b}'$ são LD sobre Z_p. ■

A partir dos Teoremas 1.15 e 1.16, temos o corolário a seguir:

Corolário 1.2 : A matriz verificação de paridade de um código de Hamming sobre Z_q pode ser tomada como sendo a matriz verificação de paridade de um código de Hamming sobre GF(p), aquela que possui como colunas todas as m-uplas (não nulas) pertencentes a Z_p e tais que quaisquer duas delas são linearmente independentes (sobre Z_p).

Observe que para um dado parâmetro m, os códigos de Hamming sobre GF(p) e sobre Z_q possuem o mesmo comprimento e dimensão, embora um número diferente de palavras-código.

Concluiremos esta seção com dois exemplos de códigos de Hamming sobre Z_q.

Exemplo 1.7: Construção de um código linear sobre Z_4. Seguindo a notação usada, temos:

$$p = 2 \quad \text{e} \quad q = p^2 = 4.$$

Vamos adotar $m = 3$. Daí vem:

$$n = (p^m - 1) / (p - 1) = 7 \quad \text{e} \quad k = n - m = 4.$$

A matriz verificação de paridade H do código de Hamming sobre GF(2) com comprimento $n = 7$ e dimensão $k = 4$ é dada por:

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

Pelo Corolário 1.2, esta é também a matriz verificação de paridade de um código de Hamming sobre Z_4 com comprimento $n = 7$, dimensão $k = 3$ e distância mínima $d_{min} = 3$.

Exemplo 1.8: Construção de um código linear sobre Z_9. Temos aqui:

$$p = 3; \quad q = p^2 = 9.$$

Como no Exemplo 1.7, considere $m = 3$. Disto segue que:

$$n = (p^m - 1) / (p - 1) = 13 \quad \text{e} \quad k = n - m = 10.$$

A matriz verificação de paridade H do código de Hamming sobre GF(3) com comprimento $n = 13$ e dimensão $k = 10$ é dada por:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 2 & 1 & 2 & 1 & 1 & 2 & 0 & 1 \\ 0 & 1 & 0 & 2 & 1 & 0 & 1 & 2 & 2 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Pelo Corolário 1.2, esta é também a matriz verificação de paridade de um código de Hamming sobre Z_9 com comprimento $n = 7$, dimensão $k = 3$ e distância mínima $d_{min} = 3$.

1.4. Códigos Reed-Solomon sobre Anéis

Nesta seção vamos apresentar a construção de códigos Reed-Solomon sobre anéis da forma Z_q onde $q = p^r$, com p um primo ímpar, i.e., $p \neq 2$ [11]. Esta construção é muito semelhante a de códigos Reed-Solomon sobre GF(q), o corpo de Galois de ordem q.

Seja $\alpha \in Z_q$ um elemento primitivo dos inteiros módulo p, i.e., α é tal que $\alpha^{p-1} = 1 \pmod p$ e $\alpha^s \neq 1 \pmod p$ se $1 \leq s \leq p-2$. Considere ainda $k = p - 2$ e m um inteiro não-negativo. Defina agora a matriz :

$$H = \begin{bmatrix} 1 & \alpha^{m} & \alpha^{2.m} & \cdots & \alpha^{k.m} \\ 1 & \alpha^{m+1} & \alpha^{2(m+1)} & \cdots & \alpha^{k(m+1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{m+d-2} & \alpha^{2(m+d-2)} & \cdots & \alpha^{k(m+d-2)} \end{bmatrix} \qquad (1.1)$$

Teorema 1.17: O espaço nulo da matriz H sobre Z_q é um código de comprimento (p - 1), distância mínima igual a d e dimensão (p - d).

Prova: Mostraremos que esta matriz H possui posto d - 1, i.e., qualquer conjunto de (d - 1) colunas é linearmente independente sobre Z_q. Além disso, estaremos provando também que as suas (d - 1) linhas são linearmente independentes. Seja então uma submatriz H' de H:

$$H' = \begin{bmatrix} \alpha^{j_1.m} & \alpha^{j_2.m} & \cdots & \alpha^{j_{d-1}.m} \\ \alpha^{j_1(m+1)} & \alpha^{j_2(m+1)} & \cdots & \alpha^{j_{d-1}(m+1)} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha^{j_1(m+d-2)} & \alpha^{j_2(m+d-2)} & \cdots & \alpha^{j_{d-1}(m+d-2)} \end{bmatrix}$$

Vamos mostrar que ela é não-singular, onde $j_i \in \{0, 1, 2, ..., k\}$ para $1 \leq i \leq d-1$.

O determinante de H' pode ser expresso na forma:

$$\det H' = \alpha^{(j_1+j_2+\ldots+j_{d-1})m} \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{j_1} & \alpha^{j_2} & \cdots & \alpha^{j_{d-1}} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha^{j_1(d-2)} & \alpha^{j_2(d-2)} & \cdots & \alpha^{j_{d-1}(d-2)} \end{vmatrix}$$

Portanto,

$$\det H' = \alpha^{(j_1+j_2+\ldots+j_{d-1})m} \cdot \prod_{i>\ell} \left(\alpha^{j_i} - \alpha^{j_\ell}\right)$$

Agora esta matriz H' será não-singular (i.e., terá posto completo d - 1) se, e somente se, esta última expressão resultar em um elemento inversível de Z_q. Vamos analisá-la. As potências de α serão sempre elementos inversíveis, já que α é um elemento inversível de Z_q. Os termos que fazem parte do produtório são da forma $\alpha^{j_i} - \alpha^{j_\ell} = \alpha^{j_i}(1 - \alpha^{j_\ell - j_i})$. Ambos os termos deste produto são inversíveis: o primeiro é uma potência de α e o segundo é da forma 1 - α^j, onde $1 \le j \le p$-2. Repare que para $1 \le j \le p$-2,

$\alpha^j = \ell.p + a$, onde $\quad 2 \le a \le p$-1

Disto segue que α^j - 1 não tem fator comum com p, o que implica que (α^j - 1, p^r) = 1 para $1 \le j \le p$-2. Portanto, α^j - 1 para $1 \le j \le p$-2, não pode ser um divisor de zero em Z_q, $q = p^r$. A conclusão é então que det H' é um elemento inversível em Z_q e com isto H tem posto (d - 1). Como conseqüência, $d_{min} \ge d$. ■

De fato, usando o limitante de Singleton (Teorema 1.6), vamos concluir que o valor da distância mínima é exatamente d. Recorde que:

$$d_{min} \le n - D + 1,$$

onde n é o comprimento das palavras-código e D é a dimensão do código.

No nosso caso,

$$n - k + 1 = (p - 2) + 1 = p - 1$$

$$D = n - (d - 1) = p - 1 - d + 1 = p - d$$

(Lembre que n é o número de colunas de H e D é o número de colunas subtraído do número de linhas).

Portanto, $n - D + 1 = d = d_{min}$. Este código é chamado de "separável à máxima distância" (SMD) já que o limitante de Singleton foi atingido com igualdade.

Obs. : O parâmetro k pode ainda assumir outros valores no intervalo $1 \le k \le p$-2; entretanto, os códigos obtidos possuirão uma taxa menor.

Exemplo 1.9 : Vamos construir um código Reed-Solomon sobre Z_{49} com $d_{min} = 5$. Temos : $p = 7$, $r = 2$, $q = p^r = 49$, $k = p - 2 = 5$, $\alpha = 3$ (elemento primitivo de Z_q), $D = p - d = 2$ (dimensão do código) e vamos supor que $m = 1$. Portanto,

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} \end{bmatrix} = \begin{bmatrix} 1 & 3 & 9 & 27 & 32 & 47 \\ 1 & 9 & 32 & 43 & 44 & 4 \\ 1 & 27 & 43 & 34 & 36 & 41 \\ 1 & 32 & 44 & 36 & 25 & 16 \end{bmatrix}$$

Da teoria apresentada acima, concluímos que H descreve um (6, 2, 5) - código linear sobre Z_{49}. Ele tem capacidade de correção de $(d_{min} - 1)/2 = 2$ erros aleatórios.

Obs. importantes :

Esta classe de códigos Reed-Solomon sobre anéis não é formada por códigos cíclicos. Por inspeção direta podemos comprovar que $\underline{v} = (4 \quad 44 \quad 31 \quad 27 \quad 1 \quad 0)$ é uma palavra do código do Exemplo 1.9, pois $\underline{v}.H^t = \underline{0}$, onde H é a matriz verificação de paridade. Entretanto, a palavra $\underline{v}^{(2)} = (1 \quad 0 \quad 4 \quad 44 \quad 31 \quad 27)$ não pertence a este código, já que $\underline{v}^{(2)}.H^t \neq \underline{0}$, Portanto, em geral, os códigos Reed-Solomon sobre anéis não são cíclicos.

Entretanto, podemos afirmar que todo polinômio v(x) de um código Reed-Solomon é múltiplo de um determinado polinômio mônico g(x), também pertencente ao código. Isto é justificado da seguinte maneira: pelo fato de que $\underline{v}.H^t = \underline{0}$ (onde H é a matriz verificação de paridade, como em (1.1)), então α^m, α^{m+1}, α^{m+2}, ..., α^{m+d-2} são raízes de v(x). Além disso, os coeficientes de v(x) pertencem ao anel Z_q (que possui identidade multiplicativa), o que leva a concluir que:

$$f_1(x) = x - \alpha^m, \quad f_2(x) = x - \alpha^{m+1}, \quad f_3(x) = x - \alpha^{m+2}, \ldots,$$

$$f_{d-1}(x) = x - \alpha^{m+d-2}$$

e

$$g(x) = \text{mmc}\,\{f_1(x), f_2(x), f_3(x), \ldots, f_{d-1}(x)\}$$

são fatores de v(x). Isto implica que todo polinômio código v(x) é múltiplo de g(x). Por outro lado, nem todo múltiplo de g(x) é um polinômio código!

1.5. Códigos BCH sobre Anéis

Primeiramente vamos recordar que em corpos GF(p), define-se um código BCH de comprimento n (onde mdc (p, n) = 1) e distância de projeto δ, como sendo um código cíclico cujo polinômio gerador tem como raízes

$$\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}$$

onde $\alpha \in GF(p^m)$ é uma raiz primitiva de $x^n - 1$, b é um inteiro não-negativo e m é tal que $n \mid p^m - 1$. Quando $n = p^m - 1$, temos um ***código BCH primitivo***.

Nesta seção iremos descrever a construção de códigos BCH sobre anéis da forma $Z_q(q = p^k, k \geq 2)$ a qual é bastante análoga a de códigos BCH sobre corpos [12]. A diferença é que aqui as raízes do polinômio gerador estarão em um ***anel de extensão*** de Z_q.

Vamos assumir também que a ordem do anel e o comprimento do código sejam relativamente primos, i.e., (p, n) = 1. Da Seção 1.2, recordamos que um código cíclico principal de comprimento n sobre Z_q é um ideal no anel de polinômios com coeficientes em Z_q módulo $x^n - 1$ e é gerado por qualquer polinômio g(x) que divide $x^n - 1$.

Seja $Z_q[y]$ o anel de polinômios na variável y sobre Z_q e $\phi(y)$ um polinômio primitivo de grau r, irredutível sobre GF(p) e conseqüentemente também sobre Z_q. Vamos denotar por GR(p^k, r) o anel $Z_q[y] / < \phi(y) >$, o conjunto de classes residuais de polinômios em y sobre Z_q módulo $\phi(y)$. Este anel consiste, portanto, de todos os polinômios de grau r - 1 cujas operações de adição e multiplicação são sempre realizadas módulo $\phi(y)$. Mais ainda, ele é um anel comutativo com identidade e é chamado de ***extensão de Galois*** de dimensão r de Z_q. A menos de isomorfismo, a extensão de dimensão r é única [13, Teorema XV.7].

Vamos introduzir agora algumas notações das quais faremos uso durante esta seção.

Estrutura	Ordem
R : GR(p^k, r);	$(p^k)^r$
K : GF(p^r);	p^r
R* : grupo dos elementos inversíveis de R;	$p^{(k-1)r}(p^r - 1)$
K* : grupo multiplicativo de GF(p^r);	p^r - 1

Definimos $R_m(n)$ como o resto da divisão de n por m e para um polinômio $f(y) = f_0 + f_1.y + f_2.y^2 + ... + f_n.y^n$,

$$R_m(f(y)) \equiv R_m(f_0) + R_m(f_1)y + ... + R_m(f_n)y^n$$

Procederemos agora, então, à fatoração de $x^n - 1$ a fim de construirmos o código cíclico. Como R^* é um grupo abeliano multiplicativo, ele pode ser expresso como um produto direto de grupos cíclicos. O grupo cíclico de interesse será aquele cujos elementos são todas as raízes de $x^n - 1$ para algum n tal que $(n, p) = 1$ (isto serve apenas para garantir que $x^n - 1$ não tenha fatores repetidos). Uma vez identificado este grupo, o problema da construção dos códigos reduz-se a escolher certos elementos do mesmo para serem as raízes do polinômio gerador g(x), o qual será um divisor de $x^n - 1$.

A seguir enunciaremos um conjunto de teoremas (1.18 - 1.21) [12] os quais servirão de base para a construção de G_n, o subgrupo cíclico de R^* (como um anel de extensão de Z_p^k) o qual contém todas as raízes de $x^n - 1$.

Teorema 1.18 : R^* tem um e somente um subgrupo cíclico de ordem relativamente prima a p. Este subgrupo cíclico tem ordem $p^r - 1$.

Teorema 1.19 : Suponha que f gere um subgrupo de ordem n em R^*, onde $(n, p) = 1$. Então, o polinômio $x^n - 1$ pode ser fatorado como $x^n - 1 = (x - f).(x - f^2) ... (x - f^n)$ se, e somente se, $R_p(f)$ tem ordem n em K^*.

Corolário 1.3 : Um polinômio h(x), divisor de $x^n - 1$, com coeficientes em Z_q pode ser fatorado sobre G_n como

$$h(x) = (x - \beta^{e_1})(x - \beta^{e_2}) \cdots (x - \beta^{e_\ell})$$

se, e somente se, $R_p(h(x))$ pode ser fatorado sobre $GF(p^r)$ como:

$$R_p(h(x)) = (x - (R_p(\beta))^{e_1})(x - (R_p(\beta))^{e_2}) \cdots (x - (R_p(\beta))^{e_\ell})$$

Teorema 1.20 : Suponha que $\bar{f} = R_p(f)$ gere um subgrupo cíclico de ordem n em K^*. Então f gera um subgrupo cíclico de ordem $n.d$ em R^*, onde d é um inteiro maior ou igual a 1 e f^d gera o subgrupo cíclico G_n de R^*.

Este Teorema 1.20 é útil na determinação do gerador de G_n e do Corolário 1.3 segue que $M_i(x)$, o polinômio minimal de β^i sobre R^* (onde β é primitivo em G_n) terá como raízes todos os elementos distintos na seqüência

$$\beta^i, (\beta^i)^p, (\beta^i)^{p^2}, \ldots, (\beta^i)^{p^r-1}$$

Portanto, $M_i(x)$ pode ser construído de maneira idêntica à construção de $m_i(x)$, o polinômio minimal de $R_p(\beta^i)$ sobre GF(p).

Vamos agora especificar um código BCH de comprimento n sobre Z_q (onde $n \mid p^r - 1$) em termos das raízes do polinômio gerador g(x) em G_n. Seja β um elemento primitivo de G_n. Se $\beta^{e_1}, \beta^{e_2}, \ldots, \beta^{e_j}$ são raízes de g(x), então podemos gerar um código tipo BCH com símbolos de Z_q se escolhermos g(x) como:

$$g(x) = \text{mmc}\left(M_{e_1}(x), M_{e_2}(x), \ldots, M_{e_j}(x)\right)$$

onde $M_{e_i}(x)$ é o polinômio minimal de β^{e_i}. Além disso,

$$\bar{g}(x) = R_p(g(x)) = \text{mmc}\,(m_{e_1}(x), m_{e_2}(x), \ldots, m_{e_j}(x))$$

onde $m_{e_i}(x)$ é o polinômio minimal de $R_p(\beta^{e_i})$, gera um código BCH com símbolos de GF(p).

<u>*Obs.*</u> : O método sistemático para a determinação do mmc de um conjunto de polinômios $\{p_1(x), p_2(x), \ldots, p_n(x)\}$, consiste em se calcular primeiramente o seu mdc (usando o algoritmo de Euclides) e depois o seu mmc é calculado através de:

$$\text{mmc}\,(p_1(x), p_2(x),\ldots, p_n(x)) = [\prod_{i=1}^{n} p_i(x)] / \text{mdc}\,(p_1(x), p_2(x),\ldots, p_n(x))$$

O Teorema 1.21 estabelece o limitante inferior para a distância mínima do código obtido.

Teorema 1.21: Seja g(x) o polinômio gerador de um código cíclico de comprimento n com símbolos de Z_q e sejam também $\alpha^{e_1}, \alpha^{e_2},\ldots, \alpha^{e_m}$ raízes de g(x) em G_n, onde α tem ordem n. Então, a distância mínima do código é maior do que o número máximo de inteiros consecutivos módulo n no conjunto $E = \{e_1, e_2, \ldots, e_m\}$.

Vamos agora fornecer um exemplo da construção desses códigos BCH sobre anéis Z_q $(q = p^k)$.

Exemplo 1.10: Vamos considerar um código BCH sobre o anel Z_8. Temos aqui $p = 2$ e $k = 3$. O polinômio $\phi(x) = x^4 + x + 1$ (de grau $r = 4$) é irredutível sobre GF(2) e conseqüentemente também sobre Z_8. O corpo $K = GF(p^r) = GF(2^4)$ é formado pelas classes residuais de polinômios em GF(2)[x] módulo $x^4 + x + 1$. Este corpo possui 16 elementos. Em notação de r-uplas, eles são:

$$\begin{aligned}
0 &= (0\,0\,0\,0)\\
1 &= (1\,0\,0\,0)\\
x = \alpha &= (0\,1\,0\,0)\\
x^2 = \alpha^2 &= (0\,0\,1\,0)\\
x^3 = \alpha^3 &= (0\,0\,0\,1)\\
x^4 = \alpha^4 &= (1\,1\,0\,0)\\
&\vdots\\
x^{14} = \alpha^{14} &= (1\,0\,0\,1)
\end{aligned}$$

O anel R (extensão de Galois de Z_8) é formado pelas classes residuais de polinômios sobre Z_8 módulo $x^4 + x + 1$, i.e., $R = GR(8, 4) = Z_8[x] / \langle x^4 + x + 1 \rangle$. Seja agora $f = (0\,1\,0\,0) \in R^*$; $\bar{f} = R_2(f) = (0\,1\,0\,0) = \alpha$ gera um subgrupo de ordem $n = 2^4 - 1 = 15$

em K^*. Pelo Teorema 1.20, f deverá gerar um grupo de ordem 15 . d em R^*, onde $d \geq 1$. Lembrando que as operações em R^* são sempre módulo $x^4 + x + 1$, temos que $x^4 = -x - 1 = 7x + 7$. Então, sendo $f = (0\ 1\ 0\ 0) = x$, vamos encontrar (realizando os cálculos) que $f^{60} = (1\ 0\ 0\ 0)$. Portanto, neste caso, $d = 4$ e f gera um grupo de ordem $15.4 = 60$ em R^* e disto segue que $f^4 = x^4 = 7x + 7 = (7\ 7\ 0\ 0)$ gera um grupo de ordem 15 (em R^*). Com isto, temos que $\beta = 7x + 7$ é um elemento primitivo de G_{15}.

Agora já estamos em condições de especificar um código BCH de comprimento $n = 15$ sobre Z_8. Vamos escolher o polinômio gerador g(x) tal que ele tenha como raízes β, β^2, β^3, β^4, i.e., g(x) será o mmc dos polinômios minimais de cada um desses elementos. Para calculá-los, vamos lembrar que o polinômio mínimo de β^i tem como raízes todos os elementos distintos na seqüência $\beta^i, (\beta^i)^2, (\beta^i)^4, (\beta^i)^8$.

Portanto, o polinômio minimal de β é dado por:

$$M_1(x) = (x-\beta)(x-\beta^2)(x-\beta^4)(x-\beta^8) = x^4 + 4x^3 + 6x^2 + 3x + 1$$

Este é o mesmo polinômio minimal de β^2 e β^4, i.e., $M_1(x) = M_2(x) = M_4(x)$. Finalmente o polinômio minimal de β^3 é dado por:

$$M_3(x) = (x-\beta^3)(x-\beta^6)(x-\beta^9)(x-\beta^{12}) = x^4 + x^3 + x^2 + x + 1$$

E assim g(x) = mmc $(M_1(x), M_3(x)) = x^8 + 5x^7 + 3x^6 + 6x^5 + 7x^4 + 6x^3 + 2x^2 + 4x + 1$ gera um código BCH de comprimento 15 e $d_{min} \geq 5$ sobre Z_8 (lembre que pelo Teorema 1.21, a distância mínima é maior do que o número máximo de inteiros consecutivos em {1, 2, 3, 4}, o conjunto das potências de β escolhidas para serem raízes de g(x).

- **<u>Propriedades Adicionais dos Anéis</u> GR(p^k, r)**

O anel $R = GR(p^k, r)$ é um ***anel local*** [13], i.e., os seus elementos divisores de zero formam um grupo abeliano aditivo e consistem daqueles polinômios de grau r - 1 (ou menos) cujos coeficientes são todos divisores de zero em $Z_p{}^k$. Um polinômio $p(x) \in R$ que possui pelo menos um coeficiente inversível em $Z_p{}^k$ não é um divisor de zero em R e, portanto, pertence a R^*, ou seja, é possível se encontrar um polinômio $q(x) \in R$, tal que $p(x).q(x) = 1$. A partir disto, podemos enunciar o Teorema 1.22.

Teorema 1.22 : Seja β um elemento primitivo de G_n (o subgrupo cíclico de R^* o qual contém todas as raízes de $x^n - 1$), onde $n = p^r - 1$. Então, o elemento $\delta = \beta^{\ell_1} - \beta^{\ell_2}$ é inversível em R se $0 \leq \ell_1, \ell_2 \leq n-1$ e $\ell_1 \neq \ell_2$.

Prova : Defina o mapeamento $\mu : Z_p{}^k[x] \mapsto Z_p[x]$ através de:

$$f(x) = \sum_{i=1}^{m} f_i.x^i \xrightarrow{\mu} \overline{f}(x) = \sum_{i=1}^{m} \overline{f}_i.x^i$$

onde $\overline{f}_i = R_p(f_i)$. Este mapeamento induz um homomorfismo de anel de R em K. Suponha agora, por absurdo, que δ seja um divisor de zero em R, i.e.,

$$\delta = p\delta_0 + p\delta_1.x + \ldots + p\delta_{r-1}.x^{r-1},$$

onde $\delta_i \in Z_p{}^k$ para $0 \le i \le r\text{-}1$. Com isto, temos que $\mu(\delta) = 0$. Seja $\theta = \mu(\beta)$; θ é então um elemento de K cujo polinômio mínimo é de grau r (pelo Corolário 1.3) e é um gerador de K^*. Agora como

$$\mu(\delta) = \mu(\beta^{\ell_1}) - \mu(\beta^{\ell_2}),$$

então,

$$0 = (\mu(\beta))^{\ell_1} - (\mu(\beta))^{\ell_2} = \theta^{\ell_1} - \theta^{\ell_2}.$$

Mas isto implica que $\theta^{\ell_1} = \theta^{\ell_2}$ o que é impossível quando $\ell_1 \neq \ell_2$ e $0 \le \ell_1, \ell_2 \le n\text{-}1$. Portanto, $\delta = \beta^{\ell_1} - \beta^{\ell_2}$ é de fato inversível em R, i.e., $\delta \in R^*$. ■

1.5.1. Matriz verificação de paridade

Iremos agora especificar a matriz verificação de paridade destes códigos BCH sobre anéis de inteiros residuais.

Sejam as seguintes raízes consecutivas de g(x) : β, β^2, β^3, ..., β^{2t}. Portanto, se v(x) for uma palavra-código, então, v(x) = c(x) . g(x), o que implica que β, β^2, ..., β^{2t} são raízes de v(x). Por outro lado, se β, β^2, ..., β^{2t} são raízes de v(x), então v(x) é divisível por cada um dos polinômios minimais de β, β^2, ..., β^{2t} e conseqüentemente pelo seu mmc, g(x). Portanto, v(x) é palavra-código. Em suma, v(x) é palavra-código se, e somente se, β, β^2, ..., β^{2t} são raízes de v(x).

Baseados neste fato, podemos construir a matriz verificação de paridade H da seguinte forma:

$$H = \begin{bmatrix} 1 & \beta & \beta^2 & \cdots & \beta^{n-1} \\ 1 & (\beta^2) & (\beta^2)^2 & \cdots & (\beta^2)^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & (\beta^{2t}) & (\beta^{2t})^2 & \cdots & (\beta^{2t})^{n-1} \end{bmatrix} \tag{1.2}$$

E, portanto, v(x) é palavra-código se, e somente se, $v \cdot H^t = 0$.

1.5.2. Isomorfismo entre o anel de Galois e um anel de matrizes

Um dos passos do algoritmo que iremos introduzir para a decodificação dos códigos BCH desta Seção requer o cálculo do quociente entre dois polinômios de R, onde o polinômio divisor pertence a R^*. Se o coeficiente dominante do polinômio divisor for inversível em Z_p^k, podemos então aplicar o algoritmo da divisão longa para obtermos o quociente. Entretanto, este último algoritmo não mais se aplica se caso o coeficiente dominante do polinômio divisor for um divisor de zero em Z_p^k.

Nestas circunstâncias, iremos usar o fato de que associada a cada elemento de R existe uma matriz quadrada r x r cujos elementos pertencem a Z_p^k. As matrizes correspondentes aos elementos de R^* são aquelas cujos determinantes são inversíveis em Z_p^k e, portanto, não singulares. Assim trataremos o problema da inversão de polinômios em R^* como um problema de inversão de matrizes.

Vamos agora introduzir uma notação apropriada. Sejam os polinômios

$$f(x) = f_0 + f_1x + \ldots + f_{r-1}x^{r-1}$$

e

$$g(x) = g_0 + g_1x + \ldots + g_{r-1}x^{r-1}$$

pertencentes a R. O produto f(x).g(x) é dado por:

$$f(x).g(x) = f_0.g(x) + f_1.xg(x) + \ldots + f_{r-1}.x^{r-1}g(x).$$

O coeficiente de x^i ($0 \leq i \leq r-1$) no polinômio $p(x) \in R$ será denotado por $c_i(p(x))$. Portanto, se h(x) = f(x).g(x), então:

$$c_i(h(x)) = \sum_{j=0}^{r-1} c_j(f(x)).c_i(x^j g(x)) \tag{1.3}$$

para $0 \leq i \leq r\text{-}1$.

Agora, a cada elemento $f(x) = f_0 + f_1x + \ldots + f_{r-1}x^{r-1}$ pertencente a R, associaremos uma matriz $F_{r\,x\,r}$ dada por:

$$F = \begin{bmatrix} f \\ xf \\ x^2f \\ \vdots \\ x^{r-1}f \end{bmatrix} = \begin{bmatrix} f_0 & f_1 & f_2 & \cdots & f_{r-1} \\ c_0(xf) & c_1(xf) & c_2(xf) & \cdots & c_{r-1}(xf) \\ c_0(x^2f) & c_1(x^2f) & c_2(x^2f) & \cdots & c_{r-1}(x^2f) \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ c_0(x^{r-1}f) & c_1(x^{r-1}f) & c_2(x^{r-1}f) & \cdots & c_{r-1}(x^{r-1}f) \end{bmatrix} \tag{1.4}$$

Cada linha de F é formada pelos coeficientes do polinômio $x^if(x)$, para $0 \leq i \leq r\text{-}1$. O Teorema a seguir caracteriza completamente as matrizes F que acabamos de construir.

Teorema 1.23 : O conjunto das matrizes F definidas acima forma um anel S o qual é isomorfo a R. As operações em S são as operações usuais de soma e multiplicação de matrizes e as operações entre os seus elementos são realizadas módulo p^k.

Prova : Sejam F e G quaisquer duas matrizes em S e f e g os respectivos elementos em R associados. Assim,

$$F = \begin{bmatrix} f \\ xf \\ \vdots \\ x^{r-1}f \end{bmatrix} = \begin{bmatrix} f_0 & f_1 & \cdots & f_{r-1} \\ c_0(xf) & c_1(xf) & \cdots & c_{r-1}(xf) \\ \vdots & \vdots & \cdots & \vdots \\ c_0(x^{r-1}f) & c_1(x^{r-1}f) & \cdots & c_{r-1}(x^{r-1}f) \end{bmatrix} \quad \text{e}$$

$$G = \begin{bmatrix} g \\ xg \\ \vdots \\ x^{r-1}g \end{bmatrix} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{r-1} \\ c_0(xg) & c_1(xg) & \cdots & c_{r-1}(xg) \\ \vdots & \vdots & \cdots & \vdots \\ c_0(x^{r-1}g) & c_1(x^{r-1}g) & \cdots & c_{r-1}(x^{r-1}g) \end{bmatrix}$$

A soma de F e G é a matriz

$$F+G=\begin{bmatrix}(f+g)\\ x(f+g)\\ \vdots\\ x^{r-1}(f+g)\end{bmatrix}$$

a qual é uma matriz em S, já que $f+g$ pertence a R. O produto de F por G é a matriz

$$F.G=\begin{bmatrix}\sum_{i=0}^{r-1}f_ic_0(x^ig) & \sum_{i=0}^{r-1}f_ic_1(x^ig) & \cdots & \sum_{i=0}^{r-1}f_ic_{r-1}(x^ig)\\ \sum_{i=0}^{r-1}c_i(xf).c_0(x^ig) & \sum_{i=0}^{r-1}c_i(xf).c_1(x^ig) & \cdots & \sum_{i=0}^{r-1}c_i(xf).c_{r-1}(x^ig)\\ \vdots & \vdots & \cdots & \vdots\\ \sum_{i=0}^{r-1}c_i(x^{r-1}f)c_0(x^ig) & \sum_{i=0}^{r-1}c_i(x^{r-1}f)c_1(x^ig) & \cdots & \sum_{i=0}^{r-1}c_i(x^{r-1}f)c_{r-1}(x^ig)\end{bmatrix}=$$

$$=\begin{bmatrix}(f.g)\\ x(f.g)\\ \vdots\\ x^{r-1}(f.g)\end{bmatrix}$$

o qual está também em S, já que $f.g$ pertence a R. Podemos ainda verificar que S é um grupo abeliano com relação à adição de matrizes e que as propriedades associativa, distributiva e elemento neutro com relação à multiplicação estão satisfeitas. Portanto, S é de fato um anel. Resta-nos apenas mostrar que R e S são isomorfos. Para isto defina a aplicação $\phi : R \mapsto S$ através de:

$$\phi(f)=\begin{bmatrix}f\\ xf\\ \vdots\\ x^{r-1}f\end{bmatrix}$$

É fácil ver que ϕ é uma bijeção e que

$$\phi(f+g)=\phi(f)+\phi(g) \quad \text{e}$$

$$\phi(f.g)=\phi(f).\phi(g)$$

E, portanto, os anéis R e S são isomorfos. ■

Assim, com base no Teorema 1.23, sempre que não for possível aplicar o algoritmo da divisão longa para invertermos um elemento f de R^*, associaremos a ele uma matriz $\mathbb{F}$ como em (1.4). A seguir tomamos a r-upla que forma a primeira linha da matriz $\mathbb{F}^{-1}$; esta r-upla é o elemento f^{-1}, o inverso de f em R^*.

A inversão de uma matriz $\mathbb{F}$ pode ser implementada formando-se a matriz aumentada $[\mathbb{F} \mid I_r]$ (onde I_r é a matriz identidade r x r) e transformando-a a seguir em uma matriz da forma $[I_r \mid G]$ através de operações elementares com as suas linhas (isto é sempre possível desde que $\mathbb{F}$ seja de fato uma matriz inversível). Sabemos que G é a própria matriz $\mathbb{F}^{-1}$ e que o tempo computacional requerido para esta transformação é $O(r^3)$. Isto está dentro de limites aceitáveis, já que, em aplicações práticas, os valores de r utilizados para a construção dos códigos BCH desta Seção geralmente estarão no intervalo $2 \leq r \leq 10$.

Vale a pena ressaltar ainda que para invertermos elementos em R^* é interessante recorrermos ao cálculo de matrizes inversas somente quando não for possível se aplicar o algoritmo da divisão longa, já que este último processo é muito mais rápido que o primeiro.

1.6. Conclusões

Neste capítulo estudamos as construções das principais classes de códigos corretores de erros definidos sobre anéis de inteiros residuais. Foram caracterizados códigos cíclicos (em termos de matrizes geradoras), códigos de Hamming (correção de um erro aleatório) e códigos BCH (para múltipla correção de erros). Foram focalizados códigos sobre anéis da forma Z_q onde q é uma potência de primo, já que códigos lineares sobre tais anéis possuem propriedades de distância tão boas quanto ou melhores que os códigos lineares sobre anéis da forma Z_m onde m é um produto de duas ou mais potências de primos.

No Capítulo 4 iremos concatenar códigos sobre anéis para construirmos códigos sobre grupos abelianos. As propriedades de taxa e distância destes últimos serão herdadas dos códigos apresentados neste capítulo.

CAPÍTULO 2

DECODIFICAÇÃO DOS CÓDIGOS SOBRE Zq

Neste capítulo serão descritos os métodos para a decodificação dos códigos sobre anéis de inteiros residuais, apresentados no Capítulo 1. Na Seção 2.1 descreveremos um método para a decodificação dos códigos de Hamming sobre anéis (Seção 1.3), o qual é uma extensão do simples método para a decodificação destes códigos, quando definidos sobre corpos GF(q). Na Seção 2.2 será apresentado um método para a decodificação dos códigos Reed-Solomon (Seção 1.4) e BCH (Seção 1.5). Quando estes códigos estão definidos sobre corpos GF(q), o primeiro passo da decodificação é localizar a posição dos erros na palavra recebida, através do uso do algoritmo de Berlekamp-Massey (BM). Depois disto, aplica-se um procedimento desenvolvido por Forney para a determinação da magnitude dos erros. Mostraremos que o algoritmo de Berlekamp-Massey pode ser adaptado para a decodificação dos códigos Reed-Solomon e BCH quando definidos sobre anéis e que o procedimento dado por Forney continua válido, sem alterações. Esta adaptação não altera os fundamentos do algoritmo e, portanto, a complexidade permanece praticamente inalterada.

Na Seção 2.3 mostraremos que o algoritmo de BM modificado ainda poderá ser aplicado para geração de seqüências, através de registradores de deslocamento cujos elementos pertencem a um anel comutativo (com identidade). Esta seção apenas complementa a teoria acerca deste algoritmo, mostrando mais uma aplicação do mesmo.

2.1. Códigos de Hamming

Na Seção 1.3 apresentamos a matriz verificação de paridade de um código de Hamming sobre Z_q, onde q é uma potência de primo. Suponha que as palavras-código tenham comprimento n, que o número de dígitos de informação seja k e que a matriz verificação de paridade seja H. Suponha ainda que o vetor enviado seja $\underline{v}$, o vetor erro seja $\underline{e}$ e o vetor recebido seja $\underline{r}$ ($\underline{r} = \underline{v} + \underline{e}$, onde + representa a adição módulo q). No caso da ocorrência de um único erro, o vetor síndrome é suficiente para detectar a posição (na palavra-código) e magnitude. Isto é facilmente visto através da equação que define a síndrome, $\underline{s} = \underline{r}.H^t$.

Este cálculo é equivalente a $\underline{s} = (\underline{v} + \underline{e}).H^t = \underline{v}.H^t + \underline{e}.H^t = \underline{e}.H^t$, já que $\underline{v}.H^t = \underline{0}$ (pois $\underline{v}$ é uma palavra-código). Repare então que o vetor síndrome, no caso da ocorrência de um único erro numa dada posição j, será dado pela j-ésima coluna da matriz H multiplicada por e_j. Então devemos procurar na matriz H uma coluna tal que quando multiplicada por um dado elemento de $Z_q^* = \{1,2,\ldots,q-1\}$ reproduza o vetor síndrome. A coluna encontrada representará a posição j na palavra-código em que ocorreu o erro e o elemento de Z_q^*, a magnitude do erro.

Exemplo 2.1 : Seja o código do Exemplo 1.7 (Seção 1.3.1). Suponha então que a palavra transmitida seja $\underline{v}$ = (0 0 0 0 0 0 0) (a palavra toda nula) e que o vetor erro introduzido pelo canal seja $\underline{e}$ = (0 0 0 0 0 2 0). Quando multiplicamos o vetor recebido $\underline{r} = \underline{v} + \underline{e}$ por H, encontramos que $\underline{s}$ = (2 2 0). Podemos perceber que este vetor representa a sexta coluna de H multiplicada por 2. Com isto, podemos dizer que ocorreu um erro de magnitude 2 na posição 6 da palavra-código.

Obs. :

1) No caso da ocorrência de um único erro, este método apontará univocamente a posição e a magnitude do erro (isto é devido ao fato de que duas colunas de H são sempre linearmente independentes). Em outras palavras, não existirá a possibilidade de se determinar dois ou mais padrões de erro para uma dada síndrome;

2) No caso da ocorrência de 2 ou mais erros (i.e., quando se está fora da capacidade de correção do código), pode acontecer que não se encontre uma coluna de H tal

que quando multiplicada por um elemento de Z_q^* , reproduza o vetor síndrome. Neste caso, o decodificador declara apagamento por falta de dados para a correta decodificação. Caso se encontre essa tal coluna, ocorrerá um ***erro de decodificação***.

Com o material da Seção 1.3 e o desta, concluímos a geração e a decodificação de códigos de Hamming definidos sobre anéis de inteiros residuais.

2.2. Códigos Reed-Solomon e BCH

Nas Seções 1.4 e 1.5 foram caracterizados os códigos Reed-Solomon (RS) e BCH respectivamente, através das matrizes geradora e verificação de paridade. Nesta Seção iremos discutir um método de decodificação de tais códigos. Observando as Equações (1.1) e (1.2), podemos constatar que as matrizes verificação de paridade para estes códigos, quando projetados para a correção de até t erros, são similares e possuem a forma

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \cdots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \cdots & (\alpha^{2t})^{n-1} \end{bmatrix} \tag{2.1}$$

onde n é o comprimento da palavra-código em questão. Recorde que no caso de códigos RS sobre Z_{p^k}, α é um elemento primitivo de Zp; no caso de códigos BCH (também sobre Z_{p^k}), α é uma raiz de $x^n - 1$ e pertence ao anel $GR(p^k, r)$ (a extensão de dimensão r de Z_{p^k}). Note que t deverá ser escolhido de tal forma que os elementos α, α^2, ..., α^{2t} sejam todos distintos.

Portanto, devido a esta similaridade, o procedimento de correção de erros que iremos descrever (que se baseia na informação dada pelo vetor síndrome) servirá para a decodificação de ambos os códigos. Ele sempre será capaz de corrigir qualquer combinação de t ou menos erros.

Suponha então que a palavra-código transmitida seja $\underline{v} = (v_0\ v_1 \ldots v_{n-1})$ e que o padrão de erro introduzido pelo canal seja $\underline{e} = (e_0\ e_1 \ldots e_{n-1})$. Portanto, o vetor recebido pelo decodificador será $\underline{r} = (r_0\ r_1 \ldots r_{n-1})$. Estes vetores também podem ser apresentados na forma polinomial por $v(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$, $e(X) = e_0 + e_1X + \ldots + e_{n-1}X^{n-1}$ e $r(X) = r_0 + r_1X + \ldots + r_{n-1}X^{n-1}$, respectivamente.

Vamos assumir agora que a *i*-ésima componente não nula de $\underline{e}$ ($1 \le i \le \nu \le t$) ocorra na posição *j*, onde *j* pode ser qualquer inteiro entre 0 e *n*-1 (inclusive). Então, associaremos a esta *i*-ésima componente não nula um par ordenado (X_i, Y_i) tal que:

X_i : é um número de localização de erro dado por α^j

e

Y_i : é a magnitude do erro ocorrido na posição *j*.

O vetor síndrome é dado por $\underline{s} = \underline{r}.H^t = (\underline{v} + \underline{e}).H^t = \underline{e}.H^t$, onde H é a matriz verificação de paridade dada por (2.1). Então, em termos dos pares (X_i, Y_i), as componentes s_j de $\underline{s}$ são dadas por

$$s_j = r(\alpha^j) = e(\alpha^j) = \sum_{i=1}^{\nu} Y_i X_i^j \qquad (2.2)$$

onde $1 \le j \le 2t$ e ν representa o número de erros ocorridos.

Então, um método para se corrigir erros é resolver o Sistema de Equações (2.2) o qual produzirá como resposta os pares (X_i, Y_i) que representam as posições e magnitudes dos mesmos.

Iremos atacar primeiramente o problema da localização dos erros para depois resolver o problema da determinação das magnitudes dos mesmos. Recorde que no caso dos códigos serem binários, a localização dos erros já implica necessariamente na determinação das magnitudes dos mesmos.

Sejam os valores $\sigma_1, \sigma_2, \ldots, \sigma_\nu$ definidos através da equação

$$(X - X_1)(X - X_2) \ldots (X - X_\nu) = X^\nu + \sigma_1 X^{\nu-1} + \ldots + \sigma_{\nu-1}X + \sigma_\nu \qquad (2.3)$$

Esses valores são conhecidos como as ***funções simétricas elementares*** dos X_i. Multiplicando ambos os membros da Equação (2.3) por $Y_i X_i^j$ e depois substituindo X_i

($1 \leq i \leq \nu$) em X, o seguinte conjunto de equações resulta:

$$Y_i X_i^{j+\nu} + Y_i X_i^{j+\nu-1}\sigma_1 + \ldots + Y_i X_i^{j+1}\sigma_{\nu-1} + Y_i X_i^{j}\sigma_\nu = 0 \qquad (2.4)$$

Somando-as para $1 \leq i \leq \nu$ e usando as Equações (2.2) vem que

$$s_{j+\nu} + s_{j+\nu-1}.\sigma_1 + \ldots + s_{j+1}.\sigma_{\nu-1} + s_j.\sigma_\nu = 0 \qquad (2.5)$$

e todos os s_j são conhecidos se $1 \leq j \leq 2t - \nu$. Portanto, o cálculo dos σ_i's a partir do vetor síndrome é feito resolvendo-se o Sistema Linear (2.5) de modo que ν tenha o menor valor possível (isto é requerido pois sempre estaremos assumindo que o vetor erro que ocorre é aquele que possui o menor peso de Hamming possível). Por construção, sabemos que sempre existe uma solução para o Sistema (2.5). Pode-se provar que ela é única se, e somente se, todas as magnitudes Y_i ($1 \leq i \leq \nu$) forem elementos inversíveis no anel sobre o qual o código está definido (ver Apêndice 2.1).

Com isto, acabamos de mostrar que o procedimento de decodificação dos códigos RS e BCH compreende os seguintes passos:

- *Passo 1* : Cálculo do vetor síndrome $\underline{s} = (s_1\ s_2\ \ldots\ s_{2t})$ a partir do vetor recebido $\underline{r}$;
- *Passo 2* : Cálculo das funções simétricas elementares $\sigma_1, \sigma_2, \ldots, \sigma_\nu$, a partir de $\underline{s}$;
- *Passo 3* : Cálculo dos números de localização de erro $X_1, X_2, \ldots, X_\nu$ a partir dos σ_i's ($1 \leq i \leq \nu$);
- *Passo 4* : Cálculo das magnitudes dos erros Y_i a partir dos X_i e $\underline{s}$.

A seguir, passaremos a caracterizar cada um destes passos.

Passo 1* - *Cálculo do Vetor Síndrome

$\underline{s} = \underline{r}.H^t$

Passo 2* - *Cálculo das Funções Simétricas Elementares [14]

O problema que precisamos resolver neste estágio é o seguinte: dada uma seqüência de elementos $s_1, s_2, \ldots, s_{2t}$ (as componentes do vetor síndrome)

pertencentes a um anel comutativo R (lembre que se estivermos decodificando o código RS (Seção 1.4), R é o anel Z_{p^k} ($k \geq 1$), enquanto que se estivermos decodificando o código BCH (Seção 1.5), R é o anel GR(p^k, r) ($k \geq 1$ e $r \geq 1$), determine a solução do Sistema Linear (2.5) nas incógnitas σ_i ($1 \leq i \leq \nu$) tal que ν seja mínimo (já vimos que a solução só será única no caso em que as magnitudes de todos os erros forem inversíveis em R).

A solução do Sistema Linear (2.5) requer a generalização do algoritmo de Berlekamp-Massey (BM), descrito em [15], devido ao fato de que as suas entradas pertencem a um anel comutativo R, ao invés de um corpo F. Este algoritmo é iterativo, no sentido de que no n-ésimo passo o decodificador procura determinar um conjunto de ℓ_n valores $\sigma_i^{(n)}$ tal que as n - ℓ_n equações (conhecidas como ***somas de potência***)

$$\begin{aligned} &s_n.\sigma_0^{(n)} + s_{n-1}.\sigma_1^{(n)} + \cdots + s_{n-\ell_n}.\sigma_{\ell_n}^{(n)} = 0 \\ &s_{n-1}.\sigma_0^{(n)} + s_{n-2}.\sigma_1^{(n)} + \cdots + s_{n-\ell_n-1}.\sigma_{\ell_n}^{(n)} = 0 \\ &\qquad\qquad \cdots \\ &s_{\ell_n+1}.\sigma_0^{(n)} + s_{\ell_n}.\sigma_1^{(n)} + \cdots + s_1.\sigma_{\ell_n}^{(n)} = 0 \end{aligned} \qquad (2.6)$$

sejam satisfeitas com o menor ℓ_n possível e onde $\sigma_0^{(n)} = 1$. Este conjunto de valores $\sigma_i^{(n)}$ é comumente escrito na forma polinomial

$$\sigma^{(n)}(X) = \sigma_0^{(n)} + \sigma_1^{(n)}.X + \sigma_2^{(n)}.X^2 + \cdots + \sigma_{\ell_n}^{(n)}.X^{\ell_n}.$$

Note que esta forma representa a solução no n-ésimo estágio. A n-ésima ***discrepância***, será denotada por d_n e é definida por

$$d_n = s_{n+1}.\sigma_0^{(n)} + s_n.\sigma_1^{(n)} + \ldots + s_{n+1-\ell_n}\,\sigma_{\ell_n}^{(n)}$$

A seguir, apresentaremos dois lemas que serão importantes na generalização de parte do algoritmo sendo proposto. Estes lemas estão diretamente relacionados com a determinação de $\sigma^{(n+1)}(X)$ (não necessariamente com o menor valor de ℓ_{n+1} possível) a partir de $\sigma^{(n)}(X)$.

Lema 2.1 (Generalização do Lema 9.3 [15]) : Suponha que $\sigma^{(n)}(X)$ seja um polinômio solução minimal para as n primeiras somas de potência (i.e., tal que as

Equações (2.6) sejam satisfeitas com ℓ_n mínimo) e tenha a próxima discrepância $d_n \neq 0$. Seja

$$\sigma^{(m)}(X) = 1 + \sigma_1^{(m)}.X + \sigma_2^{(m)}.X^2 + \ldots + \sigma_{\ell_m}^{(m)}.X^{\ell_m}$$

qualquer polinômio solução para as m primeiras somas de potência, com $1 \leq m < n$ e tal que a equação $d_n - y.d_m = 0$ admita uma solução em y sobre o anel R (note que mesmo sendo d_m um divisor de zero em R, pode ocorrer que esta equação ainda admita uma solução). Então o polinômio

$$\sigma^{(n)}(X) - y.X^{n-m}.\sigma^{(m)}(X) = \sigma^{(n+1)}(X)$$

é uma solução para as $n+1$ primeiras somas de potência. Mais ainda, $\ell_{n+1} = \max[\ell_n, \ell_m + n - m]$.

Prova : Como $\sigma^{(n)}(X)$ é uma solução para as n primeiras somas de potência, então as Equações (2.6) estão verificadas. Isto é,

$$\begin{aligned} \sum_{i=0}^{\ell_n} s_{j-i}.\sigma_i^{(n)} &= 0 \quad ; \quad \ell_n + 1 \leq j \leq n \\ &= d_n \quad ; \quad j = n+1 \end{aligned} \tag{2.7}$$

Similarmente, como $\sigma^{(m)}(X)$ é uma solução para as m primeiras somas de potência, então:

$$\begin{aligned} \sum_{i=0}^{\ell_m} s_{j-i}.\sigma_i^{(m)} &= 0 \quad ; \quad \ell_m + 1 \leq j \leq m < n \\ &= d_m \neq 0 \quad ; \quad j = m+1. \end{aligned} \tag{2.8}$$

Se $\sigma^{(n+1)}(X) = \sigma^{(n)}(X) - y.X^{n-m}.\sigma^{(m)}(X)$ for de fato uma solução para as $n+1$ primeiras somas de potência, então deve-se ter que:

$$\sum_{i=0}^{\ell_{n+1}} s_{j-i}.\sigma_i^{(n+1)} = 0 \quad ; \quad \ell_{n+1} + 1 \leq j \leq n+1 \tag{2.9}$$

Esta soma tem a forma

$$\sum_{i=0}^{\ell_{n+1}} s_{j-i}.\left(\sigma_i^{(n)} - y.\sigma_{i-(n-m)}^{(m)}\right) \tag{2.10}$$

Como $\sigma_i^{(n)} = 0$ para $i < 0$ e $i > \ell_n$ e também $\sigma_i^{(m)} = 0$ para $i < 0$ e $i > \ell_m$, então a Soma (2.10) pode ser escrita como

$$\sum_{i=0}^{\ell_n} s_{j-i}.\sigma_i^{(n)} - y.\sum_{i=n-m}^{\ell_m+n-m} s_{j-i}.\sigma_{i-(n-m)}^{(m)} \qquad (2.11)$$

(ou alternativamente, como $\sum_{i=0}^{\ell_n} s_{j-i}.\sigma_i^{(n)} - y.\sum_{i=0}^{\ell_m} s_{j-i-(n-m)}.\sigma_i^{(m)}$).

Note que para $j = n + 1$, a primeira Soma em (2.11) é igual a d_n e a segunda é igual a d_m. A Equação (2.11) então reduz-se a $d_n - y.d_m = 0$ e, portanto, é verificada. Pela Equação (2.7) a primeira soma na Equação (2.11) é igual a zero desde que $\ell_n + 1 \le j \le n$. Pela Equação (2.8) a segunda soma na Equação (2.11) é igual a zero desde que $\ell_m + 1 \le j - (n - m) \le m$ ou equivalentemente, desde que $n - m + \ell_m + 1 \le j \le n$. Em resumo, a Equação (2.11) é satisfeita desde que

$$\max(\ell_n, \ell_m + n - m) + 1 \le j \le n + 1.$$

Como $n + 1 - \max(\ell_n, \ell_m + n - m)$ Equações (2.6) são satisfeitas por $\sigma^{(n+1)}(X)$, então o seu grau é dado formalmente por $\ell_{n+1} = \max(\ell_n, \ell_m + n - m)$. Finalmente, note que os coeficientes das potências mais altas da indeterminada X em $\sigma^{(n+1)}(X)$ podem ser nulos, implicando que certas Equações (2.6) adicionais ainda podem ser satisfeitas, i.e., $\sigma^{(n+1)}(X)$ pode não ser minimal. ■

Lema 2.2 (Generalização do Lema 9.4 [15]) : Sejam $\sigma^{(n)}(X)$, ℓ_n e $d_n \neq 0$ como definidos no Lema 2.1. Suponha que $\sigma^{(n+1)}(X)$ seja um polinômio solução das Equações (2.6) satisfazendo $n + 1 - \ell_{n+1}$ equações e que

$$\sigma^{(n+1)}(X) - \sigma^{(n)}(X) = aX^{n-m}\,\sigma^{(m)}(X),$$

onde a é inversível em R e $\sigma_0^{(m)} = 1$. Então, o polinômio $\sigma^{(m)}(X)$ é um polinômio solução para as $m - \ell_m$ primeiras Equações (2.6), tendo próxima discrepância d_m satisfazendo $d_n + ad_m = 0$ e $\ell_m = \ell_{n+1} - (n - m)$.

Prova : Por hipótese

$$\sum_{i=0}^{\ell_{n+1}} s_{j-i}.\sigma_i^{(n+1)} = 0 \quad ; \quad \ell_{n+1} + 1 \le j \le n + 1. \qquad (2.12)$$

e

$$\sum_{i=0}^{\ell_n} s_{j-i}.\sigma_i^{(n)} = 0 \quad ; \quad \ell_n + 1 \le j \le n$$
$$= d_n \neq 0 \; ; \; j = n + 1 \qquad (2.13)$$

Como $\sigma^{(n)}(X)$ é uma solução minimal, $\ell_{n+1} \geq \ell_n$. Subtraindo as Equações (2.13) das Equações (2.12) para $\ell_{n+1} + 1 \leq j \leq n + 1$ obtém-se que

$$\begin{aligned} \sum_{i=0}^{\ell_{n+1}} s_{j-i}.(\sigma_i^{(n+1)} - \sigma_i^{(n)}) &= 0 \quad ; \quad \ell_{n+1}+1 \leq j \leq n \\ &= -d_n \; ; \; j = n+1 \end{aligned} \tag{2.14}$$

Suponha agora que os n - m primeiros coeficientes de s_{j-i} sejam nulos (note que como $\sigma_0^{(n+1)} = \sigma_0^{(n)} = 1$, então n - m > 0, i.e., $n > m$). Então, a Equação (2.14) reduz-se a

$$\begin{aligned} \sum_{i=n-m}^{\ell_{n+1}} s_{j-i}.(\sigma_i^{(n+1)} - \sigma_i^{(n)}) &= 0 \quad ; \quad \ell_{n+1}+1 \leq j \leq n \\ &= -d_n \; ; \; j = n+1 \end{aligned} \tag{2.15}$$

Fazendo $\ell_m = \ell_{n+1}$ - (n - m), a Equação (2.15) pode ser escrita como:

$$\begin{aligned} \sum_{i=0}^{\ell_m} s_{j-i}.(\sigma_{i+n-m}^{(n+1)} - \sigma_{i+n-m}^{(n)}) &= 0 \quad ; \quad \ell_m+1 \leq j \leq m \\ &= -d_n \; ; \; j = m+1 \end{aligned} \tag{2.16}$$

Finalmente, defina o polinômio $\sigma^{(m)}(X)$ por $\sigma_i^{(m)} = (\sigma_{i+n-m}^{(n+1)} - \sigma_{i+n-m}^{(n)}).a^{-1}$, para $0 \leq i \leq \ell_m$. Portanto,

$$\begin{aligned} \sum_{i=0}^{\ell_n} s_{j-i}.\sigma_i^{(m)} &= 0 \quad ; \quad \ell_m+1 \leq j \leq m \\ &= -d_n.a^{-1} = d_m \; ; \; j = m+1 \end{aligned} \tag{2.17}$$

Então, pelas Equações (2.17), $\sigma^{(m)}(X)$ é uma solução para as primeiras m - ℓ_m Equações (2.6) e tem próxima discrepância d_m tal que $d_n + ad_m = 0$. O grau de $\sigma^{(m)}(X)$ é dado formalmente por $\ell_m = \ell_{n+1}$ - (n - m). Note que os coeficientes das potências mais altas da indeterminada X em $\sigma^{(m)}(X)$ podem ser nulos, implicando que certas Equações (2.6) adicionais podem ser satisfeitas, i.e., $\sigma^{(m)}(X)$ pode não ser minimal. ■

Baseados nestes dois lemas, iremos agora mostrar o seguinte:

Teorema 2.1 (Generalização do Teorema 9.10 [15]) : Seja $\sigma^{(n)}(X)$ uma solução minimal no n-ésimo estágio e $\sigma^{(m)}(X)$ uma das soluções minimais anteriores,

$1 \leq m < n$, tal que a equação $d_n - y.d_m = 0$ admita uma solução em y e $m - \ell_m$ tenha o máximo valor. Então uma solução no estágio $n + 1$ é $\sigma^{(n+1)}(X)$, onde:

- se $d_n = 0$, então

$$\sigma^{(n+1)}(X) = \sigma^{(n)}(X) \quad \text{e} \quad \ell_{n+1} = \ell_n; \tag{2.18}$$

- se $d_n \neq 0$, então

$$\sigma^{(n+1)}(X) = \sigma^{(n)}(X) - y.X^{n-m}.\sigma^{(m)}(X) \quad \text{e} \quad \ell_{n+1} = \max[\ell_n, \ell_m + n - m] \tag{2.19}$$

Vamos mostrar que em muitos casos, esta solução $\sigma^{(n+1)}(X)$ já é uma solução minimal.

Prova : Se $d_n = 0$, então claramente $\sigma^{(n+1)}(X) = \sigma^{(n)}(X)$ é uma solução minimal já que $\sigma^{(n)}(X)$ o é. Agora considere o caso onde $d_n \neq 0$. Como $\sigma^{(m)}(X)$ e $\sigma^{(n)}(X)$ são conhecidos, então $\sigma^{(n+1)}(X)$ também o é através da Equação (2.19). Pelo Lema 2.1, $\sigma^{(n+1)}(X)$ é um polinômio solução de grau dado por $\ell_{n+1} = \max[\ell_n, \ell_m + n - m]$. Resta mostrar que esta é uma solução minimal:

- se $m - \ell_m \geq n - \ell_n$, então $\ell_{n+1} = \ell_n$ pelo Lema 2.1 e daí $\sigma^{(n+1)}(X)$ será uma solução minimal no estágio $n + 1$;

- Por outro lado, se $m - \ell_m < n - \ell_n$, então $\ell_{n+1} = \max[\ell_n, \ell_m + n - m] = \ell_m + n - m > \ell_n$. Vamos analisar quando $\sigma^{(n+1)}(X)$ ainda é uma solução minimal. Assuma, por absurdo, que exista um polinômio $D^{(n+1)}(X)$ com grau d tal que $\ell_n \leq d < \ell_m + n - m$ e que o coeficiente da menor potência da indeterminada X em $D^{(n+1)}(X) - \sigma^{(n)}(X)$ seja inversível em R. Considere agora dois casos:

  i) se $d = \ell_n$, então pelo Lema 2.2, existe uma solução $\sigma^{(m')}(X)$ tal que $\ell_{m'} = d - (n - m')$, i.e., tal que $m' - \ell_{m'} = n - \ell_n$. Como por hipótese $m - \ell_m < n - \ell_n$, então $m - \ell_m < m' - \ell_{m'}$. Mas $m - \ell_m$ foi escolhido como o maior dos valores $k - \ell_k$ das soluções prévias. Contradição;

  ii) se $d > \ell_n$, então pelo Lema 2.2, $d = \ell_{m'} + n - m'$. Mas como $m - \ell_m \geq m' - \ell_{m'}$, então $d = n - (m' - \ell_{m'}) \geq n - (m - \ell_m) = \ell_{n+1} > d$, i.e., $d > d$, o que é uma contradição.

  Portanto, quando o coeficiente da menor potência de X em $D^{(n+1)}(X) - \sigma^{(n)}(X)$ é uma unidade em R, então $\sigma^{(n)}(X)$ é uma solução minimal ■

Note que a solução $\sigma^{(n+1)}(X)$ apresentada pelo Teorema 2.1 não é necessariamente a resposta procurada devido ao fato de que o mesmo não garante a mini-

milidade quando no caso ii), o coeficiente da menor potência da indeterminada X em $D^{(n+1)}(X) - \sigma^{(n)}(X)$ não for inversível em R. Entretanto, em muitos casos este Teorema já aponta corretamente a solução minimal.

Antes de discutirmos o emprego da generalização do algoritmo de BM na determinação iterativa da solução minimal, vamos fixar as condições iniciais. Como em [15], defina:

$$\sigma^{(-1)}(X) = 1 , \qquad \ell_{-1} = 0 , \; d_{-1} = 1 ,$$

$$\sigma^{(0)}(X) = 1 , \qquad \ell_0 = 0 , \; d_0 = s_1 ,$$

onde 1 é a unidade do anel em consideração e s_1 é a primeira componente não nula do vetor síndrome $\underline{s}$.

Pelo Lema 1 [25], podemos constatar que se $\sigma^{(n)}(X)$ satisfaz $n - \ell_n$ Equações (2.6), mas não satisfaz $n + 1 - \ell_n$ Equações (2.6), então a solução $\sigma^{(n+1)}(X)$ deverá satisfazer $n + 1 - \ell_{n+1}$ Equações (2.6), onde:

$$\ell_{n+1} \geq \max[\ell_n, \; n + 1 - \ell_n].$$

Agora, usando os argumentos da Seção III de [25] é imediato mostrar que se a equação em y, $d_{n'} - y \, . \, d_{m'} = 0$, sempre admite uma solução para $1 \leq m' < n' \leq n$, então a desigualdade acima pode ser substituída por uma igualdade, i.e.,

$$\ell_{n+1} = \max[\ell_n, \; n + 1 - \ell_n] = \max[\ell_n, \; \ell_m + n - m].$$

Caso contrário, se existir n' tal que $d_{n'} - y \, . \, d_{m'} = 0$ não admita solução em y para nenhum m' tal que $1 \leq m' < n'$, então as soluções $\sigma^{(n)}(X)$, para $n \geq n'$, produzidas pelo Teorema 2.1, i.e., através das Equações (2.18)-(2.19) não serão, necessariamente, minimais. Neste caso, suponha então que $\sigma^{(n)}(X)$ seja uma solução (minimal) no estágio n e $\sigma^{(n+1)}(X)$ seja uma solução no estágio $(n + 1)$, calculada a partir do Teorema 2.1. Vamos analisá-la agora:

a) se $\ell_{n+1} = \max[\ell_n, \; n + 1 - \ell_n]$, então $\sigma^{(n+1)}(X)$ já é a solução minimal procu-rada (no estágio $n + 1$);

b) se $\ell_{n+1} > \max[\ell_n, \; n + 1 - \ell_n]$, então é possível que exista um polinômio solução minimal $D^{(n+1)}(X)$ com grau ℓ tal que $\max[\ell_n, \; n + 1 - \ell_n] \leq \ell < \ell_{n+1}$. Um polinômio qualquer $D^{(n+1)}(X)$ com menor grau possível ℓ (no intervalo $\max[\ell_n, \; n + 1 - \ell_n] \leq \ell < \ell_{n+1}$) será uma solução minimal (no estágio $n + 1$); se, e somente se, o polinômio $\sigma^{(m)}(X)$ definido pela relação

$$D^{(n+1)}(X) - \sigma^{(n)}(X) = X^{n-m} \, \sigma^{(m)}(X)$$

for uma solução para as m primeiras somas de potência, com $d_m = -d_n$ e com $\sigma_0^{(m)}$ sendo um divisor de zero em R. Isto deve estar claro a partir das provas dos Lemas 2.1, 2.2 e do Teorema 2.1.

Iremos então agora descrever o algoritmo que resolve o problema original, i.e., as Equações (2.5). As suas entradas são as componentes do vetor síndrome s_1, s_2, ..., s_{2t}, todas pertencendo ao anel R. O algoritmo produzirá como saída um conjunto de valores σ_i ($1 \le i \le \nu$) tal que as Equações (2.5) sejam satisfeitas com o mínimo valor de ν possível.

► **Algoritmo de Berlekamp-Massey Modificado para Anéis Comutativos**[1] :

Inicie com a tabela:

Tabela 2.1

n	$\sigma^{(n)}(X)$	d_n	ℓ_n	$n - \ell_n$
-1	1	1	0	-1
0	1	s_1	0	0
1				
2				
⋮				
2t				

e complete-a da seguinte maneira:

1) $n \leftarrow 0$;

2) se $d_n = 0$, então $\sigma^{(n+1)}(X)$ e ℓ_{n+1} são dados por (2.18). Vá para 5);

3) se $d_n \neq 0$, então encontre m tal que a equação $d_n - y \cdot d_m = 0$[2] tenha solução em y sobre o anel R e $m - \ell_m$ tenha o máximo valor possível. $\sigma^{(n+1)}(X)$ e ℓ_{n+1} são dados por (2.19);

4) se $\ell_{n+1} = \max[\ell_n, n + 1 - \ell_n]$ então vá para o passo 5; caso contrário, procure uma solução $D^{(n+1)}(X)$ com grau ℓ mínimo possível no intervalo $\max[\ell_n, n + 1 - \ell_n] \le \ell < \max[\ell_n, \ell_m + n - m]$ tal que o polinômio $\sigma^{(m)}(X)$ definido pela Equação

[1] com identidade

[2] Esta equação é discutida no Apêndice 2.2.

$D^{(n+1)}(X) - \sigma^{(n)}(X) = X^{n-m} \cdot \sigma^{(m)}(X)$

seja uma solução para as m primeiras somas de potência tal que $d_m = -d_n$ e $\sigma_0^{(m)}$ seja um divisor de zero em R. Se este polinômio for encontrado, então $\sigma^{(n+1)}(X) \leftarrow D^{(n+1)}(X)$;

5) se $n < 2t - 1$ calcule $d_{n+1} = s_{n+2} + s_{n+1} \cdot \sigma_1^{(n+1)} + \ldots + s_{n+2-\ell_{n+1}} \cdot \sigma_{\ell_{n+1}}^{(n+1)}$;

6) $n \leftarrow n + 1$; se $n < 2t$ vá para 2); caso contrário, fim.

A resposta procurada será dada pelo polinômio $\sigma^{(2t)}(X)$, i.e., os seus coeficientes formam uma solução para as Equações (2.5).

Complexidade : comparando-se o Algoritmo de BM Modificado ao Algoritmo original, notamos que a principal diferença está na introdução do Passo 4), no qual a solução $\sigma^{(n)}(X)$ calculada no Passo 3) é testada ser uma solução minimal. Em caso negativo, uma busca deve ser feita para se encontrar uma solução minimal. Esta busca consiste em se testar se um polinômio $\sigma^{(m)}(X)$ (satisfazendo certas condições) é uma solução para as m primeiras somas de potência. Foi observado (mas não provado) que o número de polinômios $\sigma^{(m)}(X)$ a serem testados não é proibitivamente grande e a complexidade do algoritmo modificado é essencialmente a mesma do algoritmo original.

Terminado então este passo da determinação dos σ_i's ($1 \le i \le \nu$) que satisfazem as Equações (2.5), passaremos agora a descrever o terceiro passo do procedimento de decodificação dos códigos RS e BCH. Este passo realiza o cálculo dos números de localização de erro, $X_1, X_2, \ldots, X_\nu$, a partir dos σ_i's ($1 \le i \le \nu$).

Passo 3 *- Cálculo dos Números de Localização de Erro*

Este cálculo é feito resolvendo-se a equação polinomial $\rho(Z) = 0$, onde $\rho(Z) = Z^\nu \cdot \sigma^{(2t)}(Z^{-1}) = Z^\nu + \sigma_1 \cdot Z^{\nu-1} + \ldots + \sigma_{\nu-1} \cdot Z + \sigma_\nu$, no anel R ($Z_{p^k}$ ou $GR(p^k, r)$) (ver Apêndice 2.3). Note que as raízes de $\sigma^{(2t)}(Z) = 1 + \sigma_1 \cdot Z + \ldots + \sigma_\nu \cdot Z^\nu$ devem estar em R^*, já que R é um anel local e, portanto, as raízes de $\rho(Z)$ (o polinômio recíproco de $\sigma^{(2t)}(Z)$) serão as inversas das raízes de $\sigma^{(2t)}(Z)$.

Em geral, como mencionado, a solução do Sistema (2.5) não é única e, portanto, o conjunto dos valores σ_i's ($1 \le i \le \nu$) produzido pelo Algoritmo de BM modificado para anéis *pode não ser o mesmo definido pela Equação* (2.3). Como

conseqüência, as raízes do polinômio $\rho(Z)$ (com os coeficientes σ_i's provenientes do Algoritmo de BM modificado) *podem não ser os corretos números de localização de erro*. Entretanto, mostraremos a seguir que estes números podem de fato ser obtidos a partir das raízes de $\rho(Z)$, desde que os coeficientes σ_i's ($1 \le i \le \nu$) sejam uma solução do Sistema (2.5).

- **Relação entre as raízes de $\rho(Z)$ e os números de localização de erro**

Suponha que $\rho(Z)$ tenha pelo menos ν raízes distintas sobre o anel R, a saber, $Z_1, Z_2, \ldots, Z_\nu$, tais que

$$\rho(Z) = (Z - Z_1)(Z - Z_2) \ldots (Z - Z_\nu). \tag{2.20}$$

Note que pelo menos uma solução $\sigma(Z)$ produzida pelo Algoritmo de BM Modificado terá esta propriedade.

Iremos agora desenvolver um método que permite converter as raízes de $\rho(Z)$ nos corretos números de localização de erro. Suponha que estes números sejam $X_1, X_2, \ldots, X_\nu$, que as magnitudes dos erros sejam $Y_1, Y_2, \ldots, Y_\nu$, respectivamente, e que as raízes de $\rho(Z)$ sejam $Z_1, Z_2, \ldots, Z_\nu$. Isto implica que

$$Y_i X_i^j (Z^\nu + \sigma_1 . Z^{\nu-1} + \ldots + \sigma_{\nu-1} . Z + \sigma_\nu) = Y_i X_i^j (Z - Z_1)(Z - Z_2)\ldots(Z - Z_\nu) \tag{2.21}$$

para $1 \le i \le \nu$ e $1 \le j \le 2t - \nu$. Agora, substituindo Z por X_i e somando o primeiro membro para $1 \le i \le \nu$, obtemos

$$s_{j+\nu} + s_{j+\nu-1} . \sigma_1 + \ldots + s_{j+1} . \sigma_{\nu-1} + s_j . \sigma_\nu. \tag{2.22}$$

Note que a Equação (2.22) se anula para todo j tal que $1 \le j \le 2t$-ν (já que os σ_i's formam uma solução para o Sistema (2.5)) e conseqüentemente,

$$\sum_{i=1}^{\nu} Y_i . X_i^j (X_i - Z_1)(X_i - Z_2)\ldots(X_i - Z_\nu) = 0 \tag{2.23}$$

para $1 \le j \le 2t$-ν (o lado esquerdo de (2.23) é a soma do segundo membro de (2.21) para $1 \le i \le \nu$). Na forma matricial, as Equações (2.23) podem ser escritas como

$$\begin{bmatrix} X_1 & X_2 & \cdots & X_\nu \\ X_1^2 & X_2^2 & \cdots & X_\nu^2 \\ \vdots & \vdots & \cdots & \vdots \\ X_1^{2t-\nu} & X_2^{2t-\nu} & \cdots & X_\nu^{2t-\nu} \end{bmatrix} \begin{bmatrix} Y_1 P_1 \\ Y_2 P_2 \\ \vdots \\ Y_\nu P_\nu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.24}$$

onde $P_i = \prod_{\ell=1}^{\nu}(X_i - Z_\ell)$, para $1 \le i \le \nu$. A Equação (2.24) pode ser vista como um sistema linear homogêneo sobre o anel R nas incógnitas Y_1P_1, Y_2P_2, ..., $Y_\nu P_\nu$. O valor $2t$ - ν é sempre maior ou igual a ν (já que $\nu \le t$) e o posto[3] da matriz em (2.24) é ν (ver Apêndice 2.1), que é exatamente o número de incógnitas. Pelo Teorema 5.3 [16], isto implica que a solução única de (2.24) é a solução trivial, i.e., $Y_i.P_i = 0$, para $1 \le i \le \nu$.

A partir deste resultado podemos concluir que cada produtório P_i é necessariamente um divisor de zero em R e, portanto, em cada P_i ($1 \le i \le \nu$) existe pelo menos um ℓ-ésimo fator (X_i - Z_ℓ) que é um divisor de zero em R. Mais ainda, vale a seguinte propriedade: se o ℓ_1-ésimo fator em P_i é um divisor de zero d_1 e o ℓ_2-ésimo fator em P_k é também um divisor de zero d_2, então $\ell_1 \neq \ell_2$ para $i \neq k$. Isto pode ser mostrado da seguinte forma: suponha, por absurdo, que $\ell_1 = \ell_2$ para $i \neq k$. Então, $X_i - Z_{\ell_1} = d_1$ e $X_k - Z_{\ell_1} = d_2$ e, portanto, X_i - X_k seria um divisor de zero em R, o que é uma contradição se i ≠ k (ver Apêndice 2.1). Com tudo isto, podemos afirmar que correspondendo a cada Z_i existe um único número de localização de erro X_i ($1 \le i \le \nu$).

Baseados nestes fatos, podemos agora deduzir o seguinte procedimento para o cálculo dos números de localização de erro:

1) Calcule as raízes de $\rho(Z) = Z^\nu.\sigma^{(2t)}(Z^{-1})$ (o recíproco do polinômio produzido pelo Algoritmo de BM modificado), Z_1, Z_2, ..., Z_ν (um método para obtenção destas raízes está contido no Apêndice 2.3);

2) Sejam os elementos $X_0 = \alpha^0$, $X_1 = \alpha^1$, ..., $X_{n-1} = \alpha^{n-1}$; dentre estes, selecione aqueles X_i's que tornam as diferenças X_i - Z_j ($1 \le j \le \nu$) elementos divisores de zero em R (sabemos do parágrafo anterior que estes X_i's são univocamente determinados).

Portanto, os X_i's selecionados em 2) são os corretos números de localização de erro. Recorde que se $X_i = \alpha^i$, então isto significa que ocorreu um erro na posição i da palavra código.

Concluído este passo da localização dos erros, passaremos agora ao quarto e último passo do procedimento de decodificação dos códigos RS e BCH.

---

[3] O posto de uma matriz sobre um anel comutativo R (finito) é o maior número r tal que toda submatriz $r \times r$ tem determinante inversível em R [18].

Passo 4* - *Determinação das Magnitudes dos Erros

Para a completa decodificação, resta-nos agora descrever um método para a determinação das magnitudes Y_i ($1 \leq i \leq \nu$) dos erros. Inicialmente iremos mostrar que estas magnitudes ficam univocamente determinadas uma vez conhecidos os X_i's ($1 \leq i \leq \nu$) que são os números de localização de erro calculados no passo anterior.

Repare que as ν primeiras Equações (2.2) (que relacionam as componentes do vetor síndrome com os X_i's e Y_i's) podem ser escritas na forma:

$$\begin{bmatrix} X_1 & X_2 & \cdots & X_\nu \\ X_1^2 & X_2^2 & \cdots & X_\nu^2 \\ \vdots & \vdots & \cdots & \vdots \\ X_1^\nu & X_2^\nu & \cdots & X_\nu^\nu \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\nu \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_\nu \end{bmatrix} \tag{2.25}$$

No Apêndice 2.1, está mostrado que esta matriz ν x ν acima é não-singular, i.e., o seu determinante é um elemento inversível em R (Z_{p^k} ou $GR(p^k, r)$). Assim sendo, o vetor $\underline{Y} = (Y_1, Y_2, ..., Y_\nu)$ fica univocamente determinado. Ao invés de invertermos a matriz em (2.25) para encontrarmos o vetor $\underline{Y}$, iremos mostrar que o procedimento dado por Forney [19] para a determinação dos erros em códigos BCH sobre corpos ainda pode ser aplicado aqui.

Este procedimento requer o conhecimento dos números de localização de erro $X_1, X_2, ..., X_\nu$ e das suas funções simétricas elementares $\sigma_1, \sigma_2, ..., \sigma_\nu$ calculadas a partir da Equação (2.3).

Primeiramente, defina as funções simétricas elementares $\sigma_{j\ell}$ dos números de localização de erro $X_1, X_2, ..., X_{j-1}, X_{j+1}, ..., X_\nu$ através da relação:

$$\prod_{i \neq j} (X - X_i) = \sum_{\ell=0}^{\nu-1} \sigma_{j\ell} . X^{\nu-1-\ell} \tag{2.26}$$

Da Equação (2.3), temos que:

$$\prod_{i=1}^{\nu} (X - X_i) = \sum_{i=0}^{\nu} \sigma_i . X^{\nu-i} \tag{2.27}$$

onde σ_0 e $\sigma_{j,0}$ são iguais a 1, a unidade do anel R em consideração. A partir de (2.26) e (2.27), obtemos que:

$$(X - X_j) . \sum_{\ell=0}^{\nu-1} \sigma_{j\ell} . X^{\nu-1-\ell} = \sum_{i=0}^{\nu} \sigma_i . X^{\nu-i} \tag{2.28}$$

E disto vem que:

$$\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.X^{\nu-\ell} - \sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.X_j.X^{\nu-1-\ell} = \sum_{i=0}^{\nu} \sigma_i.X^{\nu-i} \qquad (2.29)$$

De (2.29) podemos concluir que os coeficientes $\sigma_{j\ell}$ podem ser obtidos recursivamente a partir de X_i e σ_i (já conhecidos) através da relação:

$$\sigma_{ji} = \sigma_i + X_j.\sigma_{j,i-1} \qquad (2.30)$$

para $0 \le i \le \nu\text{-}1$ e com $\sigma_0 = \sigma_{j,0} = 1$.

Denotando a magnitude de cada erro por Y_j, temos agora:

$$\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.s_{\nu-\ell} = \sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.\sum_{i=1}^{\nu} Y_i.X_i^{\nu-\ell} = \sum_{i=1}^{\nu} Y_i.X_i.\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.X_i^{\nu-1-\ell} \qquad (2.31)$$

Por (2.26) isto implica que

$$\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.s_{\nu-\ell} = \sum_{i=1}^{\nu} Y_i.X_i.\prod_{m\neq j}(X_i - X_m) = Y_j.X_j.\prod_{m\neq j}(X_j - X_m) \qquad (2.32)$$

A última igualdade em (2.32) segue do fato de que o somatório em questão só não se anula se $i = j$. Pela Equação (2.32) podemos concluir que:

$$\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.s_{\nu-\ell} = Y_j.\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.X_j^{\nu-\ell} \qquad (2.33)$$

E disto segue que cada Y_j ($1 \le j \le \nu$) é dado pela seguinte expressão:

$$Y_j = \frac{\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.s_{\nu-\ell}}{\sum_{\ell=0}^{\nu-1} \sigma_{j\ell}.X_j^{\nu-\ell}} \qquad (2.34)$$

Obs. : Note que o denominador de (2.34) é inversível em R pois é igual a $X_j.\prod_{m\neq j}(X_j - X_m)$ e como já mostramos nas Seções 1.4 e 1.5, essas diferenças $(X_j - X_m) = (\alpha^{\ell_1} - \alpha^{\ell_2})$ são inversíveis em R (Z_{p^k} ou GR(p^k, r)), para $0 \le \ell_1, \ell_2 \le n\text{-}1$ e $\ell_1 \neq \ell_2$.

Com isto terminamos o quarto passo da decodificação dos códigos RS e BCH que é a determinação das magnitudes Y_i ($1 \le i \le \nu$) dos erros, através da Equação (2.34).

Com isso, concluímos o procedimento de decodificação completo dos códigos RS e BCH definidos sobre anéis de inteiros residuais Z_{p^k}, para p um primo e k um inteiro maior ou igual a 1. Terminaremos esta seção com dois exemplos que ilustram a aplicação do procedimento de decodificação dos códigos RS e BCH, respectivamente.

Exemplo 2.2 (Decodificação de códigos RS) : Considere o código do Exemplo 1.9, Seção 1.4. Recordando os seus parâmetros temos que q = 49 (o código é sobre Z_{49}), p = 7, n = 6, r = 2, k = p - 2 = 5, D = 2 (dimensão do código) e d_{min} = 5. A matriz verificação de paridade é dada por:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} & \alpha^{16} & \alpha^{20} \end{bmatrix} = \begin{bmatrix} 1 & 3 & 9 & 27 & 32 & 47 \\ 1 & 9 & 32 & 43 & 44 & 4 \\ 1 & 27 & 43 & 34 & 36 & 41 \\ 1 & 32 & 44 & 36 & 25 & 16 \end{bmatrix}$$

Suponha agora que a palavra toda-nula $\underline{v}$ = (0 0 0 0 0 0) seja transmitida através do canal e que o mesmo introduza o vetor erro $\underline{e}$ = (0 0 7 0 14 0). O receptor terá como vetor recebido $\underline{r} = \underline{v} + \underline{e}$, onde + indica a adição módulo 49. Portanto, $\underline{r}$ = (0 0 7 0 14 0). O procedimento de decodificação é:

1) Cálculo do vetor síndrome:
   $\underline{s} = \underline{r}.H^t$, onde H é a matriz dada acima. Portanto,
   $\underline{s}$ = (21 7 21 21).

2) Cálculo das variáveis $\sigma_1, \sigma_2, \ldots, \sigma_v$ que satisfazem o Sistema (2.5). Para tanto, vamos aplicar o Algoritmo de BM modificado com as entradas $s_1 = 21$, $s_2 = 7$, $s_3 = 21$, $s_4 = 21$.

n	$\sigma^{(n)}(Z)$	d_n	ℓ_n	$n - \ell_n$
-1	1	1	0	-1
0	1	21	0	0
1	1 + 28Z	7	1	0
2	1 + 44Z	35	1	1
3	$1 + 39Z + 7Z^2$	7	2	1
4	$1 + 29Z + 8Z^2$	-	2	2

3) Determinação das raízes de $\rho(Z) = Z^2 + 29.Z + 8$ (o polinômio recíproco de $\sigma^{(4)}(Z)$) :

Aplicando o procedimento descrito no Apêndice 2.3 vamos encontrar que $Z_1 = 32$ e $Z_2 = 37$ são as raízes de $\rho(Z)$. Dentre os elementos $\alpha^0 = 1$, $\alpha = 3$, $\alpha^2 = 9$, $\alpha^3 = 27$, $\alpha^4 = 32$, $\alpha^5 = 47$, temos que $X_1 = \alpha^4 = 32$ e $X_2 = \alpha^2 = 9$ são tais que $X_1 - Z_1 = 0$ (divisor de zero em Z_{49}) e $X_2 - Z_2 = 21$ (divisor de zero em Z_{49}) e, portanto, X_1 e X_2 são os corretos números de localização de erro. As suas funções simétricas elementares σ_1 e σ_2 satisfazem a relação

$$(X - X_1).(X - X_2) = X^2 + \sigma_1.X + \sigma_2$$

e, portanto, $\sigma_1 = 8$ e $\sigma_2 = 43$.

4) Cálculo das magnitudes dos erros : Este cálculo é feito através do uso da Equação (2.34). Entretanto, primeiramente devemos determinar os coeficientes $\sigma_{j\ell}$ através de (2.30):

$\sigma_{11} = \sigma_1 + X_1.\sigma_{10} = 8 + 32.1 = 40$

$\sigma_{21} = \sigma_1 + X_2.\sigma_{20} = 8 + 9.1 = 17$

Agora aplicando a Equação (2.34), obtemos $Y_1 = 14$ e $Y_2 = 7$.

Com todos estes resultados podemos dizer que ocorreram dois erros: um na posição 2 da palavra-código com magnitude 7 e outro na posição 4 com magnitude 14, confirmando o que de fato havia ocorrido.

Exemplo 2.3 (Decodificação de códigos BCH) : Seja C um (8, 3)-código BCH sobre Z_9 que tem $g(x) = x^5 + 5x^4 + 4x^3 + 4x^2 + 3x + 8$ como polinômio gerador. De acordo com a notação da Seção 1.5, seja:

$R = GR(9, 2) = Z_9[X] / < x^2 + x + 2 >$: anel de extensão de Z_9;

G_8 : subgrupo cíclico de R^* que contém todas as raízes de $x^8 - 1$;

$\beta = (2 \quad 8)$: um elemento primitivo de G_8.

Pode-se verificar que β, β^2, β^3 e β^4 são raízes de g(x). Portanto, $d_{min}(C) \geq 5$ e este código tem capacidade de correção de $t = 2$ erros. A sua matriz verificação de paridade é dada por:

$$H = \begin{bmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 & \beta^7 \\ 1 & \beta^2 & \beta^4 & \beta^6 & 1 & \beta^2 & \beta^4 & \beta^6 \\ 1 & \beta^3 & \beta^6 & \beta & \beta^4 & \beta^7 & \beta^2 & \beta^5 \\ 1 & \beta^4 & 1 & \beta^4 & 1 & \beta^4 & 1 & \beta^4 \end{bmatrix}$$

Suponha agora que a palavra toda-nula $\underline{v} = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$ seja transmitida através do canal e que o mesmo introduza o vetor erro $\underline{e} = (0\ 3\ 0\ 0\ 0\ 0\ 6\ 0)$. Assim, o vetor recebido será $\underline{r} = \underline{v} + \underline{e} = (0\ 3\ 0\ 0\ 0\ 0\ 6\ 0)$. O procedimento de decodificação é:

1) Cálculo do vetor síndrome:
   $\underline{s} = \underline{r}.H^t$, onde H é a matriz dada acima. Portanto,
   $\underline{s} = (a_1\ a_2\ a_1\ a_1)$.

   onde $a_1 = (3\ 0)$ e $a_2 = (0\ 3)$

2) Cálculo das variáveis $\sigma_1, \sigma_2, \ldots, \sigma_v$ que satisfazem o Sistema (2.5). Para tanto, vamos aplicar o Algoritmo de BM modificado com as entradas $s_1 = a_1$, $s_2 = a_2$, $s_3 = a_1$, $s_4 = a_1$.

n	$\sigma^{(n)}(Z)$	d_n	ℓ_n	$n - \ell_n$
-1	1	1	0	-1
0	1	a_1	0	0
1	$1 + a_3 Z$	a_2	1	0
2	$1 + a_4 Z$	a_2	1	1
3	$1 + a_5 Z + a_1 Z^2$	a_6	2	1
4	$1 + a_7 Z + a_8 Z^2$	-	2	2

   onde : $a_1 = (3\ 0)$, $a_2 = (0\ 3)$, $a_3 = (6\ 0)$, $a_4 = (6\ 8)$, $a_5 = (5\ 8)$, $a_6 = (0\ 6)$, $a_7 = (3\ 8)$ e $a_8 = (0\ 2)$.

3) Determinação das raízes de $\rho(Z) = Z^2 + a_7.Z + a_8$ (o polinômio recíproco de $\sigma^{(4)}(Z)$) : Por inspeção direta, podemos encontrar que $Z_1 = (5\ 8)$ e $Z_2 = (1\ 2)$ são as raízes de $\rho(Z)$. Dentre os elementos $\beta^0 = 1$, $\beta = (2\ 8)$, $\beta^2 = (2\ 4)$, $\beta^3 = (3\ 1)$, $\beta^4 = (8\ 0)$, $\beta^5 = (7\ 1)$, $\beta^6 = (7\ 5)$, $\beta^7 = (6\ 8)$, temos que $X_1 = \beta$ e $X_2 = \beta^6$ são tais que $X_1 - Z_1 = (6\ 0)$ (divisor de zero em GR(9, 2)) e, $X_2 - Z_2 = (6\ 3)$ (divisor de zero em GR(9, 2)) e, portanto, X_1 e X_2 são os corretos números de localização de erro. As suas funções simétricas elementares satisfazem a relação:

   $(X - X_1).(X - X_2) = X^2 + \sigma_1.X + \sigma_2$

   e, portanto, $\sigma_1 = (0\ 5)$ e $\sigma_2 = (6\ 8)$.

4) Cálculo das magnitudes dos erros : Como no Exemplo 2.1, primeiro determinaremos os coeficientes $\sigma_{j\ell}$ através de (2.30):

   $\sigma_{11} = \sigma_1 + X_1.\sigma_{10} = (0\ 5) + (2\ 8).1 = (2\ 4)$
   $\sigma_{21} = \sigma_1 + X_2.\sigma_{20} = (0\ 5) + (7\ 5).1 = (7\ 1)$

Agora aplicando a Equação (2.34), obtemos $Y_1 = 3$ e $Y_2 = 6$.

Podemos então afirmar que ocorreram dois erros: um na posição 1 da palavra-código com magnitude 3 e outro na posição 6 com magnitude 6.

2.3. Geração de Seqüências

Como já dissemos anteriormente, o objetivo desta seção é apenas apresentar uma aplicação do algoritmo de BM modificado. Esta aplicação é a síntese de circuitos lineares de deslocamentos com realimentação (ou LFSR, "linear feedback shift-register", do inglês) que geram uma dada seqüência finita de dígitos. Este problema foi tratado em [25] para o caso em que estes dígitos pertencem a um corpo. O que faremos aqui é generalizá-lo para o caso em que a seqüência pertence a um anel comutativo R com identidade.

Vamos agora fazer uma breve revisão da teoria. Um *LFSR* de comprimento L, mostrado na Figura 2.1, consiste de uma cascata de L atrasadores (registros de deslocamento) e alguns multiplicadores e somadores capazes de gerar uma combinação linear dos conteúdos destes registros.

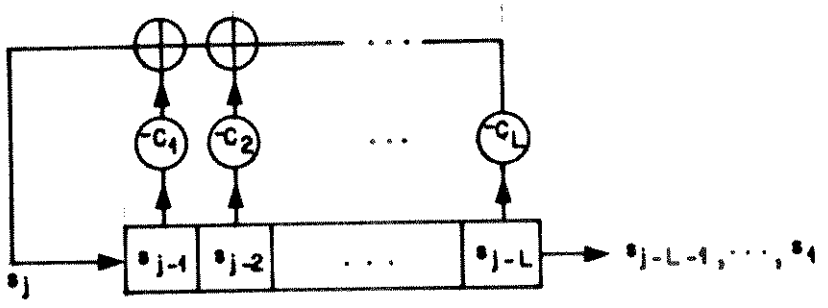

Figura 2.1 - LFSR de comprimento L.

A saída do LFSR é o conteúdo do último registro. Os conteúdos iniciais $s_1, s_2, .., s_L$ dos L atrasadores coincidem com os L primeiros dígitos de saída e os dígitos seguintes de saída são determinados através da relação de recorrência

$$s_j = -\sum_{i=1}^{L} c_i \cdot s_{j-i} \qquad (2.35)$$

contrário, haverá mais de um LFSR minimal de comprimento L que gera $\{s_i\}_{i=1}^{N}$ [24].

O algoritmo de BM pode ainda ser usado como parte de um codificador de fonte ou compressor de dados, conforme discutido em [25].

Finalmente, vale ressaltar que o problema da síntese de LFSR's minimais que geram seqüências de elementos pertencentes a anéis do tipo Z_m foi tratado em [25]. Entretanto, o algoritmo apresentado difere substancialmente do algoritmo original de BM para corpos GF(q), aumentando inclusive o número de variáveis envolvidas. Além disso, se m fatorar em um produto de r primos distintos, então o algoritmo deve ser aplicado r vezes para depois usarmos o Teorema do Resto Chinês [23] que finalmente produzirá a resposta desejada.

Os exemplos a seguir ilustram a aplicação do algoritmo descrito neste capítulo para a síntese de LFSR's. O Exemplo 2.4 também foi apresentado em [26], i.e., lá procurava-se determinar um LFSR minimal que gerasse a mesma seqüência inicial. Apesar dos LFSR's sintetizados por ambos algoritmos produzirem de fato a seqüência finita desejada, eles não são equivalentes, i.e., a partir de um certo ponto começam a gerar elementos diferentes. Isto é devido ao fato de que a equação em y, $d_n - y \,.\, d_m = 0$, apresentou mais de uma solução em algum estágio.

Exemplo 2.4 : Encontrar o LFSR minimal que gera a seqüência $s_1 = 6$, $s_2 = 3$, $s_3 = 1$, $s_4 = 5$ e $s_5 = 6$ sobre o anel Z_9. Aplicando o algoritmo de BM modificado, encontramos:

n	$\sigma^{(n)}(X)$	d_n	ℓ_n	$n - \ell_n$
-1	1	1	0	-1
0	1	6	0	0
1	$1 + 3X$	3	1	0
2	$1 + 7X$	4	1	1
3	$1 + 7X + 5X^3$	6	3	0
4	$1 + X + 3X^2 + 5X^3$	2	3	1
5	$1 + X + 7X^2 + 6X^3$	-	3	2

Portanto, $C(X) = 1 + X + 7X^2 + 6X^3$ e o LFSR minimal que gera a seqüência dada tem as suas saídas relacionadas através de:

$$s_n + s_{n-1} + 7.s_{n-2} + 6.s_{n-3} = 0 \pmod 9, \quad n \geq 4.$$

para $j = L+1, L+2, \ldots$. Os dígitos de saída e os ***coeficientes de realimentação*** $c_1, c_2, \ldots, c_L$ pertencem ao mesmo anel R. Quando $c_L = 0$, dizemos que o LFSR é ***singular***.

Diz-se que um LFSR ***gera*** uma seqüência finita de dígitos $s_1, s_2, \ldots, s_N$ quando esta seqüência coincide com os N primeiros dígitos de saída do mesmo, para algum conteúdo inicial. Se $L \geq N$, o LFSR sempre gera a seqüência e se $L < N$, o LFSR gera a seqüência se, e somente se,

$$s_j + s_{j-1}.c_1 + \ldots + s_{j-L+1}.c_{L-1} + s_{j-L}.c_L = 0 \qquad (2.36)$$

para $L+1 \leq j \leq N$.

2.3.1. Algoritmo para síntese de LFSR's

Em [25] está mostrado que o algoritmo usado para decodificação de códigos BCH [24] também pode ser usado para sintetizar um LFSR de comprimento mínimo L que gera uma seqüência prescrita. Em outras palavras, os problemas de geração de LFSR's e decodificação de código BCH são equivalentes.

De forma análoga também podemos aplicar o algoritmo de BM modificado (Seção 2.2) para sintetizarmos um LFSR de comprimento mínimo que gera uma dada seqüência $\{s_i\}_{i=1}^{N}$ de elementos pertencentes a um anel R. Isto é justificado quando comparamos os Sistemas de Equações (2.5) e (2.36). Em ambos os casos, o objetivo é encontrar a menor quantidade de variáveis (v ou L) que satisfazem os respectivos conjuntos de equações.

Aqui as entradas do algoritmo serão os elementos $s_1, s_2, \ldots, s_N$ que formam a seqüência dada e a saída do mesmo será o polinômio

$$C(X) = 1 + c_1.X + \ldots + c_L.X^L$$

na indeterminada X, cujos coeficientes são os coeficientes de realimentação do LFSR minimal de comprimento L que gera $\{s_i\}_{i=1}^{N}$. Este LFSR minimal será único se, e somente se, $2L \leq N$ e em cada estágio do algoritmo a equação linear em y, $d_n - y \,.\, d_m = 0$, apresentar solução única (d_n e d_m são a n-ésima e a m-ésima discrepância, respectivamente). Caso

Exemplo 2.5 : Encontrar o LFSR minimal que gera a seqüência $s_1 = 3$, $s_2 = 7$, $s_3 = 0$, $s_4 = 8$ e $s_5 = 3$ sobre o anel Z_9. Aplicando o algoritmo de BM modificado, encontramos:

n	$\sigma^{(n)}(X)$	d_n	ℓ_n	$n - \ell_n$
-1	1	1	0	-1
0	1	3	0	0
1	$1 + 6X$	7	1	0
2	$1 + 6X + 2X^2$	3	2	0
3	$1 + 3X + 2X^2$	4	2	1
4	$1 + 6X + 4X^2$	6	2	2
5	$1 + 4X^2 + 6X^3$	-	3	2

Portanto, $C(X) = 1 + 4X^2 + 6X^3$ e o LFSR minimal que gera a seqüência dada tem as suas saídas relacionadas através de:

$$s_n + 4.s_{n-2} + 6.s_{n-3} = 0 \pmod 9, \quad n \geq 4.$$

Exemplo 2.6 : Encontrar o LFSR minimal que gera a seqüência $s_1 = 33$, $s_2 = 12$, $s_3 = 1$, $s_4 = 5$ e $s_5 = 6$ sobre o anel Z_{36}. Aplicando o algoritmo de BM modificado, encontramos:

n	$\sigma^{(n)}(X)$	d_n	ℓ_n	$n - \ell_n$
-1	1	1	0	-1
0	1	33	0	0
1	$1 + 3X$	3	1	0
2	$1 + 28X$	13	1	1
3	$1 + 28X + 23X^3$	0	3	0
4	$1 + 28X + 23X^3$	26	3	1
5	$1 + 28X + 34X^2 + 3X^3$	-	3	2

Portanto, $C(X) = 1 + 28X + 34X^2 + 3X^3$ e o LFSR minimal que gera a seqüência dada tem as suas saídas relacionadas através de:

$$s_n + 28.s_{n-1} + 34.s_{n-2} + 3.s_{n-3} = 0 \pmod{36}, \quad n \geq 4.$$

2.4. Conclusões

Neste capítulo foram apresentados os métodos para a decodificação dos códigos de Hamming, RS e BCH definidos sobre anéis de inteiros residuais Z_q, onde q é uma potência de primo. Os algoritmos apresentados mantêm as características essenciais dos correspondentes algoritmos para decodificação de códigos sobre corpos GF(q), não implicando, portanto, em um aumento considerável de complexidade.

Mostramos ainda que o algoritmo de BM modificado, usado na decodificação de códigos BCH sobre anéis Z_q, também pode ser aplicado para sintetizar LFSR's minimais que geram seqüências de elementos pertencentes a anéis comutativos.

Os algoritmos descritos neste capítulo serão usados quando da decodificação de códigos sobre grupos abelianos (Capítulo 4), contruídos a partir da concatenação de códigos sobre anéis Z_q. No Capítulo 3 será apresentada uma generalização do algoritmo de BM para multiseqüências a qual será usada para decodificar outras classes de códigos cíclicos.

APÊNDICE 2.1

Iremos mostrar que a solução do Sistema Linear (2.5) (nas incógnitas $\sigma_1, \sigma_2, ..., \sigma_\nu$) dado por

$$\begin{bmatrix} s_1 & s_2 & \cdots & s_\nu \\ s_2 & s_3 & \cdots & s_{\nu+1} \\ \vdots & \vdots & & \vdots \\ s_{2t-\nu} & s_{2t-\nu+1} & \cdots & s_{2t-1} \end{bmatrix} \begin{bmatrix} \sigma_\nu \\ \sigma_{\nu-1} \\ \vdots \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} s_{\nu+1} \\ s_{\nu+2} \\ \vdots \\ s_{2t} \end{bmatrix} \tag{2.37}$$

é única se, e somente se, todas as magnitudes Y_i $(1 \leq i \leq \nu)$ dos erros ocorridos forem elementos inversíveis no anel sobre o qual o código está definido. Caso contrário, o sistema terá mais do que uma solução. Recorde que os valores s_i's são as componentes do vetor síndrome e que os valores σ_i's são as funções simétricas elementares.

Por construção, sabemos que este Sistema (2.37) admite ao menos uma solução que são as próprias funções simétricas elementares dos X_i's. Considere agora a submatriz M associada às ν primeiras Equações (2.37). Então,

$$M = \begin{bmatrix} s_1 & s_2 & \cdots & s_\nu \\ s_2 & s_3 & \cdots & s_{\nu+1} \\ \vdots & \vdots & \cdots & \vdots \\ s_\nu & s_{\nu+1} & \cdots & s_{2\nu-1} \end{bmatrix}$$

e podemos usar as Equações (2.2) para constatarmos que $M = V.D.V^T$, onde:

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_\nu \\ \vdots & \vdots & \cdots & \vdots \\ X_1^{\nu-1} & X_2^{\nu-1} & \cdots & X_\nu^{\nu-1} \end{bmatrix} \quad \text{e} \quad D = \begin{bmatrix} Y_1X_1 & 0 & \cdots & 0 \\ 0 & Y_2X_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Y_\nu X_\nu \end{bmatrix}$$

A matriz V é uma matriz de Vandermonde e seu determinante é expresso por

det $V = \prod_{i>j}(X_i - X_j)$. Agora recorde que os X_i's são os números de localização de erro e são da forma α^{ℓ} (onde α está definido pela Equação (2.1)) para algum ℓ entre 0 e n-1 (inclusive). Já vimos nas Seções 1.4 e 1.5 que em ambos os anéis R (Z_{p^k} e GR(p^k, r)) a expressão $\alpha^{\ell_1} - \alpha^{\ell_2}$ resulta sempre em um elemento inversível desde que $0 \leq \ell_1, \ell_2 \leq n$-1 e $\ell_1 \neq \ell_2$. Assim sendo, det V e conseqüentemente, det V^T são inversíveis em R.

A matriz D tem determinante det $D = \prod_{i=1}^{v} Y_i X_i$ o qual é um divisor de zero em R se, e somente se, algum Y_i ($1 \leq i \leq v$) for um divisor de zero em R.

Portanto, podemos afirmar que det M = (det V)^2.(det D) é inversível (i.e., M tem posto v) se, e somente se, todos os Y_i ($1 \leq i \leq v$) forem elementos inversíveis em R, i.e., se, e somente se, as magnitudes de todos os erros ocorridos forem inversíveis em R. E neste caso, o Sistema Linear (2.37) ou (2.5) terá solução única.

Finalmente, para o leitor interessado em maiores detalhes a respeito de sistemas de equações lineares sobre anéis comutativos, [16] ou [17] são boas referências.

APÊNDICE 2.2

Iremos discutir aqui a equação linear em x, $ax - b = 0$, sobre o anel R, considerando dois casos:

<u>*Caso 1*</u> : R é o anel Z_m, $m \geq 2$. Defina d = mdc (a, m)

i) se $d \nmid b$, então a equação $ax - b = 0$ não admite solução em Z_m;

ii) se $d \mid b$, então a equação $ax - b = 0$ admite solução. Temos dois casos:

a) $d = 1$: a solução é única e é dada por

$x = b.a^{\phi(m)-1}$

onde ϕ é a função de Euler;

b) $d \neq 1$: defina $a' = a/d$, $b' = b/d$ e $m' = m/d$. Uma solução é dada por

$x = x_0 = b'.(a')^{\phi(m')-1}$

e as outras $d - 1$ soluções (módulo m) são dadas por

$x = x_0 + t.m'$, $\quad 1 \leq t \leq d - 1.$

A função de Euler, $\phi(m)$, definida para $m \geq 2$, representa o número de inteiros positivos menores que m e que são relativamente primos a m. Por exemplo: $\phi(2) = 1$, $\phi(3) = 2$, $\phi(4) = 2$, ... e em geral mostra-se que se $m = p_1^{e_1}.p_2^{e_2}...p_r^{e_r}$ (onde os p_i, $0 \leq i \leq m$, são os fatores primos distintos de m), então

$$\phi(m) = m.\left(1-\frac{1}{p_1}\right).\left(1-\frac{1}{p_2}\right)...\left(1-\frac{1}{p_r}\right)$$

O leitor interessado na demonstração das propriedades usadas neste Caso 1 deverá referir-se a [20]-[22].

<u>*Caso 2*</u> : R é o anel de Galois GR(p^k, r). Temos três casos a considerar:

i) se $a \in R^*$, então a única solução de $ax - b = 0$ é dada por

$x = b.a^{-1}$;

ii) se a pertence ao grupo aditivo dos divisores de zero de R, i.e., $a \in pR$ e $b \in R^*$, então a equação $ax - b = 0$ não admite solução em R;

iii) se $a, b \in pR$, defina $a = p^s a'$ e $b = p^t b'$ onde $a', b' \in R^*$. Então $ax - b = 0 \Leftrightarrow (p^s a').x - (p^t b') = 0$. Temos dois casos a considerar:

a) se $t \geq s$, então uma solução de $ax - b = 0$ é dada por

$x = x_0 = p^{t-s} b'.(a')^{-1}$;

b) se $t < s$, então a equação $ax - b = 0$ não admite solução pois para tanto a expressão $(p^{s-t} a')x - b'$ deveria pertencer a pR, o que nunca ocorre.

Obs. : O cálculo de uma expressão do tipo $f.g^{-1}$, onde $f \in R$ e $g \in R^*$, está descrito com maiores detalhes na Seção 1.5.2.

APÊNDICE 2.3

Iremos discutir aqui a solução da equação polinomial f(x) = 0, onde $f(x) = f_0 + f_1.x + f_2.x^2 + ... + f_n.x^n$, sobre o anel R, considerando dois casos.

Caso 1 : R é o anel Z_{p^r}, onde p é primo e $r \geq 1$ (o caso mais geral, $R = Z_m$ ($m \geq 2$), será deduzido a partir deste. Vamos apenas exibir os passos do algoritmo; as demonstrações podem ser encontradas em [23].

O problema é então determinar as soluções de $f(x) \equiv 0 \pmod{p^r}$. A solução será obtida de forma iterativa, i.e., iremos passo a passo determinando a solução de $f(x) \equiv 0 \pmod{p^i}$ para i de 1 até r. O procedimento é:

i) Resolva $f(x) \equiv 0 \pmod{p}$ por tentativa e erro, por exemplo. Seja c_1 uma solução encontrada;

ii) A solução da congruência $f(x) \equiv 0 \pmod{p^i}$ ($i \geq 2$) é dada por $c_i = c_{i-1} + t.p^{i-1}$, onde c_{i-1} é uma solução de $f(x) \equiv 0 \pmod{p^{i-1}}$ e t é uma solução de

$$f'(c_{i-1}).t \equiv -\frac{f(c_{i-1})}{p^{i-1}} \pmod{p} \quad ,$$

onde f' representa a derivada formal de f. No Apêndice 2.2 está descrito um método de solução desta congruência linear em t.

O processo termina após tivermos obtido todos os c_r's, i.e., todas as soluções de $f(x) \equiv 0 \pmod{p^r}$.

Obs. : No caso geral, para se resolver uma congruência polinomial do tipo $f(x) \equiv 0 \pmod{m}$ onde $m = p_1^{e_1}.p_2^{e_2}...p_s^{e_s}$, resolvemos individualmente cada uma das congruências

$$f(x) \equiv 0 \pmod{p_i^{e_i}}$$

para $1 \leq i \leq s$. Seja c_i uma solução da i-ésima congruência. Então, uma solução x de $f(x) \equiv 0 \pmod{m}$ é encontrada resolvendo-se o sistema de congruências

$$\begin{cases} x \equiv c_1 \pmod{p_1^{e_1}} \\ x \equiv c_2 \pmod{p_2^{e_2}} \\ \quad \vdots \\ x \equiv c_s \pmod{p_s^{e_s}} \end{cases}$$

através do Teorema do Resto Chinês [22].

Caso 2 : R é o anel de Galois GR(p^k, r). Aqui os valores de x que satisfazem f(x) = 0 são r-uplas; portanto, a busca de soluções irá requerer um método eficiente de resolução de congruências polinomiais módulo p^k, mas envolvendo duas ou mais variáveis. Até que se disponha de tal método, o procedimento que iremos usar é o de se fazer busca exaustiva em R das raízes de f(x). Isto é análogo ao que se faz quando da decodificação de códigos BCH sobre corpos GF(q).

CAPÍTULO 3

GERAÇÃO DE MULTISEQÜÊNCIAS E DECODIFICAÇÃO DE CÓDIGOS CÍCLICOS

Neste capítulo o algoritmo de BM adaptado para anéis comutativos será generalizado para o caso de geração de multiseqüências.

Primeiramente será mostrado que este problema é um caso especial de um problema mais geral que é o de se encontrar o menor conjunto inicial de colunas linearmente dependentes em uma matriz sobre um anel comutativo e que é resolvido através de um algoritmo conhecido como Algoritmo Iterativo Fundamental (AIF). O algoritmo de BM generalizado para multiseqüências será então derivado a partir de um refinamento do AIF.

A seguir, consideraremos o problema da decodificação de códigos cíclicos definidos sobre anéis Z_q, até o limitante de Hartmann-Tzeng. Neste caso, são apresentadas duas ou mais seqüências de síndromes e então o procedimento referente à localização dos erros será implementado via o uso do algoritmo de BM para multiseqüências. A seguir, o problema referente à determinação da magnitude dos erros será resolvido através do procedimento proposto por Forney (para códigos BCH), porém com pequenas modificações.

A teoria apresentada neste capítulo é uma generalização da teoria apresentada em [27].

3.1. Descrição do Problema

Sejam:

$$s_1^{(h)}, s_2^{(h)}, \ldots, s_N^{(h)}$$

para $h = 1, 2, \ldots, t$, t seqüências cada uma de comprimento N, sobre um anel comutativo R (com identidade). Para um inteiro positivo ℓ menor que N, seja $\sigma(X) = \sigma_0 + \sigma_1.X + \ldots + \sigma_\ell.X^\ell$ um polinômio sobre R, onde $\sigma_0 = 1$ e $\sigma_1, \sigma_2, \ldots, \sigma_\ell$ podem assumir qualquer valor em R. Se

$$s_j + \sigma_1.s_{j-1}^{(h)} + \ldots + \sigma_\ell.s_{j-\ell}^{(h)} = 0, \tag{3.1}$$

para $j = \ell + 1, \ell + 2, \ldots, N$ e $h = 1, 2, \ldots, t$, então ℓ e $\sigma(X)$ especificam completamente um LFSR de comprimento ℓ que gera cada uma destas t seqüências e com conteúdo inicial $s_1^{(h)}, s_2^{(h)}, \ldots, s_\ell^{(h)}$, conforme ilustra a Figura 3.1.

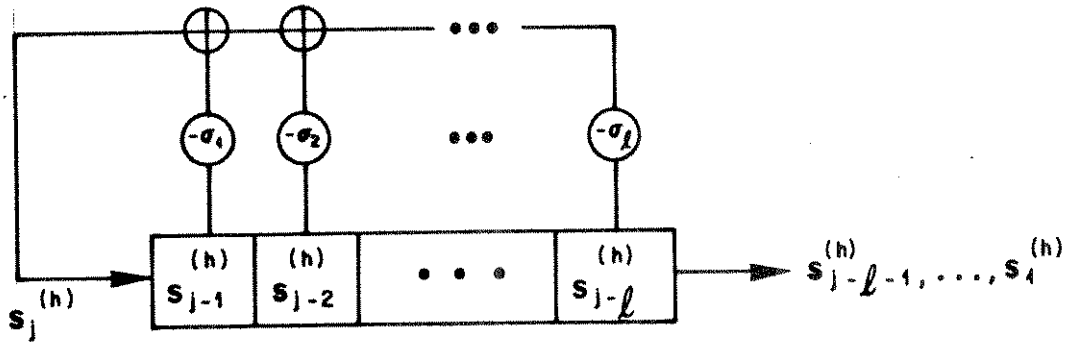

Figura 3.1 - LFSR de comprimento ℓ com conexões especificadas por $\sigma(X)$.

Dadas as t seqüências

$$\left\{s_i^{(h)}\right\}_{i=1}^{N}$$

para $h = 1, 2, \ldots, t$, o problema é então determinar um polinômio $\sigma(X)$ de grau ℓ (mínimo possível) satisfazendo 3.1, i.e., determinar um LFSR de comprimento mínimo que é capaz de gerar estas t seqüências.

Agora considere a seguinte matriz:

$$
S = \left[\begin{array}{cccccccc}
s_1^{(1)} & s_2^{(1)} & \cdots & s_{\ell}^{(1)} & s_{\ell+1}^{(1)} & \cdots & s_{N-1}^{(1)} & s_N^{(1)} \\
s_1^{(2)} & s_2^{(2)} & \cdots & s_{\ell}^{(2)} & s_{\ell+1}^{(2)} & \cdots & s_{N-1}^{(2)} & s_N^{(2)} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
s_1^{(t)} & s_2^{(t)} & \cdots & s_{\ell}^{(t)} & s_{\ell+1}^{(t)} & \cdots & s_{N-1}^{(t)} & s_N^{(t)} \\
\hline
s_2^{(1)} & s_3^{(1)} & \cdots & s_{\ell+1}^{(1)} & s_{\ell+2}^{(1)} & \cdots & s_N^{(1)} & x_{N+1}^{(1)} \\
s_2^{(2)} & s_3^{(2)} & \cdots & s_{\ell+1}^{(2)} & s_{\ell+2}^{(2)} & \cdots & s_N^{(2)} & x_{N+1}^{(2)} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
s_2^{(t)} & s_3^{(t)} & \cdots & s_{\ell+1}^{(t)} & s_{\ell+2}^{(t)} & \cdots & s_N^{(t)} & x_{N+1}^{(t)} \\
\hline
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\hline
s_{N-1}^{(1)} & s_N^{(1)} & \cdots & x_{N-2+\ell}^{(1)} & x_{N-1+\ell}^{(1)} & \cdots & x_{2N-3}^{(1)} & x_{2N-2}^{(1)} \\
s_{N-1}^{(2)} & s_N^{(2)} & \cdots & x_{N-2+\ell}^{(2)} & x_{N-1+\ell}^{(2)} & \cdots & x_{2N-3}^{(2)} & x_{2N-2}^{(2)} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
s_{N-1}^{(t)} & s_N^{(t)} & \cdots & x_{N-2+\ell}^{(t)} & x_{N-1+\ell}^{(t)} & \cdots & x_{2N-3}^{(t)} & x_{2N-2}^{(t)} \\
\hline
s_N^{(1)} & x_{N+1}^{(1)} & \cdots & x_{N-1+\ell}^{(1)} & x_{N+\ell}^{(1)} & \cdots & x_{2N-2}^{(1)} & x_{2N-1}^{(1)} \\
s_N^{(2)} & x_{N+1}^{(2)} & \cdots & x_{N-1+\ell}^{(2)} & x_{N+\ell}^{(2)} & \cdots & x_{2N-2}^{(2)} & x_{2N-1}^{(2)} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
s_N^{(t)} & x_{N+1}^{(t)} & \cdots & x_{N-1+\ell}^{(t)} & x_{N+\ell}^{(t)} & \cdots & x_{2N-2}^{(t)} & x_{2N-1}^{(t)}
\end{array}\right]
$$

Podemos então concluir que o problema de se determinar o mínimo ℓ e $\sigma(X)$ que satisfaça o Sistema de Equações (3.1) é equivalente a determinar o valor mínimo ℓ tal que as $\ell+1$ primeiras colunas de S sejam "linearmente dependentes" (o significado dessa expressão será explicado mais adiante) e ainda encontrar os coeficientes pertencentes a esta combinação linear.

Os valores $x_i^{(h)}$ em S, para $N+1 \leq i \leq 2N-1$ e $1 \leq h \leq t$, podem em princípio assumir

qualquer valor em R. Entretanto, eles podem ser especificados a fim de que satisfaçam a relação de dependência.

Dessa forma, o problema da geração de multiseqüências pode ser tratado como um caso especial do problema mais geral que é o de encontrar o menor conjunto inicial de colunas linearmente dependentes em uma matriz M x N, com posto menor que N, sobre um anel comutativo R (com identidade).

Seja:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix}$$

Dizemos que as $\ell + 1$ primeiras colunas de A, para $0 \leq \ell < N$, são "linearmente dependentes" se existirem $c_1, c_2, \ldots, c_\ell$ em R, não todos nulos, tais que:

$$a_{i,\ell+1} + c_1.a_{i,\ell} + \ldots + c_\ell.a_{i,1} = 0 \ , \qquad (3.2)$$

para $i = 1, 2, \ldots, M$.

Obs.: Note que o termo "linearmente dependente" aparece entre aspas pois por definição, as n-uplas $\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_\ell \in R^n$ são linearmente dependentes se existirem $\alpha_1, \alpha_2, \ldots, \alpha_\ell$, não todos nulos, tais que:

$$\alpha_1.\underline{v}_1 + \alpha_2.\underline{v}_2 + \ldots + \alpha_\ell.\underline{v}_\ell = 0$$

Repare então que a definição dada neste capítulo foi adaptada convenientemente para o problema que desejamos resolver.

Exemplo 3.1 : Considere a matriz A (dada abaixo), com entradas em Z_6:

$$A = \begin{bmatrix} 2 & 3 & 5 & 2 & 1 & \cdots \\ 3 & 5 & 2 & 1 & 2 & \cdots \\ 5 & 2 & 1 & 0 & 4 & \cdots \\ 2 & 2 & 4 & 3 & 5 & \cdots \end{bmatrix}$$

Temos aqui $\ell = 2$, $c_1 = 5$, $c_2 = 5$.

3.2. O Algoritmo Iterativo Fundamental

Nesta seção iremos descrever um algoritmo - algoritmo iterativo fundamental (AIF) - que resolve o problema, descrito na Seção 3.1, da determinação do menor conjunto inicial de colunas linearmente dependentes de uma matriz M x N (de posto menor que N), definida sobre um anel comutativo R (com identidade).

Vamos primeiramente introduzir uma notação apropriada. Sejam:

$$C(X) = c_0 + c_1.X + ... + c_\ell.X^\ell$$

e

$$a^{(i)}(X) = a_{i,0} + a_{i,1}.X + ... + a_{i,N}.X^N ,$$

onde $c_0 = 1$ e $a_{i,0} = 1$ para $i = 1, 2, ..., M$. Seja ainda $[C(X).a^{(i)}(X)]_n$ o coeficiente de x^n em $C(X).a^{(i)}(X)$ para $\ell+1 \leq n \leq N$. Então:

$$[C(X).a^{(i)}(X)]_n = c_0.a_{i,n} + c_1.a_{i,n-1} + ... + c_\ell.a_{i,n-\ell} = \sum_{j=0}^{\ell} c_j.a_{i,n-j} \qquad (3.3a)$$

Também,

$$[C(X).a^{(i)}(X).x^p]_n = [C(X).a^{(i)}(X)]_{n-p} = \sum_{j=0}^{\ell} c_j.a_{i,n-p-j}$$

Com isto, podemos reformular o problema da seguinte maneira: encontrar o mínimo ℓ e um polinômio C(X) de grau $\partial C \leq \ell$ tal que:

$$[C(X).a^{(i)}(X)]_{\ell+1} = 0 , \qquad (3.4)$$

para $i = 1, 2, ..., M$. O algoritmo que iremos descrever para resolver (3.4) é do tipo

iterativo. Começando com a primeira coluna, examinamos os elementos nas sucessivas colunas de A, um por um, desde a primeira até a última linha.

Para cada coluna j, para $j = 1, 2, \ldots, \ell$, seja:

$$C^{(i-1,j)}(X) = c_0^{(i-1,j)} + c_1^{(i-1,j)}.X + \ldots + c_{j-1}^{(i-1,j)}.X^{j-1} = \sum_{k=0}^{j-1} c_k^{(i-1,j)}.X^k \qquad (3.5)$$

onde $1 \leq i \leq M$ e $c_0^{(i-1,j)} = 1$, o polinômio com a propriedade de que:

$$[C^{(i-1,j)}(X).a^{(h)}(X)]_j = a_{h,j} + c_1^{(i-1,j)}.a_{h,j-1} + \ldots + c_{j-1}^{(i-1,j)}.a_{h,1} = 0\,,$$

para $h \leq i$ -1.

O polinômio inicial para a coluna j é denotado por $C^{(0,j)}$ e $C^{(0,1)}(X) = 1$ é o polinômio inicial para a primeira coluna. Seja:

$$d_{i,j} = [C^{(i-1,j)}(X).a^{(i)}(X)]_j = a_{i,j} + c_1^{(i-1,j)}.a_{i,j-1} + \ldots + c_{j-1}^{(i-1,j)}.a_{i,1} \qquad (3.6)$$

a discrepância na linha i e coluna j.

Para uma determinada coluna j, se $d_{i,j} = 0$ para $i = 1, 2, \ldots, r$-1, então temos que:

$$C^{(0,j)}(X) = C^{(1,j)}(X) = \ldots = C^{(r-1,j)}(X) \qquad (3.7a)$$

Se $d_{i,j} \neq 0$ e o vetor $(a_{1,j}, a_{2,j}, \ldots, a_{r,j})^T$ não for linearmente dependente dos j-1 vetores anteriores $(a_{1,k}, a_{2,k}, \ldots, a_{r,k})^T$, para $k = 1, 2, \ldots, j$-1, então definimos o polinômio final na coluna j como:

$$C^{(j)}(X) = C^{(r-1,j)}(X) \qquad (3.7b)$$

e avançamos para a próxima coluna. O primeiro elemento desta próxima coluna é então examinado com $C^{(0,j+1)}(X) = C^{(j)}(X) = C^{(r-1,j)}(X)$.

Se $d_{r,j} \neq 0$ e o vetor $(a_{1,j}, a_{2,j}, ..., a_{r,j})$ for linearmente dependente dos j - 1 vetores anteriores $(a_{1,k}, a_{2,k}, ..., a_{r,k})^T$ para $k = 1, 2, ..., j-1$, então em alguns casos pode-se encontrar $C^{(r,j)}$ de acordo com o procedimento dado no lema a seguir.

Lema 3.1 : Dados $C^{(r-1,j)}(X)$ e $d_{r,j} \neq 0$, se existir um polinômio final $C^{(u)}(X)$ na coluna u, onde $1 \leq u < j$, tal que $C^{(u)}(X) = C^{(r-1,u)}(X)$ e que a equação linear $d_{r,j} - y \, . \, d_{r,u} = 0$, admita uma solução em R, então:

$$C^{(r,j)}(X) = C^{(r-1,j)}(X) - y \, . \, C^{(u)}(X) \, . \, X^{j-u} \tag{3.8}$$

é tal que

$$[C^{(r,j)}(X) \, . \, a^{(i)}(X)]_j = 0 \quad \text{para} \quad i = 1, 2, ..., r-1, r \tag{3.9}$$

Prova : Da definição de $C^{(r-1,j)}(X)$, $C^{(u)}(X)$ e de (3.8) temos:

$$[C^{(r,j)}(X).a^{(i)}(X)]_j = [C^{(r-1,j)}(X).a^{(i)}(X)]_j - y[C^{(u)}(X).a^{(i)}(X)]_u =$$

$$= \begin{cases} 0 - y.0, & \text{para} \quad i = 1,2,...,r-1 \\ d_{r,j} - y.d_{r,u}, & \text{para} \quad i = r \end{cases}$$ ■

Temos agora $C^{(r,j)}(X)$. O mesmo procedimento é seguido para os elementos restantes na coluna j (quando possível), i.e., calculamos $d_{i,j}$ de acordo com (3.6), com relação a $C^{(r,j)}(X)$ e $i \geq r+1$ até encontrarmos $d_{i,j} \neq 0$ para algum $r < i \leq M$. Se $d_{i,j} = 0$ para $1 \leq i \leq M$, então não temos mais discrepâncias não nulas e o polinômio $C^{(r,j)}(X)$ e j - 1 dão a solução para o problema e o procedimento termina aqui.

Iremos agora, a partir deste lema, deduzir o algoritmo iterativo fundamental (AIF) para a solução do problema geral. São requeridas duas tabelas D e C que guardarão as discrepâncias e os polinômios finais de cada coluna, respectivamente.

► **Algoritmo Iterativo Fundamental (AIF):**

1) $s \leftarrow 1$, $r \leftarrow 1$, $C^{(0,s)}(X) \leftarrow 1$;

2) Calcule $d_{r,s} = [C^{(r-1,s)}(X).a^{(r)}(X)]_s$;

3) se $d_{r,s} = 0$, então

   a) se $r = M$, então $\ell \leftarrow s\text{-}1$ e $C(X) \leftarrow C^{(r-1,s)}(X)$, fim;

   b) se $r \neq M$, então $C^{(r,s)}(X) \leftarrow C^{(r-1,s)}(X)$ e $r \leftarrow r+1$, volte ao 2);

4) se $d_{r,s} \neq 0$, então

   a) se existirem $d_{r,u} \in D$, para algum $1 \leq u < s$, e y tais que a equação $d_{r,s} - y \, . \, d_{r,u} = 0$ admita solução, então:

   $C^{(r-1,s)}(X) \leftarrow C^{(r-1,s)}(X) - y \, . \, C^{(u)}(X) \, . \, X^{s-u}$,

   volte ao 3a);

   b) caso não exista um valor $d_{r,u} \in D$, para algum $1 \leq u < s$, ou se a equação $d_{r,s} - y \, . \, d_{r,u} = 0$, não admitir solução, então deve-se fazer uma busca para testar se a coluna s é linearmente independente das anteriores;

   c) se a coluna s for linearmente independente das anteriores, então $d_{r,s}$ será guardado na Tabela D, $C^{(s)}(X) \leftarrow C^{(r-1,s)}(X)$ e $C^{(0,s+1)}(X) \leftarrow C^{(s)}(X)$ (já atualizado) e $C^{(s)}(X)$ é guardado em C; $s \leftarrow s+1$, $r \leftarrow 1$ e volte para 2)

Teorema 3.1 : O valor final s e $C^{(r-1,s)}(X)$ provenientes da aplicação do AIF é a solução do problema geral com o mínimo valor possível para s.

Prova : A partir do AIF, obtém-se s e $C^{(r-1,s)}(X)$ o qual é o último polinômio para a coluna s de acordo com o Lema 1, tal que $d_{M,1} = 0$ o que implica que $C^{(r-1,s)}(X) \, . \, a^{(i)}(X)]_s = 0$ para $i = 1, \ldots, M$. Portanto, as s primeiras colunas de A são linearmente dependentes. Por construção, a minimalidade de s está garantida. ■

Iremos agora ilustrar a aplicação do AIF através de um exemplo:

Exemplo 3.2 : Considere a matriz A sobre Z_{100}:

$$A = \begin{bmatrix} 36 & 12 & 1 & 5 & 11 & \cdots \\ 12 & 1 & 5 & 11 & 2 & \cdots \\ 1 & 5 & 11 & 37 & 15 & \cdots \\ 5 & 11 & 13 & 67 & 28 & \cdots \\ 11 & 6 & 14 & 34 & 40 & \cdots \end{bmatrix}$$

Seguindo os passos do algoritmo, temos:

- $r = 1,\ s = 1,\ C^{(0,1)}(X) = 1$

  $d_{1,1} = [1.(1 + 36X + 12X^2 + X^3 + ...)]_1 = 36 \neq 0$

  $\nexists\ d_{1,u} \in D$ tal que $1 \leq u < 1$. Portanto, $C^{(1)}(X) = 1$ e $d_{1,1}$ é guardado em D;

- $r = 1,\ s = 2,\ C^{(0,2)}(X) = 1$

  $d_{1,2} = [1.(1 + 36X + 12X^2 + X^3 + ...)]_2 = 12 \neq 0.$

  Para $u = 1$, $d_{1,u} = 36$ e $d_{1,2} - y \,.\, d_{1,u} = 0$ admite $y = 42$ como solução. Portanto, $C^{(1,2)}(X) = 1 + 58X$ e $r = 2$.

  $d_{2,2} = [(1 + 58X)(1 + 12X + X^2 + 5X^3 + ...)]_2 = 97 \neq 0.$

  $\nexists\ d_{2,u} \in D$ tal que $1 \leq u < 2$. Portanto, $C^{(2)}(X) = 1 + 58X$ e $d_{2,2}$ é guardado em D;

- $r = 1,\ s = 3,\ C^{(0,3)}(X) = 1 + 58X$

  $d_{1,3} = [(1 + 58X)(1 + 36X + 12X^2 + X^3 + ...)]_3 = 97 \neq 0.$

  Para $u = 1$, $d_{1,u} = 36$ mas $d_{1,3} - y \,.\, d_{1,u} = 0$ <u>não</u> admite solução. Portanto, $C^{(3)}(X) = 1 + 58X$ e $d_{1,3}$ é guardado em D;

- $s = 4,\ r = 1,\ C^{(0,4)}(X) = 1 + 58X$

  $d_{1,4} = [(1 + 58X)(1 + 36X + 12X^2 + X^3 + 5X^4 + ...)]_4 = 63 \neq 0.$

  Para $u = 3$, $d_{1,u} = 97$ e $d_{1,4} - y \,.\, d_{1,u} = 0$ admite $y = 79$ como solução. Portanto, $C^{(1,4)}(X) = 1 + 79X + 68X^2$ e $r = 2$;

  $d_{2,4} = [(1 + 79X + 68X^2)(1 + 12X + X^2 + 5X^3 + 11X^4 + ...)]_4 = 74$

Para $u = 2$, $d_{2,u} = 97$ e $d_{2,4} - y \cdot d_{2,u} = 0$ admite $y = 42$ como solução. Portanto, $C^{(2,4)}(X) = 1 + 79X + 26X^2 + 64X^3$.

Podemos verificar ainda que $d_{3,4} = d_{4,4} = d_{5,4} = 0$ e, portanto, a solução é dada por $\ell = s - 1 = 3$ e $C(X) = 1 + 79X + 26X^2 + 64X^3$, i.e., $c_1 = 79$, $c_2 = 26$ e $c_3 = 64$.

3.3. Generalização do Algoritmo de Berlekamp-Massey

Iremos agora refinar o Algoritmo Iterativo Fundamental para que o mesmo possa ser aplicado à síntese de LFSR's que geram multiseqüências. O algoritmo resultante deste refinamento é o que chamamos de Algoritmo de Berlekamp-Massey Generalizado.

Se compararmos as formas da matriz S do tipo Nt x N e da matriz A do tipo M x N, na Seção 3.1, perceberemos que S é uma forma particular de A e pode ser obtida atribuindo-se o valor

$$s_n^{(h)} \text{ a } a_{t(n-j)+h,j} ,$$

para $n = 1, 2, ..., N$, $h = 1, 2, ..., t$ e $1 \le j \le n$ e X's para o restante das entradas.

Ao aplicarmos o AIF para S, suponha que tenhamos completado de processar o elemento $s_n^{(t)}$ o qual corresponde a $a_{r-1,s}$ na linha $r - 1$ e coluna s, onde $r - 1 = t(n - s) + t$. Portanto, já teremos obtido $C^{(r-1,s)}(X) = 0$. Como

$$[C^{(r-1,s)}(X) \cdot s^{(h)}(X)]_k = 0 ,$$

para $h = 1, 2, ..., t$ e $k = s, s+1, ..., n$, então de acordo com (3.1), $s - 1$ e $C^{(r-1)}(X)$ especificam completamente o comprimento e o polinômio de conexões de um LFSR minimal que gera $s_1^{(h)}, s_2^{(h)}, ..., s_n^{(h)}$, para $h = 1, 2, ..., t$. Vamos denotar por $< \sigma^{(n,t)}(X), \ell_n^{(t)} > = < C^{(r-1,s)}(X), s - 1 >$ este LFSR minimal e faça $d_n^{(1)} = d_{r,s}$. Note que $d_n^{(1)} = [\sigma^{(n,t)}(X) \cdot s^{(1)}(X)]_{n+1}$

Seja $< \sigma^{(m_t,t)}(X), \ell^{(t)}_{m_t} >$ um LFSR minimal obtido pelo AIF o qual gera $s^{(h)}_1$, $s^{(h)}_2$,..., $s^{(h)}_{m_t}$, para $h = 1, 2, ..., t$, onde $1 \le m_t < n$, $d_{m_t} \neq 0$ e $d^{(1)}_{m_t} \in D$, é tal que $m_t - \ell^{(t)}_{m_t}$ tenha o máximo valor.

O próximo passo será então determinar $< \sigma^{(n+1,1)}(X), \ell^{(1)}_{n+1} >$, o qual gera $s^{(h)}_1$, $s^{(h)}_2$,..., $s^{(h)}_n$, para $h = 1, 2, ..., t$ e $s^{(1)}_{n+1}$. De acordo com o AIF, precisamos calcular $d_{r,s} = d^{(1)}_n$.

Se $d^{(1)}_n = 0$, então $d_{r,s} = 0$ e do AIF temos que $C^{(r,s)}(X) = C^{(r-1,s)}(X)$. Portanto:

$$\sigma^{(n+1,1)}(X) = \sigma^{(n,t)}(X) \quad \text{e} \quad \ell^{(1)}_{n+1} = \ell^{(t)}_n$$

Se $d^{(1)}_n \neq 0$, então isto significa que $d_{r,s} \neq 0$. Os teoremas a seguir ajudam a encontrar as próximas soluções.

Teorema 3.2 : Se $d^{(1)}_n = 0$, então

$$\sigma^{(n+1)}(X) = \sigma^{(n,t)} \quad \text{e} \quad \ell^{(1)}_{n+1} = \ell^{(t)}_n. \tag{3.10}$$

Se $d^{(1)}_n \neq 0$ e y é uma solução de $d^{(1)}_n - y.d^{(1)}_{m_t} = 0$, então

$$\sigma^{(n+1,1)}(X) = \sigma^{(n,t)} - y.\sigma^{(m_t,t)}(X).X^{n-m_t}$$

e

$$\ell^{(1)}_{n+1} = \max\left[\ell^{(t)}_n, n - m_t + \ell^{(t)}_{m_t}\right]. \tag{3.11}$$

Prova : A partir de AIF e do Lema 3.1. ■

De forma análoga, seja $< \sigma^{(n+1,h)}(X), \ell^{(h)}_{n+1} >$ um LFSR minimal obtido pelo AIF

capaz de gerar $s_1^{(h)}$, $s_2^{(h)}$,..., $s_n^{(h)}$, para $h = 1, 2, ..., t$ e $s_{n+1}^{(1)}$, $s_{n+1}^{(2)}$,..., $s_{n+1}^{(h)}$, para $h = 1, 2, ..., t-1$ e seja $d_n^{(h+1)} = [\sigma^{(n+1,h)}(X).s^{(h+1)}(X)]_{n+1}$ para $h = 1, 2, ..., t-1$.

Então, podemos determinar $< \sigma^{(n+1,\, h+1)}(X),\ \ell_{n+1}^{(h+1)} >$ de acordo com o seguinte teorema:

Teorema 3.3 : Se $d_n^{(h+1)} = 0$, então

$$\sigma^{(n+1,h+1)}(X) = \sigma^{(n+1,h)}(X) \quad \text{e} \quad \ell_{n+1}^{(h+1)} = \ell_{n+1}^{(h)}. \tag{3.12}$$

Se $d_n^{(h+1)} \neq 0$ e $\sigma^{(m_h+1,h)}(X)$ $(1 \leq m_h < n+1)$ com $d_{m_h}^{(h+1)} \neq 0$ é tal que $m_t - \ell_{m_h}^{(h)}$ tem o máximo valor e y é uma solução de $d_n^{(h+1)} - y.d_{m_h}^{(h+1)} = 0$, então

$$\sigma^{(n+1,h+1)}(X) = \sigma^{(n+1,h)}(X) - y.\sigma^{(m_h+1,h)}(X).X^{n-m_h}$$

e

$$\ell_{n+1}^{(h+1)} = \max\left[\ell_{n+1}^{(h)}, n - m_h + \ell_{m_h}^{(h)}\right]. \tag{3.13}$$

Prova : A partir de AIF e do Lema 3.1. ■

Baseados nestes Teoremas 3.2 e 3.3, iremos descrever o algoritmo para síntese de LFSR's minimais que geram multiseqüências prescritas de comprimento N.

Antes porém, como no caso de uma única seqüência (Capítulo 2), precisamos fixar as condições iniciais:

$$\sigma^{(-1,t)}(X) = 1, \quad \ell_{-1}^{(t)} = 0, \quad \sigma^{(0,h)}(X) = 1,$$

$$\ell_0^{(h)} = 0, \quad m_h = -1, \quad d_{-1}^{(h)} = 1.$$

para $h = 1, 2, ..., t$.

► **Algoritmo de Berlekamp-Massey Generalizado para Multiseqüências:**

1) $n \leftarrow 0$;

2a) Cálculo de $\sigma^{(n+1,1)}(X)$ a partir de $\sigma^{(n,t)}(X)$;

i) se $d_n^{(1)} = 0$, então $\sigma^{(n+1,1)}(X)$ e $\ell_{n+1}^{(1)}$ são dados por (3.10). Vá para 3);

ii) se $d_n^{(1)} \neq 0$, então encontre m_t tal que a Equação $d_n^{(1)} - y.d_{m_t}^{(1)} = 0$ tenha solução em y sobre o anel R. $\sigma^{(n+1,1)}(X)$ e $\ell_{n+1}^{(1)}$ são dados por (3.11);

iii) se $\ell_{n+1}^{(1)} = \max[\ell_n^{(1)},\ n+1-\ell_n^{(1)}]$, então vá para 3); caso contrário procure uma solução $D^{(n+1,t)}(X)$ com grau ℓ mínimo possível no intervalo $\max[\ell_n^{(1)},\ n+1-\ell_n^{(1)}] \leq \ell < \max[\ell_n,\ \ell_{m_t}^{(t)} + n - m_t]$ tal que o polinômio $\sigma^{(m_t,t)}(X)$ definido pela relação

$$D^{(n+1,t)}(X) - \sigma^{(n,t)}(X) = X^{n-m_t}.\sigma^{(m_t,t)}(X)$$

seja uma solução para as m_t primeiras somas de potência e tal que $d_{m_t}^{(1)} = -d_n^{(1)}$ e $\sigma_0^{(m_t)}$ seja um divisor de zero em R. Se este polinômio for encontrado, então

$$\sigma^{(n+1,t)}(X) \leftarrow D^{(n+1,t)}(X)$$

e

$$\ell_{n+1}^{(1)} \leftarrow \ell$$

2b) Cálculo de $\sigma^{(n+1,h+1)}(X)$ a partir de $\sigma^{(n+1,h)}(X)$, para $h = 1, 2, ..., t-1$;

i) se $d_n^{(h+1)} = 0$, então $\sigma^{(n+1,h+1)}(X)$ e $\ell_{n+1}^{(h+1)}$ são dados por (3.12). Vá para 3);

ii) se $d_n^{(h+1)} \neq 0$, então encontre m_h tal que a Equação $d_n^{(h+1)} - y.d_{m_h}^{(h+1)} = 0$ tenha solução em y sobre o anel R. $\sigma^{(n+1,h+1)}$ e $\ell_{n+1}^{(h+1)}$ são dados por (3.13);

iii) se $\ell_{n+1}^{(h+1)} = \max[\ell_{n+1}^{(h)},\ n+1-\ell_{m_h}^{(h)}]$, então vá para 3); caso contrário procure

uma solução $D^{(n+1,h)}(X)$ com grau ℓ mínimo possível no intervalo $\max[\ell_{n+1}^{(h)},\ n+1-\ell_{n+1}^{(h)}] \le \ell < \max[\ell_{n+1}^{(h)},\ n-m_h+\ell_{m_h}^{(h)}]$ tal que o polinômio $\sigma^{(m_h,h+1)}(X)$ definido pela relação

$$D^{(n+1,h)}(X) - \sigma^{(n+1,h)}(X) = X^{n-m_h}.\sigma^{(m_h,h+1)}(X)$$

seja uma solução para as m_h primeiras somas de potência e tal que $d_{m_h}^{(h+1)} = -d_n^{(h+1)}$ e $\sigma_0^{(m_h,h)}$ seja um divisor de zero em R. Se este polinômio for encontrado, então

$$\sigma^{(n+1,h)}(X) \leftarrow D^{(n+1,h)}(X)$$

e

$$\ell_{n+1}^{(h+1)} \leftarrow \ell$$

3) se $n < N-1$, então $d_{n+1}^{(l)} \leftarrow [\sigma^{(n+1,t)}(X).s^{(l)}(X)]_{n+2}$

4) $n \leftarrow n+1$; se $n < N$. Vá para 2); caso contrário, fim.Cálculo de $\sigma^{(n+1,1)}(X)$ a partir de $\sigma^{(n,t)}(X)$;

A resposta desejada será dada pelo polinômio $\sigma^{(N,t)}(X)$, i.e., os seus coeficientes formam uma solução para as Equações (3.1). Este algoritmo será usado na próxima seção quando da decodificação de certas classes de códigos cíclicos.

3.4. Decodificação de Códigos Cíclicos

Em princípio, o decodificador de Meggitt apresentado em [9] e [15] para a decodificação de códigos cíclicos sobre corpos GF(q) (onde q é uma potência de primo) também poderá ser aplicado diretamente, sem modificações, para a decodificação de códigos cíclicos definidos sobre anéis de inteiros residuais (Seção 1.2).

Entretanto, mostraremos aqui que é possível aplicar o algoritmo de BM (generalizado para multiseqüências) para a decodificação de certas classes de códigos

cíclicos sobre anéis Z_q (onde q é uma potência de primo) até o limitante de Hartmann-Tzeng (HT) [28].

Vamos fazer uma breve revisão a respeito deste limitante. Considere um código cíclico C de comprimento n sobre GF(q) gerado por g(x). Seja α uma n-ésima raiz primitiva da unidade em GF(q^m). Se $\alpha^{b+ic_1+hc_2}$ são raízes de g(x) para $i = 1, 2, ..., d_0 - 1$ e $h = 1, 2, ..., s+1$, onde $(c_1, n) = (c_2, n) = 1$, então

$$d_{min}(C) \geq d_{HT} = d_0 + s.$$

Note que quando $s = 0$, o limitante de HT reproduz o limitante dos códigos BCH. Essa é a generalização do limitante da distância mínima de códigos BCH para uma classe mais ampla de códigos cíclicos.

É possivel ainda mostrar que este limitante continua válido (sob as mesmas hipóteses) se considerarmos códigos cíclicos definidos sobre anéis Z_q. Neste caso, o corpo de extensão GF(q^m) é substituído por um anel de extensão de Z_q, i.e., GR(q, r) e α será uma raiz primitiva da unidade em R^*, o grupo multiplicativo dos elementos inversíveis em GR(q, r).

Seja então c(x) um polinômio código (de um código cíclico sobre Z_q, onde $q = p^k$), e(x) um polinômio erro e r(x) um vetor recebido tal que r(x) = c(x) + e(x). As síndromes $s_i^{(h)}$ são definidas por:

$$s_i^{(h)} = s_{b+ic_1+hc_2} = r(\alpha^{b+ic_1+hc_2}), \tag{3.14}$$

para $i = 1, 2, ..., d_0-1$ e $h = 1, 2, ..., s+1$.

Como $c(\alpha^{b+ic_1+hc_2}) = 0$, então

$$s_i^{(h)} = e(\alpha^{b+ic_1+hc_2}) \tag{3.15}$$

para $i = 1, 2, ..., d_0-1$ e $h = 1, 2, ..., s+1$.

Portanto, temos $s+1$ seqüências de síndromes, cada uma de comprimento d_0-1.

Seja agora ν o número de erros ocorridos e a_μ e Y_μ, para $\mu = 1, 2, ..., \nu$, as

localizações e os valores dos erros, onde $\nu = \lfloor d_0 + s - 1)/2 \rfloor$, $0 \le a_\mu < n$ e $Y_\mu \in Z_q$. Então,

$$e(x) = \sum_{\mu=1}^{\nu} Y_\mu . x^{a_\mu} \tag{3.16}$$

Seja $X_\mu = \alpha^{a_\mu}$. Então, de (3.15) e (3.16) temos:

$$s_i^{(h)} = \sum_{\mu=1}^{\nu} Y_\mu . X_\mu^{b+ic_1+hc_2} \tag{3.17}$$

para $i = 1, 2, ..., d_0 - 1$ e $h = 1, 2, ..., s+1$.

O problema da decodificação de códigos cíclicos consiste então em determinar os Y_μ's e X_μ's a partir das seqüências $\{s_i^{(h)}\}$ e tal que (3.16) seja satisfeito com o mínimo valor de ν. Seja

$$\sigma(X) = \prod_{\mu=1}^{\nu} (X - X_\mu^{c_1}) = X^\nu + \sigma_1 X^{\nu-1} + ... + \sigma_{\nu-1} X + \sigma_\nu \tag{3.18}$$

o polinômio localizador de erros. Agora, usando argumentos análogos aos usados na Seção 2.2, podemos mostrar que:

$$s_i^{(h)} + \sigma_1 . s_{i-1}^{(h)} + ... + \sigma_{\nu-1} . s_{i-\nu+1}^{(h)} + \sigma_\nu . s_{i-\nu}^{(h)} = 0, \tag{3.19}$$

para $i = \nu+1, \nu+2, ..., d_0 - 1$ e $h = 1, 2, ..., s+1$.

Portanto, para localizar a posição dos erros precisamos resolver o Sistema de Equações (3.19) nas incógnitas $\sigma_1, \sigma_2, ..., \sigma_\nu$ com o menor valor de ν possível. Esse sistema será resolvido aplicando-se o algoritmo de BM generalizado para multiseqüências. Por um argumento um pouco mais elaborado que o usado no Apêndice 2.1 podemos mostrar que a solução do Sistema (3.19) é única se, e somente se, os Y_i (as magnitudes dos erros) forem elementos inversíveis em Z_{p^k}, para $1 \le i \le \nu$.

Caso esse sistema apresente mais do que uma solução, então teremos vários polinômios localizadores de erro $\sigma(Z)$. Sabemos da Seção 2.2 (Passo 3) que basta escolher um deles para obtermos os corretos números de localização de erro $X_i^{c_1}$, $1 \le i \le \nu$. Lembre que as raízes Z_i ($1 \le i \le \nu$) de $\rho(Z)$ (o polinômio recíproco de $\sigma(Z)$) estão relacionadas com os corretos números de localização de erro através de:

$$Z_i - X_i^{c_1} = \text{divisor de zero em } GR(p^k, r)$$

e são determinados de forma única.

Quanto ao cálculo da magnitude dos erros, o procedimento proposto por Forney (Seção 2.2, Passo 4) continua válido, com pequenas alterações. Pode-se mostrar que:

$$Y_j = \frac{\sum_{i=0}^{\nu-1} \sigma_{j\ell}\, s_{\nu-\ell}^{(h)}}{X_j^{b+c_2 h} \cdot \sum_{\ell=0}^{\nu-1} \sigma_{j\ell} \cdot X_j^{c_1(\nu-\ell)}} \qquad (3.20)$$

para $1 \le j \le \nu$, e onde os $\sigma_{j\ell}$ são obtidos recursivamente a partir de X_i e σ_i através da relação:

$$\sigma_{ji} = \sigma_i + X_j^{c_1} \cdot \sigma_{j,i-1} \qquad (3.21)$$

(o valor h que aparece em (3.20) pode ser qualquer número entre 1 e s+1).

Com isto, concluímos a decodificação de códigos cíclicos sobre anéis Z_q, até o limitante de HT.

3.5. Conclusões

Neste capítulo foi apresentado o problema da determinação do menor conjunto inicial de colunas linearmente dependentes em uma matriz A do tipo M x N, definida sobre

um anel comutativo R com identidade, i.e., encontrar a coluna de A de menor índice que pode ser expressa como combinação linear das anteriores. Este problema foi resolvido de forma iterativa através do uso de um algoritmo chamado Algoritmo Iterativo Fundamental (AIF).

Vimos que o problema da síntese de LFSR's que geram multiseqüências é um caso especial deste problema mais geral da determinação do menor conjunto inicial de colunas linearmente dependentes em uma matriz e, portanto, apresenta certas particularidades inerentes as quais nos possibilitaram refinar o AIF, tornando-o menos complexo para poder ser aplicado. Esse refinamento levou à generalização do algoritmo de BM para multiseqüências.

Finalmente apresentamos uma aplicação deste algoritmo de BM generalizado para a decodificação de códigos cíclicos sobre anéis Z_q até o limitante de Hartmann-Tzeng (HT), onde múltiplas seqüências de síndrome estão disponíveis, ao invés de uma única como no caso de códigos Reed-Solomon e BCH. Foi mostrado entretanto que os fundamentos da decodificação continuam praticamente os mesmos (dos códigos Reed-Solomon e BCH) e inclusive o passo referente à determinação das magnitudes dos erros pode ainda ser implementado pelo procedimento proposto por Forney (para códigos Reed-Solomon e BCH), com pequenas modificações.

CAPÍTULO 4

CÓDIGOS LINEARES SOBRE GRUPOS

Neste capítulo será feito um estudo a respeito de códigos sobre grupos, baseado na teoria apresentada em [29]. Começaremos apresentando as definições básicas e alguns resultados da teoria de grupos (Seções 4.1, 4.2 e 4.3) que serão úteis durante o desenvolvimento.

A partir disto será mostrado que códigos definidos sobre grupos não-abelianos apresentam baixas distâncias de Hamming e têm um comportamento assintoticamente ruim.

Como conseqüência deste fato, iremos focalizar o estudo na direção de grupos abelianos. A construção proposta em [29], baseada no Teorema da Base Normal [32], mostra que o problema da determinação de códigos lineares sobre um grupo abeliano G pode ser tratado como um problema semelhante ao da determinação de códigos lineares sobre o anel Z_m (o conjunto dos inteiros módulo m), onde m depende de certas propriedades estruturais de G.

A seguir mostraremos que a construção de códigos sobre grupos abelianos (baseados nesta proposta) que atinjam desempenho equivalente ao de códigos sobre corpos e anéis é, em geral, mais custosa que as construções utilizadas para a obtenção destes últimos. Entretanto, ela mostra-se interessante do ponto de vista teórico, no sentido de que é útil quando da determinação de limites de desempenho.

Finalmente faremos então uma proposta alternativa, através da qual os códigos lineares sobre grupos abelianos serão obtidos via concatenação generalizada de códigos sobre corpos e/ou anéis de inteiros residuais. Será indicado ainda como esta construção poderá ser utilizada em esquemas de modulação codificada, códigos de classes laterais e empacotamentos esféricos.

4.1. Definições Básicas

Seja G um grupo finito de ordem |G| cuja operação é escrita na forma multiplicativa e cujo elemento identidade é denotado por e_G.

Definição 4.1 : Um **código de bloco** C de **comprimento** n sobre G é um subconjunto não vazio do produto direto G^n, i.e., do conjunto de todas as n-uplas de elementos de G. Diz-se que G é o **alfabeto** sobre o qual o código está sendo construído.

Uma conseqüência imediata desta Definição 4.1 é que o número de palavras do código fica limitado a $|G|^n$.

Definição 4.2 : A **dimensão** de um código de bloco C de comprimento n sobre G é dada por $k = \log_{|G|} |C|$ e a sua taxa por $r = k/n$.

Vamos agora associar a cada componente das n-uplas de C um índice pertencente a um conjunto I de cardinalidade n (o comprimento do código). Temos então a seguinte

Definição 4.3 : Um **conjunto de informação** de C é qualquer subconjunto de índices $J \subseteq I$ de cardinalidade $| J | = k$ (a dimensão do código) tal que toda k-upla de elementos de G ocorre em J precisamente uma única vez ao percorrermos todas as palavras de C.

Obs.: Como k pode assumir um valor não inteiro, segue que existem códigos que não possuem um conjunto de informação.

Definição 4.4 : Um código de bloco C é **linear** sobre G se C for um subgrupo de G^n, o produto de G por ele mesmo *n* vezes.

Definição 4.5 : Uma **distância** sobre G é um mapeamento d : G x G → ℝ (o conjunto dos números reais) que verifica as seguintes propriedades:

i) $d(g, h) \geq 0$;

ii) $d(g, h) = 0 \Leftrightarrow g = h$;

iii) d(g, h) depende somente da diferença $g.h^{-1}$ entre g e h;

iv) d(g, h) = d(h, g)

Uma conseqüência da Definição 4.5 é que a distância *d* é invariante por translação à direita, i.e., para quaisquer *g*, *h*, *s* ∈ G temos que $d(g.s, h.s) = d(g, h)$, (se G for abeliano, *d* é invariante por translação à direita e à esquerda, i.e., $d(g.s, h.s) = d(s.g, s.h) = d(g, h)$).

Podemos agora definir uma distância aditiva sobre o produto direto G^n como sendo a soma das distâncias entre as respectivas componentes. Mais especificamente, se $\underline{g} = (g_1, g_2, \ldots, g_n)$ e $\underline{h} = (h_1, h_2, \ldots, h_n)$ são duas *n*-uplas de elementos de G, então:

$$d(\underline{g}, \underline{h}) = \sum_{i=1}^{n} d(g_i, h_i)$$

Um importante exemplo é a distância de Hamming, a qual é definida através de:

$$d(g,h) = \begin{cases} 0 & \text{se } g.h^{-1} = e_G \\ 1 & \text{se } g.h^{-1} \neq e_G \end{cases}$$

Um outro exemplo é a distância Euclidiana a qual não faremos uso neste capítulo. Seja R(G) uma representação de G (uma ***representação matricial*** de grau *n* de um grupo G é um homomorfismo $\phi : g \rightarrow \phi(g)$ de G em GL(*n*, K), o grupo multiplicativo das matrizes inversíveis do tipo *n* x *n* definidas sobre um corpo K [32]) por matrizes reais e ortogonais *n* x *n* e X ∈ $\mathbb{R}^n$ um vetor inicial tal que a ação de R(G) sobre X gere um

conjunto de |G| vetores. Definimos então a distância Euclidiana entre dois elementos g e h pertencentes a G como:

$$d(g,h) = \|R(g).X - R(h).X\|^2 ,$$

onde $\|\cdot\|$ denota a norma Euclidiana. Estes códigos são conhecidos como ***Códigos do espaço Euclidiano***; para maiores detalhes, veja [2], [30] e [31].

Com esta seção introduzimos as definições e notações que serão necessárias durante a exposição da teoria. Vamos agora na Seção 4.2 apresentar uma primeira idéia de construção de códigos sobre grupos.

4.2. Geração de Códigos sobre Grupos: Uma Primeira Aproximação

Vamos recordar do Capítulo 1 que a construção de cada palavra de um código linear sobre um corpo ou anel é feita concatenando-se uma k-upla de elementos, que são os símbolos de informação, a uma $(n-k)$-upla de elementos que são os símbolos de paridade. Os símbolos de informação podem ser escolhidos livremente dentro do alfabeto que está sendo usado, enquanto que os símbolos de paridade são escolhidos de forma tal que certas equações lineares conhecidas como equações de verificação de paridade, sejam satisfeitas pelo conjunto de todos os símbolos da palavra código em questão.

Considere agora um conjunto de relações da forma:

$$g_1^{\ell_{j1}} \cdot g_2^{\ell_{j2}} \cdots g_n^{\ell_{jn}} = e_G , \qquad (4.1)$$

para $1 \le j \le s$, onde os g_i's são elementos de G e os expoentes ℓ_{ji} ($1 \le j \le s$ e $1 \le i \le n$) são inteiros entre 0 e o expoente de G, inclusive (o ***expoente*** de G é o menor inteiro e tal que $x^e = e_G$ para todo $x \in G$; mostra-se que e é o mínimo múltiplo comum das ordens dos elementos de G).

Podemos então construir um código de bloco C sobre G da seguinte forma: *uma n-upla de elementos de G, $\underline{g} = (g_1, g_2, \ldots, g_n)$, pertence a C se, e somente se, as relações*

(4.1) se verificam. A partir disto podemos definir uma "matriz verificação de paridade" H para C que tem como entradas os expoentes ℓ_{ji} de (4.1):

$$H=\begin{bmatrix} \ell_{11} & \ell_{12} & \cdots & \ell_{1n} \\ \ell_{21} & \ell_{22} & \cdots & \ell_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \ell_{s1} & \ell_{s2} & \cdots & \ell_{sn} \end{bmatrix} \tag{4.2}$$

Exemplo 4.1 : Seja C um código tal que $n = 3$, $k = 2$ e definido pela relação $g_1.g_2.g_3 = e_G$, i.e., a "matriz verificação de paridade" é dada por H = [1 1 1]. Este código tem distância mínima de Hamming igual a 2 e é conhecido como *código de verificação de paridade*.

Exemplo 4.2 : A "matriz verificação de paridade"

$$H=\left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & -1 \end{array}\right]$$

define um código de Hamming com $n = 7$ e $k = 4$. Por inspeção direta, podemos concluir que a distância mínima de Hamming vale 3.

Mostra-se que os códigos dos Exemplos 4.1 e 4.2 são lineares se, e somente se, os grupos sobre os quais os mesmos estão definidos forem abelianos.

<u>*Obs. importantes*</u> :

1) A "matriz verificação de paridade" (4.2), ao contrário das matrizes verificação de paridade de códigos definidos sobre corpos e anéis, não provê informações sobre a distância mínima de Hamming do código definido por ela;

2) Para grupos não abelianos, o arranjo dos elementos nas Equações 4.1 é importante já que em geral $g.h.g \neq g^2.h$. Com isto, concluímos que não se pode usar um procedimento baseado em matrizes verificação de paridade para definir códigos sobre grupos em geral (especialmente os não abelianos). Apesar disto, veremos mais tarde que para grupos abelianos este é o procedimento a ser seguido.

Portanto, a fim de não perdermos generalidade, diremos que *cada palavra de um código sobre um grupo G (abeliano ou não) é formada a partir da concatenação de k símbolos de informação* $(x_1, x_2, ..., x_k)$ *com* $(n-k)$ *símbolos de paridade* $(y_1, y_2, ..., y_{n-k})$ *através das relações*:

$$y_i = \phi_i (x_1, x_2, ..., x_k) \quad ,$$

para $1 \leq i \leq n-k$ e onde os ϕ_i's são mapeamentos de G^k em G. Temos então a seguinte

Definição 4.6 : Um (n, k)-***código de bloco sistemático*** C de comprimento n e dimensão k sobre um grupo G é um subconjunto de G^n com ordem $|G|^k$ formado pelas n-uplas

$$(x_1, x_2, ..., x_k, y_1, y_2, ..., y_{n-k}) \tag{4.3}$$

tal que:

$$y_i = \phi_i(x_1, x_2, ..., x_k) \quad , \tag{4.4}$$

onde os ϕ_i's $(1 \leq i \leq n-k)$ são mapeamentos de G^k em G.

Apresentaremos agora uma proposição a qual dará a condição de linearidade para o código sistemático.

Proposição 4.1 : O (n, k)-código sistemático com as palavras código (4.3) é linear se, e somente se, os mapeamentos (4.4) são homomorfismos de G^k em G.

Prova : Considere duas palavras-código

$$\underline{x} = (x_1, x_2, ..., x_k, y_1, ..., y_{n-k})$$
$$\underline{t} = (t_1, t_2, ..., t_k, z_1, ..., z_{n-k})$$

e seu produto:

$$\underline{x} \cdot \underline{t} = (x_1.t_1, x_2.t_2, ..., x_k.t_k, y_1.z_1, ..., y_{n-k}.z_{n-k});$$

Para linearidade, devemos ter:

$$y_i \cdot z_i = \phi_i(x_1.t_1, ..., x_k.t_k) = \phi_i(x_1, ..., x_k) \cdot \phi_i(t_1, ..., t_k)$$

para $1 \leq i \leq n-k$, o que caracteriza os ϕ_i's como homomorfismos de G^k em G. ■

O exemplo a seguir terá papel relevante quando formos mostrar que códigos sobre grupos não abelianos não têm boas propriedades de distância.

Exemplo 4.3 : Um (n, 1)-***código de repetição*** sobre um grupo G é definido como o conjunto de |G| palavras-código da forma:

$$(g \mid \theta_2(g)\ \theta_3(g)\ \ldots\ \theta_n(g))$$

para todo $g \in G$ e onde os θ_i's são ***endomorfismos*** (homomorfismos de um grupo nele próprio) de G. Se todos os θ_i's forem ***automorfismos*** (isomorfismos de um grupo nele próprio), o código será chamado de código de repetição ***automórfico***. Observe que um código de repetição terá distância mínima de Hamming igual a n se, e somente se, ele for automórfico. Isto é devido ao fato de que se algum θ_i for um endomorfismo próprio (i.e., um endomorfismo que não é um automorfismo), então existe $g \neq e_G$ tal que $\theta_i(g) = e_G$. Portanto, a palavra código formada a partir deste elemento g teria ao menos uma posição com o elemento e_G e assim a distância de Hamming seria no máximo n - 1.

A proposição a seguir mostra que todos os (n, k) códigos sistemáticos e lineares sobre um grupo G são isomorfos a G^k. Isto ilustra o fato de que a estrutura algébrica do código em si não é o fator relevante, já que diferentes códigos, apesar de isomorfos, podem não ter as mesmas propriedades de distância.

Proposição 4.2 : Todos os (n, k)-códigos sistemáticos e lineares sobre o mesmo grupo G são isomorfos a G^k.

Prova : Qualquer código sistemático é isomorfo ao código linear G^k x $\{e_G\}^{n-k}$. De fato, a projeção

$$\pi : \underline{g} = (g_1, g_2, \ldots, g_n) \rightarrow (g_1, g_2, \ldots, g_k, e_G, e_G, \ldots, e_G)$$

é um isomorfismo porque ela é bijetora e preserva a operação de grupo:

$$\pi(\underline{g}.\underline{g}') = \pi(\underline{g}).\pi(\underline{g}') = (g_1.g_1', \ldots, g_k.g_k', e_G, e_G, \ldots, e_G).$$ ■

4.3. Algumas Propriedades sobre Fatoração de Grupos

Na próxima seção iremos mostrar que códigos sobre grupos não abelianos não têm boas propriedades de distância. O argumento a ser usado baseia-se no fato de que tais códigos (sob certas condições) são obtidos pela "combinação" de subcódigos menores (i.e., cada palavra $\underline{c} \in C$ pode ser escrita como $\underline{c} = \underline{c}_1.\underline{c}_2 \ldots \underline{c}_m$, onde cada palavra $\underline{c}_i$ pertence a um subcódigo C_i de C ($1 \leq i \leq m$)) e como conseqüência, a distância mínima de Hamming do mesmo fica limitada pela menor das distâncias mínimas de Hamming desses subcódigos, os quais mostraremos ser do tipo repetição (veja o Exemplo 4.3). Os conceitos e teoremas que iremos rever nesta seção serão importantes durante a demonstração destes fatos.

Seja G um grupo e H e K subgrupos próprios de G. G é chamado ***fatorável*** se G = H.K e H e K comutam elemento a elemento (i.e., todo $g \in G$ pode ser escrito na forma $g = h.k = k.h$, com $h \in H$ e $k \in K$). Se além disso, $H \cap K = \{e_G\}$, então diremos que G é ***decomponível***. G é dito ser o ***produto direto*** de H e K (onde $H \cap K = e_G$) se seus elementos são pares (h, k), com $h \in H$ e $k \in K$ (ver Definição 1.13). Uma propriedade decorrente destes conceitos é que qualquer grupo G que é um produto direto é fatorável, mas o contrário não é necessariamente válido. Entretanto, se G é decomponível, então ele é isomorfo a um produto direto. O ***centro*** do grupo G é o conjunto:

$$Z(G) = \{x \in G : x.a = a.x \quad \text{para todo } a \in G\}$$

e uma conseqüência imediata é que Z(G) é um subgrupo normal de G.

Apresentaremos agora uma proposição que descreve algumas propriedades acerca de grupos fatoráveis e decomponíveis.

Proposição 4.3 : Seja G um grupo fatorável, com $G = G_1.G_2$. Então:

a) G_1 e G_2 são subgrupos normais de G;

b) $H = G_1 \cap G_2$ é um subgrupo central, i.e., H é um subgrupo de Z(G), o centro de G;

c) Se G_1 e G_2 são abelianos, então G é decomponível.

Prova : a) Vamos provar que G_1 é normal e daí a prova para G_2 é análoga. Seja $w \in G_1$ e $g = g_1.g_2 \in G$. Então:

$g^{-1}.w.g = g_2^{-1}.g_1^{-1}.w.g_1.g_2 = g_1^{-1}.w.g_1 \in G_1$,

o que prova que G_1 é normal;

b) Os elementos de H comutam com todos os elementos de G e, portanto, pertencem ao centro de G;

c) Se G_1 e G_2 são abelianos, então G é abeliano e, portanto, decomponível (quando formos estudar códigos sobre grupos abelianos, veremos que todo grupo abeliano pode ser escrito como um produto direto de determinados subgrupos cíclicos). ■

Proposição 4.4 : A distância mínima de Hamming de um código fatorável $C = C_1.C_2$ não pode exceder a menor das distâncias mínimas de Hamming de C_1 e C_2.

Prova : Sejam $\underline{c}_1$ e $\underline{c}_2$ as palavras de menor peso em C_1 e C_2, respectivamente. As palavras $\underline{c} = \underline{c}_1.\underline{e}_G$ e $\underline{c}' = \underline{e}_G.\underline{c}_2$ (onde $\underline{e}_G = (e_G, e_G, ..., e_G)$) pertencem a C e têm peso de Hamming igual a $d_{min}(C_1)$ e $d_{min}(C_2)$, respectivamente. Portanto, $d_{min}(C) \leq \min\{d_{min}(C_1), d_{min}(C_2)\}$. ■

Proposição 4.5 : Um código C linear e sistemático sobre um grupo fatorável é fatorável.

Prova : Por definição, $G = G_1.G_2$. Então $G^{\ell} = G_1^{\ell}.G_2^{\ell}$, para qualquer $\ell \geq 1$. Tome agora $\underline{x} \in G^k$. Daí teremos que $\underline{x} = \underline{x}_1.\underline{x}_2$, com $\underline{x}_1$ e $\underline{x}_2$ ambos em G^k. Sendo ϕ um homomorfismo, temos:

$$(\underline{x} \mid \phi(\underline{x})) = (\underline{x}_1.\underline{x}_2 \mid \phi(\underline{x}_1.\underline{x}_2)) = (\underline{x}_1.\underline{x}_2 \mid \phi(\underline{x}_1).\phi(\underline{x}_2)) = (\underline{x}_1 \mid \phi(\underline{x}_1)).(\underline{x}_2 \mid \phi(\underline{x}_2))$$

o que prova que C é fatorável. ■

Proposição 4.6 : Seja $\mu : G \times G \rightarrow G$ uma operação binária de G, i.e., $\mu(a, b) = ab$. Se G x G for visto como um grupo, então μ é um homomorfismo se, e somente se, G é abeliano.

Prova : Do fato de que $\mu(a, b) = ab$ (para todo $a, b \in G$) e $\mu(c, d) = cd$ (para todo $c, d \in G$) temos que $\mu(ac, bd) = \mu(a, b).\mu(c, d) \Leftrightarrow acbd = abcd \Leftrightarrow cb = bc$ (esta última implicação segue da lei do cancelamento para grupos). E isto mostra que G é abeliano. ■

Proposição 4.7: Todo homomorfismo

$$\phi : G \times G \times \ldots \times G \rightarrow G$$

de G^k em G admite a decomposição canônica

$$\phi(x_1, \ldots, x_k) = \prod_{j=1}^{k} \phi(e_G, \ldots, e_G, x_j, e_G, \ldots, e_G)$$

onde os fatores no produto podem ser tomados em qualquer ordem, i.e., as imagens dos diferentes fatores comutam.

Prova : Segue direto da definição de homomorfismo de grupos. ■

Recorde que um (n, k)-código linear e sistemático é caracterizado por $(n\text{-}k)$ homomorfismos $\phi_i : G^k \rightarrow G$ e cada um deles é caracterizado por k endomorfismos $\varphi_h : G \rightarrow G$ tais que para $h \neq j$,

$$\varphi_h(G).\varphi_j(G) = \varphi_j(G).\varphi_h(G),$$

elemento a elemento. O grupo

$$W = \varphi_1(G).\varphi_2(G) \ldots \varphi_k(G)$$

é um subgrupo de G.

Considere agora um dos homomorfismos ϕ_i. A seguinte proposição é válida.

Proposição 4.8 : Seja G um grupo não abeliano. Se ϕ mapeia G^k em G de forma sobrejetora e é escrito na forma

$$\phi(g_1, ..., g_k) = \varphi_1(g_1) ... \varphi_k(g_k)$$

e se G for não-fatorável, então um dos φ_i's é um automorfismo e os restantes são endomorfismos tais que $\varphi_j(G) \subset Z(G)$, o centro de G.

Prova : Como ϕ é sobrejetor, segue que

$$G = \prod_{j=1}^{k} \varphi_j(G) = \varphi_\ell(G).\omega_\ell$$

para algum ℓ e ainda mais ω_ℓ e $\varphi_\ell(G)$ comutam elemento a elemento (pela Proposição 4.7). Como G não é fatorável, então ou $\varphi_\ell(G) = G$ e $\omega_\ell \subseteq Z(G)$ ou $\varphi_\ell(G) \subseteq Z(G)$ e $\omega_\ell = G$. Na primeira hipótese, temos que φ_ℓ é um automorfismo. Na segunda hipótese, repete-se o mesmo argumento para o fator ω_ℓ até que o automorfismo seja encontrado. ■

4.4. Códigos Lineares sobre Grupos Não Abelianos são Assintoticamente Ruins

Baseados nos resultados da Seção 4.3 iremos agora provar o seguinte:

Teorema 4.1 : Sejam C um código linear sobre um grupo não abeliano G. Suponha que as suas palavras-código tenham a forma

$$(x_1, x_2, ..., x_k, y_1, y_2, ..., y_{n-k})$$

onde

$$y_i = \phi_i(x_1, x_2, ..., x_k)$$

e que cada ϕ_i seja um homomorfismo sobrejetor de G^k em G, para $1 \leq i \leq n-k$. Então existem palavras de C que podem ser escritas como concatenação de palavras pertencentes a códigos tipo repetição.

Prova : Primeiramente suponha que G seja não fatorável. Da Proposição 4.8 segue então que

$$y_i = \phi_i(e_G, \ldots, e_G, x_{\ell(i)}, e_G, \ldots, e_G).Z_i$$

onde ϕ_i é um automorfismo de G, $\ell(i)$ é um índice que varia de 1 a k e

$$Z_i = \prod_{h=1, h\neq\ell(i)}^{k} \phi_i(e_G, \ldots, e_G, x_h, e_G, \ldots, e_G)$$

pertence a Z(G). Para simplificação na notação, colocamos:

$$\theta_i(x_{\ell(i)}) = \phi_i(e_G, \ldots, e_G, x_{\ell(i)}, e_G, \ldots, e_G).$$

Como conseqüência disto tudo, uma palavra-código qualquer assume a forma:

$$(x_1, x_2, \ldots, x_k, \theta_1(x_{\ell(1)}) \cdot Z_1, \ldots, \theta_{n-k}(x_{\ell(n-k)}) \cdot Z_{n-k})$$

Considere agora duas palavras código, $\underline{x}$ e $\underline{t}$, e o comutador $(\underline{x}.\underline{t}) \cdot (\underline{t}.\underline{x})^{-1}$ de ambas:

$$\underline{x} = (x_1, x_2, \ldots, x_k, \theta_1(x_{\ell(1)}) \cdot Z_1, \ldots, \theta_{n-k}(x_{\ell(n-k)}) \cdot Z_{n-k})$$

$$\underline{t} = (t_1, t_2, \ldots, t_k, \theta_1(t_{\ell(1)}) \cdot U_1, \ldots, \theta_{n-k}(t_{\ell(n-k)}) \cdot U_{n-k})$$

com $Z_i, U_i \in Z(G)$. Temos então:

$$(\underline{x}.\underline{t}) \cdot (\underline{t}.\underline{x})^{-1} = ((x_1.t_1) \cdot (t_1.x_1)^{-1}, (x_2.t_2) \cdot (t_2.x_2)^{-1}, \ldots, (x_k.t_k) \cdot (t_k.x_k)^{-1},$$

$$\theta_1((x_{\ell(1)}.t_{\ell(1)}) \cdot (t_{\ell(1)}.x_{\ell(1)})^{-1}) (Z_1.U_1) \cdot (U_1.Z_1)^{-1}, \ldots,$$

$$\theta_{n-k}((x_{\ell(n-k)}.t_{\ell(n-k)}) \cdot (t_{\ell(n-k)}) \cdot x_{\ell(n-k)})^{-1}) (Z_{n-k}.U_{n-k}) \cdot (U_{n-k}.Z_{n-k})^{-1})$$

Agora para os elementos em Z(G), vale que:

$$(Z_j.U_j) \cdot (U_j.Z_j)^{-1} = e_G$$

para $1 \leq j \leq n\text{-}k$. E, portanto, o comutador $(\underline{x}.\underline{t}) \cdot (\underline{t}.\underline{x})^{-1}$, que é também uma palavra código, pode ser escrito na forma:

$$(\underline{x}.\underline{t}) \cdot (\underline{t}.\underline{x})^{-1} = ((x_1.t_1) \cdot (t_1.x_1)^{-1}, (x_2.t_2) \cdot (t_2.x_2)^{-1}, \ldots, (x_k.t_k) \cdot (t_k.x_k)^{-1},$$

$$\theta_1((x_{\ell(1)}.t_{\ell(1)}) \cdot (t_{\ell(1)}.x_{\ell(1)})^{-1}), \ldots, \theta_{n-k}((x_{\ell(n-k)}.t_{\ell(n-k)}) \cdot (t_{\ell(n-k)}) \cdot x_{\ell(n-k)})^{-1}))$$

Suponha agora que G seja fatorável como G = H.K, com H não fatorável e não abeliano. Considere então o subcódigo sobre H (veja a Proposição 4.5). O mesmo argumento acima se aplica e este subcódigo também contém palavras de códigos tipo repetição. ■

Vamos agora fornecer um limitante superior para a distância mínima de Hamming destes códigos sobre grupos não abelianos. Para isto, vamos considerar a palavra-código

$(\underline{x}.\underline{t})$. $(\underline{t}.\underline{x})^{-1}$ como acima. Repare que ela é formada por "sub-palavras" de códigos tipo repetição. Apesar de não conhecermos o comprimento de cada uma dessas "sub-palavras" (pois não conhecemos precisamente os índices $\ell(i)$, podemos dizer que na melhor das hipóteses cada uma delas teria o mesmo comprimento $\lfloor n/k \rfloor$. Lembre que em um código tipo repetição de comprimento n, a distância mínima de Hamming é limitada superiormente por n (veja o Exemplo 4.3). Dessa forma, podemos dizer que o peso desta palavra (composta de "sub-palavras") está limitado superiormente por $\lfloor n/k \rfloor$. Portanto, o limitante superior da distância mínima de Hamming de um código linear sistemático sobre um grupo não abeliano é dado por:

$$d_{min} \leq \lfloor n/k \rfloor .$$

Fazendo $r = k/n$ e $\delta = d_{min}/n$, obtemos :

$$r.\delta \leq 1/n$$

o que mostra que o produto $r.\delta$ (e conseqüentemente, δ) vai a zero à medida que n cresce, para r fixo. Portanto, para todo código sobre um grupo não abeliano satisfazendo as hipóteses do Teorema 4.1, a razão $\delta = d_{min}/n$ é assintoticamente zero. É então devido a isto que passaremos agora a direcionar o estudo somente para códigos sobre grupos abelianos.

4.5. Construção de Códigos sobre Grupos Abelianos (Proposta I)

Iniciaremos nesta seção o estudo dos códigos sobre grupos abelianos. Apresentaremos primeiramente a construção proposta em [29], a qual baseia-se em duas propriedades fundamentais:

1) a de que a linearidade de códigos sistemáticos sobre grupos é garantida se as funções ϕ que mapeiam os símbolos de informação nos símbolos de paridade forem homomorfismos, conforme mostrado pela Proposição 4.1;

2) o fato de que todo grupo abeliano finito pode ser escrito como o produto direto de grupos cíclicos cujas ordens são divisíveis umas pelas outras. Cada um destes grupos

cíclicos é, por sua vez, isomorfo ao grupo aditivo de um anel Z_m (para algum $m \geq 2$), o anel de inteiros módulo m.

A partir daí serão usadas algumas propriedades básicas sobre homomorfismos de grupos abelianos para se deduzir uma matriz verificação de paridade para os códigos. Esta matriz contém informação sobre os elementos geradores dos grupos cíclicos e é formada por elementos pertencentes a anéis de inteiros residuais. Será mostrado que a mesma possui também, como no caso de corpos e anéis, a informação sobre a distância mínima de Hamming do código sobre o grupo abeliano.

4.5.1. Revisão de propriedades básicas de grupos abelianos

Propriedade 4.1 [32] : Todo grupo abeliano finito G pode ser escrito como o produto direto de subgrupos cíclicos cujas ordens são divisíveis umas pelas outras:

$$G = H_{d_1} \times H_{d_2} \times \ldots \times H_{d_m} \tag{4.5}$$

onde $d_1 \mid d_2 \mid \ldots \mid d_m$; d_m é o expoente do grupo e $a^{d_m} = e_G$ para qualquer $a \in G$.

Corolário 4.1 : Seja g_i um gerador de H_{d_i}, $i = 1, \ldots, m$. Então, todo elemento de G tem a forma:

$$g = g_1^{\alpha_1} \cdot g_2^{\alpha_2} \cdots g_m^{\alpha_m},$$

para $0 \leq \alpha_i \leq d_i - 1$. O conjunto $\{g_1, g_2, \ldots g_m\}$ é chamado de base normal para G.

Propriedade 4.2 : Todos os automorfismos ϕ de um grupo cíclico G de ordem M têm a forma:

$$\phi(g) = g^k,$$

com mdc $(k, M) = 1$ e $g \in G$. Portanto, existem $\varphi(M)$ automorfismos, onde φ é a função de Euler.

Propriedade 4.3 : A imagem homomórfica de um grupo cíclico G de ordem M é também um grupo cíclico. Mais ainda, todos os endomorfismos ϕ de G têm a forma:

$$\phi(g) = g^h ,$$

com $0 \le h \le M-1$ e $g \in G$. Portanto, o número de endomorfismos é M.

Sejam agora G um grupo abeliano finito como em (4.5), com geradores $g_1, g_2, \ldots, g_m$ (i.e., $\{g_1, g_2, \ldots, g_m\}$ é uma base normal para G) e g um elemento qualquer de G. Iremos associar uma matriz a cada endomorfismo de G. Para tanto, vamos considerar que se ϕ é um endomorfismo de G, então:

$$\phi(g) = \phi(g_1^{\alpha_1} . g_2^{\alpha_2} \ldots g_m^{\alpha_m}) = \phi(g_1)^{\alpha_1} . \phi(g_2)^{\alpha_2} \ldots \phi(g_m)^{\alpha_m}$$

e

$$\phi(g_j) = g_1^{\hat{a}_{1j}} . g_2^{\hat{a}_{2j}} \ldots g_m^{\hat{a}_{mj}} ,$$

$j = 1, \ldots, m$ e onde $g_i^{\hat{a}_{ij}}$ é a imagem de g_j sob um homomorfismo de H_{d_j} em H_{d_i} se $i \neq j$ e é a imagem de um endomorfismo de H_{d_j} se $i = j$ [33]. Portanto,

$$\phi(g) = \prod_{i=1}^{m} g_i^{S_i}$$

onde

$$S_i = \sum_{j=1}^{m} \hat{a}_{ij} . \alpha_j \quad ; \quad i = 1, \ldots m.$$

Agora para que $\phi(g_j)$ seja caracterizado como um homomorfismo, devemos ter:

$$\begin{cases} \hat{a}_{ij} \in Z_{d_i} & , \text{ se } j \ge i \\ \hat{a}_{ij} \in \dfrac{d_i}{d_j} . Z_{d_i} & , \text{ se } j < i \end{cases}$$

Repare, entretanto, que esta condição acima pode ser substituída pela seguinte:

$$\begin{cases} \hat{a}_{ij} \in Z_{d_m} & , \text{ se } j \geq i \\ \hat{a}_{ij} \in \frac{d_i}{d_j}.Z_{d_m} & , \text{ se } j < i \end{cases}$$

E, portanto, o endomorfismo ϕ fica completamente descrito pela matriz $(a_{ij})_{m \times m}$:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ \frac{d_2}{d_1}.a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ \frac{d_3}{d_1}.a_{31} & \frac{d_3}{d_2}a_{32} & a_{33} & \cdots & a_{3m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{d_m}{d_1}.a_{m1} & \frac{d_m}{d_2}a_{m2} & \frac{d_m}{d_3}a_{m3} & \cdots & a_{mm} \end{bmatrix} \qquad (4.6)$$

onde $a_{ij} \in Z_{d_m}$ para $1 \leq i, j \leq m$.

Obs.: Uma conseqüência da Propriedade 4.1 é que todo grupo abeliano finito G pode ser expresso como o produto direto de subgrupos cíclicos cujas ordens são potências de primos, i.e.,

$$G = H_1 \times H_2 \times \ldots \times H_r \qquad (4.7)$$

onde $|H_i| = p_i^{e_i}$, $1 \leq i \leq r$ e os p_i's são primos não necessariamente distintos. Usando esta forma para G, um endomorfismo ϕ (de G) pode ser caracterizado por uma matriz $A = (a_{ij})_{r \times r}$, com elementos em Z_e (onde e é o expoente de G) tal que:

a) $a_{ij} = 0$ se $p_i \neq p_j$;

b) $a_{ij} \in Z_e$ se $p_i \neq p_j$ e $e_j \geq e_i$;

c) $a_{ij} \in p_i^{e_i - e_j} Z_e$ se $p_i = p_j$ e $e_i > e_j$;

Então, escrevendo G na forma (4.7) podemos concluir que não ocorrem mudanças significativas nas matrizes que descrevem os seus endomorfismos. Como em geral, $r \geq m$, a forma (4.5) será preferencialmente usada.

4.5.2. Códigos lineares sobre grupos abelianos

Vamos agora nesta seção descrever os códigos sobre grupos abelianos, baseados principalmente nos resultados da Seção 4.5.1.

Definição 4.8: Um (n, k)-***código linear e sistemático sobre um grupo abeliano*** G é um subgrupo de G^n com ordem $|G|^k$ e descrito por $(n-k)$ homomorfismos de G^k em G. Suas palavras-código têm a forma:

$$(X_1, ..., X_k \mid Y_1, ..., Y_{n-k})$$

onde $X_i \in G$, para $i = 1, ..., k$,

$$Y_\ell = \phi_\ell(X_1, ..., X_k) = \prod_{j=1}^{k} \phi_\ell(e_G, ..., e_G, X_j, e_G, ..., e_G)$$

e todo ϕ_ℓ é caracterizado por k matrizes quadradas do tipo $m \times m$, $A(\ell, h)$, $h = 1, ..., k$, com elementos em Z_{d_m}, conforme descrito em (4.6).

Vamos expressar os elementos X_i e Y_i usando uma base normal para G:

$$X_h = \prod_{i=1}^{m} g_i^{x_{ih}} \quad ; \quad Y_\ell = \prod_{i=1}^{m} g_i^{y_{i\ell}} \quad ; \quad \phi_\ell(X_h) = \prod_{i=1}^{m} g_i^{S_i(\ell,h)}$$

onde

$$S_i(\ell, h) = \sum_{j=1}^{m} a_{ij}(\ell, h).x_{jh}$$

Agora, comparando os expoentes, podemos deduzir que

$$\sum_{h=1}^{k} \sum_{j=1}^{m} [a_{ij}(\ell, h).x_{jh} - y_{i\ell}] = 0 \qquad (4.8a)$$

para $1 \le i \le m$ e $1 \le \ell \le n-k$.

Ou alternativamente,

$$\begin{cases} A_{11}.\underline{x}_1 + A_{12}.\underline{x}_2 + \ldots + A_{1m}.\underline{x}_m = \underline{y}_1 \\ A_{21}.\underline{x}_1 + A_{22}.\underline{x}_2 + \ldots + A_{2m}.\underline{x}_m = \underline{y}_2 \\ \vdots \qquad\quad \vdots \qquad\qquad \vdots \\ A_{m1}.\underline{x}_1 + A_{m2}.\underline{x}_2 + \ldots + A_{mm}.\underline{x}_m = \underline{y}_m \end{cases} \tag{4.8b}$$

onde

$$A_{ij} = \begin{bmatrix} a_{ij}(1,1) & a_{ij}(1,2) & \cdots & a_{ij}(1,k) \\ a_{ij}(2,1) & a_{ij}(2,2) & \cdots & a_{ij}(2,k) \\ \vdots & \vdots & \cdots & \vdots \\ a_{ij}(n-k,1) & a_{ij}(n-k,2) & \cdots & a_{ij}(n-k,k) \end{bmatrix},$$

com $a_{ij}(\,.\,,\,.\,) \in Z_{d_m}$ se $j \geq i$ e $a_{ij} \in (d_i/d_j).Z_{d_m}$ se $j < i$;

$$\underline{x}_i = (x_{i1}\ x_{i2} \ldots x_{ik})^T \quad ; \quad \underline{y}_i = (y_{i1}\ y_{i2} \ldots y_{i,n-k})^T$$

Escrevendo (4.8b) na forma de uma equação matricial, obtemos:

$$\left[\begin{array}{cccc|cccc} A_{11} & A_{12} & \cdots & A_{1m} & -I_{n-k} & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2m} & 0 & -I_{n-k} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} & 0 & 0 & \cdots & -I_{n-k} \end{array}\right] \cdot \begin{bmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_m \\ \underline{y}_1 \\ \vdots \\ \underline{y}_m \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \vdots \\ \underline{0} \end{bmatrix} \tag{4.9}$$

onde as matrizes A_{ij} são do tipo $(n\text{-}k)$ x k, as matrizes identidade são do tipo $(n\text{-}k)$ x $(n\text{-}k)$ e os vetores $\underline{0}$ são do tipo $(n\text{-}k)$ x 1. Esta matriz $m.(n\text{-}k)$ x $m.n$ que aparece em (4.9) (na forma escada) descreve completamente o código e é chamada de ***matriz verificação de paridade***, denotada por H e seus elementos pertencem a Z_{d_m} (lembre, entretanto, que os elementos das submatrizes A_{ij} $(i > j)$ pertencem a $(d_i/d_j).Z_{d_m}$, um subanel de Z_{d_m}). Vamos agora numerar as primeiras $m.k$ colunas de H (correspondendo às posições de

informação) como $h_1 + k.h_2$, onde $h_1 = 1, ..., k$ e $h_2 = 0, ..., m\text{-}1$. As colunas com o mesmo índice h_1 serão chamadas de ***colunas de informação companheiras***. Estas colunas ocupam a mesma posição dentro das submatrizes A_{ij}. Similarmente também podemos definir ***colunas de verificação de paridade companheiras*** numerando analogamente as $m.(n\text{-}k)$ colunas remanescentes. Mais especificamente, seja:

$$
\begin{aligned}
I_1 &= \{h_i : i = 1 + j.k \;,\; 0 \le j \le m\text{-}1\} \\
I_2 &= \{h_i : i = 2 + j.k \;,\; 0 \le j \le m\text{-}1\} \\
&\vdots \\
I_k &= \{h_i : i = k + j.k \;,\; 0 \le j \le m\text{-}1\} \\
P_1 &= \{h_i : i = m.k + 1 + j.(n\text{-}k) \;,\; 0 \le j \le m\text{-}1\} \\
P_2 &= \{h_i : i = m.k + 2 + j.(n\text{-}k) \;,\; 0 \le j \le m\text{-}1\} \\
&\vdots \\
P_{n-k} &= \{h_i : i = m.k + (n\text{-}k) + j(n\text{-}k) \;,\; 0 \le j \le m\text{-}1\}
\end{aligned}
$$

Então cada um dos conjuntos I_i, $1 \le i \le k$, define colunas de informação companheiras e cada um dos conjuntos P_i, $1 \le i \le n\text{-}k$, define colunas de verificação de paridade companheiras.

Iremos agora enunciar dois teoremas os quais caracterizarão a distância mínima de Hamming de códigos sobre grupos abelianos (como em (4.5)) em termos das suas matrizes verificação de paridade. Consideraremos primeiramente o caso mais simples em que G é um grupo cíclico (finito) para posteriormente considerarmos o caso mais geral.

Teorema 4.2 : Seja um (n, k)-código linear e sistemático sobre um grupo cíclico G de ordem t. A matriz verificação de paridade H é uma matriz escada sobre Z_t. A distância mínima de Hamming d deste código é igual ao número mínimo de colunas linearmente dependentes de H.

Prova : Seja g um gerador de G. Neste caso, como G é um grupo cíclico, temos $m = 1$. Então (4.9) torna-se:

$$[A|-I_{n-k}].\begin{bmatrix}\underline{x}\\ \underline{y}\end{bmatrix}=\underline{0}$$

onde A é uma matriz $(n\text{-}k)$ x k com elementos em Z_t. O número mínimo de colunas linearmente dependentes em $H = [A \mid -I_{n-k}]$ dará o número mínimo de expoentes não nulos de g em uma dada palavra-código. Este é o número mínimo de posições nas quais uma palavra-código difere da palavra "toda nula", $(e_G, e_G, \ldots, e_G)$. ■

Teorema 4.3 : Seja agora um (n, k)-código linear e sistemático sobre um grupo abeliano G de expoente d_m. A matriz verificação de paridade H é do tipo $m.(n\text{-}k)$ x $m.n$ sobre Z_{d_m}, conforme mostrado em (4.9). A distância mínima de Hamming do código é dada pelo número mínimo de colunas linearmente dependentes na matriz H sobre o anel Z_{d_m} e com cada conjunto de colunas companheiras contadas como uma só.

Prova : A distância mínima é caracterizada pelo número mínimo de elementos não nulos em uma palavra código. Repare, entretanto, que agora diferentes geradores numa mesma posição não aumentarão a distância. Portanto, colunas companheiras que definem elementos associados com diferentes geradores de G localizados na mesma posição do vetor código contam como uma no peso da palavra código. ■

Vamos ilustrar esta teoria apresentada com um exemplo de um código sobre um grupo abeliano.

Exemplo 4.4 : Seja $n = 5$ e $k = 2$. Vamos exibir a matriz verificação de paridade de um (5, 2, 3) código linear sobre $G = Z_2 \times Z_4$. Neste caso temos dois geradores para G, os quais denotamos g_1 e g_2. Eles satisfazem as relações $g_1^2 = g_2^4 = e_G$ e $g_1 \cdot g_2 = g_2 \cdot g_1$. O código é definido pela seguinte matriz verificação de paridade:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & -1 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Para este código vamos encontrar que $d_{min} = 3$ e as suas palavras têm a forma:

$$\left(g_1^{x_1} g_2^{z_1},\ g_1^{x_2} g_2^{z_2} \,\middle|\, g_1^{x_1+z_1} g_2^{2x_1+2x_2+2z_1+2z_2},\ g_1^{x_2+z_2} g_2^{2x_1+2x_2+z_1},\ g_1^{x_1+3x_2+z_1} g_2^{2x_2+z_1+2z_2} \right)$$

- **Matriz geradora**

As "matrizes geradoras" dos códigos construídos podem ser facilmente deduzidas a partir das suas matrizes verificação de paridade. Os vetores $\underline{y}_i$, $1 \le i \le m$ (que contêm os símbolos de paridade), são obtidos a partir dos vetores $\underline{x}_i$, $1 \le i \le m$ (que contêm os símbolos de informação), através da relação $\underline{v} = \underline{u}.G$, onde:

$$\underline{u} = (\underline{x}_1^T, \underline{x}_2^T, \ldots, \underline{x}_m^T),\ \underline{v} = (\underline{x}_1^T, \underline{x}_2^T, \ldots, \underline{x}_m^T \,|\, \underline{y}_1^T, \underline{y}_2^T, \ldots, \underline{y}_m^T)$$

e

$$G = \left[\begin{array}{cccc|cccc} I_k & 0 & \cdots & 0 & A_{11}^T & A_{21}^T & \cdots & A_{m1}^T \\ 0 & I_k & \cdots & 0 & A_{12}^T & A_{22}^T & \cdots & A_{m2}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & I_k & A_{1m}^T & A_{2m}^T & \cdots & A_{mm}^T \end{array}\right] \qquad (4.10)$$

As submatrizes A_{ij} são aquelas definidas em (4.9).

- **Análise da proposta de construção I**

A construção proposta nos diz que um código sobre um grupo abeliano $G \cong Z_{d_1} \times Z_{d_2} \times \ldots \times Z_{d_m}$, onde $d_1 \mid d_2 \mid \ldots \mid d_m$ pode ser construído imitando-se o procedimento usado para corpos e anéis, i.e., através do uso de uma matriz verificação de paridade sobre o anel Z_{d_m}, dada por:

$$H = \left[\begin{array}{cccc|cccc} A_{11} & A_{12} & \cdots & A_{1m} & -I_{n-k} & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2m} & 0 & -I_{n-k} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} & 0 & 0 & \cdots & -I_{n-k} \end{array}\right]$$

A complexidade desta construção reside no fato de que a determinação de um código com distância mínima de Hamming d exige que certos conjuntos de $(d - 1).m$ colunas de H sejam linearmente independentes, de acordo com o Teorema 4.3. Além disso, os elementos das submatrizes A_{ij} $(i > j)$ de H deverão obrigatoriamente estar contidos no subanel $(d_i/d_j).Z_{d_m}$ de Z_{d_m}. Estes fatos nos mostram que esta construção é sensivelmente mais elaborada que a construção de códigos lineares de mesmo desempenho sobre corpos e anéis.

- **Decodificação**

Lembre que cada palavra-código é formada por elementos do grupo, os quais são expressos como produtos de potências dos geradores do mesmo. A matriz verificação de paridade proveniente da Proposta I relaciona essas diversas potências. O vetor erro, por sua vez, também pode ser descrito pelas potências dos geradores em cada posição.

Portanto, um método de decodificação convencional que se baseia na matriz verificação de paridade não pode ser aplicado diretamente, devido ao fato de que esses métodos sempre procuram determinar o vetor erro que possui o menor peso de Hamming possível. Neste caso pode ocorrer que o vetor de potências que descreve um erro em uma única posição tenha um alto peso de Hamming e possivelmente não será decodificado corretamente. Este é o principal obstáculo que a matriz verificação de paridade apresentada pela Proposta I oferece quando usada para decodificação.

4.6. Construção e Decodificação de Códigos Lineares sobre Grupos Abelianos (Proposta II)

Nesta nova proposta de construção, usaremos o fato de que todo grupo abeliano finito G também pode ser escrito como o produto direto de subgrupos cíclicos cujas ordens são potências de primos, i.e.,

$$G = H_1 \times H_2 \times \ldots \times H_m$$

onde $|H_i| = p_i^{e_i}$ $(1 \leq i \leq m)$ e p_i é um primo. Este fato é uma conseqüência direta da Propriedade 4.1.

Com base nisto, iremos mostrar que o desempenho de códigos lineares sobre grupos abelianos (em termos da distância mínima e taxa) é limitado pelo desempenho de subcódigos definidos sobre os subgrupos H_i de G. A partir daí faremos uma proposta de construção e posteriormente de decodificação de tais códigos.

Seja então G um grupo abeliano finito tal que

$$G \cong Z_{p_1^{e_1}} \times Z_{p_2^{e_2}} \times \ldots \times Z_{p_m^{e_m}} \tag{4.11}$$

e C um (n, k)-código linear e sistemático sobre G. Vamos analisar os seguintes casos, baseando-nos em (4.11):

a) suponha que os primos p_i sejam todos distintos, para $1 \leq i \leq m$. A partir disto, considere os (n, k)-subcódigos

$$C_i \triangleq \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} p_j^{e_j}\right).C,$$

para $1 \leq i \leq m$. Note que o subcódigo C_i está definido sobre $Z_{p_i^{e_i}}$ e, portanto, a distância mínima de C ficará limitada pela menor das distâncias mínimas dos subcódigos C_i, $1 \leq i \leq m$.

b) suponha que os primos p_i sejam todos iguais a p e os expoentes e_i sejam todos iguais a e. Neste caso, o (n, k)-subcódigo $p^{e-1}.C$, que está definido sobre $GF(p^m)$, é que limitará a distância mínima de C;

c) suponha que os primos p_i sejam todos iguais a p e os expoentes e_i não sejam todos iguais. Sem perda de generalidade, podemos supor que $e_m = e$ é o maior deles. Neste caso, o (n, k)-subcódigo $p^{e-1}.C$, que está definido sobre $GF(p)$, é que limitará a distância mínima de C;

d) finalmente considere o caso mais geral em que os primos p_i não sejam todos distintos, i.e.,

$$G \cong Z_{p_1^{e_{1,1}}} \times \ldots \times Z_{p_1^{e_{1,r_1}}} \times \ldots \times Z_{p_n^{e_{n,1}}} \times \ldots \times Z_{p_n^{e_{n,r_n}}}.$$

Sem perda de generalidade podemos supor que $e_{i,r_i} \geq e_{i,r_i-1} \geq \ldots \geq e_{i,1}$ para $1 \leq i \leq n$. Neste caso, os (n, k)-subcódigos

$$C_i \triangleq \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} p_j^{e_{j,r_j}}\right).C,$$

definidos sobre $Z_{p_i^{e_{i,1}}} \times \ldots \times Z_{p_i^{e_{i,r_i}}}$ $1 \leq i \leq m$, é que limitarão a distância mínima de C. Este caso já foi abordado em c).

Podemos então concluir por a), b), c) e d) que qualquer construção de códigos lineares e sistemáticos sobre grupos abelianos apresentará necessariamente um desempenho igual ao de um subcódigo sobre um anel de inteiros residuais da forma Z_q ou sobre um corpo GF(q), onde q é uma potência de primo.

Neste ponto chamamos a atenção para o fato de que a Proposta II a seguir é completamente distinta da Proposta I e suas notações não devem ser confundidas.

Proposta II : Construção de um (n, k, d)-código linear sobre um grupo abeliano G, tal que

$$G \cong Z_{p_1^{e_1}} \times Z_{p_2^{e_2}} \times \ldots \times Z_{p_m^{e_m}}.$$

Para isto, será necessário selecionarmos m (n, k)-códigos lineares sobre $Z_{p_1^{e_1}}, Z_{p_2^{e_2}}, \ldots, Z_{p_m^{e_m}}$, respectivamente, e tal que a distância mínima de Hamming de cada um deles seja pelo menos d. Se $d \leq 3$, os códigos de Hamming são suficientes; caso contrário, se $d > 3$, recorreremos a códigos de múltipla correção, e.g., Reed-Solomon ou BCH. O código sobre $Z_{p_i^{e_i}}$ será um código sobre um corpo Z_{p_i}(GF(p_i)) se $e_i = 1$. Caso contrário, se $e_i \geq 2$, o código deverá ser um daqueles estudados no Capítulo 1.

Para codificarmos a seqüência de informação $(g_1, g_2, \ldots, g_k)$, com $g_i \in G$, primeiramente expressamos cada elemento g_i na forma de um produto direto, i.e., $g_i = (a_{1i}, a_{2i}, \ldots, a_{mi})$, para $1 \le i \le k$ e com $a_{\ell i} \in Z_{p_\ell^{e_\ell}}$ $(1 \le \ell \le m)$. Com isto, toda a seqüência de informação é processada conforme o esquema abaixo:

$$
\begin{array}{ccccccc}
(a_{11} & a_{12} & \ldots & a_{1,k}) & \rightarrow & \boxed{\begin{array}{c}C\\O\\D\\I\\F.\end{array}} & \rightarrow \quad (a_{11} \quad \ldots \quad a_{1,k} \quad a_{1,k+1} \quad \ldots \quad a_{1,n}) \\
(a_{21} & a_{22} & \ldots & a_{2,k}) & \rightarrow & & \rightarrow \quad (a_{21} \quad \ldots \quad a_{2,k} \quad a_{2,k+1} \quad \ldots \quad a_{2,n}) \\
 & \ldots & & & & & \\
(a_{m,1} & a_{m,2} & \ldots & a_{m,k}) & \rightarrow & & \rightarrow \quad (a_{m,1} \quad \ldots \quad a_{m,k} \quad a_{m,k+1} \quad \ldots \quad a_{m,n}) \\
\uparrow & \uparrow & & \uparrow & & & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
(g_1 & g_2 & \ldots & g_k) & & & (g_1 \quad \ldots \quad g_k \quad g_{k+1} \quad \ldots \quad g_n)
\end{array}
\tag{4.12}
$$

Vamos analisá-lo. No lado esquerdo os elementos de cada linha são da forma a_{ij} (para i fixo e $1 \le j \le k$) e todos pertencem a um mesmo anel $Z_{p_i^{e_i}}$. Portanto, a seqüência de elementos $(a_{i1} \; a_{i2} \ldots a_{ik})$ pode ser considerada como os símbolos de informação de um (n, k, d_i)-código sobre $Z_{p_i^{e_i}}$. O lado direito é formado pelas palavras-código correspondentes a cada uma dessas seqüências de informação. Do mesmo modo, fazemos o produto direto de cada coluna no lado direito para obtermos os correspondentes elementos do grupo que formarão a palavra-código.

A taxa do código C sobre G, proveniente da construção em (4.12), é dada por $\log_{|G|} |G|^k / n = k/n$ e é a mesma dos m códigos que a compõem. A distância mínima de Hamming de C é

$$
d_{min}(C) = \min_{1 \le i \le m} \{d_i\} \tag{4.13}
$$

onde d_i é a distância mínima de Hamming do i-ésimo código definido sobre $Z_{p_i^{e_i}}$.

- **Decodificação**

A decodificação deverá proceder da seguinte maneira: tendo recebido uma palavra de comprimento n (com elementos em G), o receptor decompõe a mesma num produto direto (como no lado direito de (4.12)) para obter m palavras, também de comprimento n, cada uma sobre $Z_{p_i^{e_i}}$, $1 \le i \le m$. A partir daí, m decodificadores (um para cada código em (4.12)) decodificam as palavras correspondentes. Estes decodificadores são os estudados no Capítulo 2.

4.7. Esquemas de Modulação Codificada, Códigos de Classes Laterais e Empacotamentos Esféricos

Passaremos agora a associar uma modulação (conjunto de pontos do $\mathbb{R}^N$) ao alfabeto (grupo) sobre o qual um código linear C está sendo construído. Esquemas de modulação codificada foram extensivamente estudados, e.g., [36]-[38]. Iremos aqui brevemente indicar como a construção (4.12) pode ser aplicada em um tal esquema.

Sejam S um conjunto finito de pontos do $\mathbb{R}^N$, G um grupo abeliano finito tal que

$$G \cong Z_{p_1^{e_1}} \times Z_{p_2^{e_2}} \times \ldots \times Z_{p_m^{e_m}}.$$

onde os p_i's são primos não necessariamente distintos e C um código linear sobre G, construído como em (4.12). Assuma que $|S| = |G|$. Iremos agora associar rótulos (elementos de G) a cada ponto de S da seguinte forma. Suponha que S possa ser particionado, num primeiro nível, em $p_1^{e_1}$ subconjuntos, cada um recebendo um rótulo diferente, entre 0 e $p_1^{e_1} - 1$. A seguir, suponha que cada um desses subconjuntos possa ser particionado em $p_2^{e_2}$ subconjuntos (novamente cada um recebendo um rótulo diferente, agora entre 0 e $p_2^{e_2} - 1$.) e assim sucessivamente até o m-ésimo nível, contendo $\prod_{i=1}^{m} p_i^{e_i}$ subconjuntos, cada qual com

um único ponto. Cada um desses pontos no último nível receberá o rótulo $(a_1\ a_2 \ldots\ a_m) = g$ (para algum $g \in G$), onde a_i é o rótulo do subconjunto do i-ésimo nível que o contém.

Além disso, estes particionamentos são realizados de forma tal que a distância intra-conjunto (a menor distância Euclidiana quadrática entre dois pontos pertencentes a um mesmo subconjunto) Δ_i, $1 \le i \le m$, aumente a cada nível.

Mostra-se então que a distância mínima Euclidiana entre duas palavras de C satisfaz

$$\Delta_{min}(C) = \min_{1 \le i \le m} \{d_{min}(C_i).\Delta_i\}$$

onde $d_{min}(C_i)$ é a distância mínima de Hamming do i-ésimo código definido sobre $Z_{p_i^{e_i}}$ em (4.12). Neste caso, esquemas de decodificação de menor complexidade (decodificação multiestágio) [36] são utilizados, embora não sejam de máxima verossimilhança.

Terminaremos esta seção lembrando que esquemas de modulação codificada usando a modulação PSK e códigos sobre grupos abelianos foram também consideradas em [43].

* **Códigos de Classes Laterais ("Coset Codes") ([39], [40])**

Neste caso, tem-se inicialmente um reticulado Λ (subgrupo discreto e aditivo do $\mathbb{R}^N$) o qual se deseja usar para sinalização. O particionamento será realizado como descrito anteriormente, porém agora os subconjuntos do último nível contêm infinitos pontos. Portanto, a cada elemento do grupo corresponderá um subconjunto do qual se escolherá, através de um seletor de sinais, um ponto para transmissão.

Então, a partir de um código C de comprimento n sobre G constrói-se um código de classes laterais $\mathbb{C}$ que consistirá, portanto, de um conjunto infinito de nN-uplas. Tal código é denotado por

$$\mathbb{C}(\Lambda^{(0)}/\Lambda^{(1)}/\ldots/\Lambda^{(m)}\ ;\ C_1, C_2, \ldots, C_m)\ ,$$

onde $\Lambda = \Lambda^{(0)}/\Lambda^{(1)}/\ldots/\Lambda^{(m)}$ representa a cadeia de partições do reticulado Λ e C_i, $1 \le i \le m$, representa um código definido sobre $Z_{p_i^{e_i}}$, como em (4.12). Note ainda que devemos ter

$$|\Lambda^{(i-1)}/\Lambda^{(i)}| = p_i^{e_i} \quad , \quad 1 \le i \le m$$

Finalmente, note que $\mathbb{C}$ fica então definido por:

$$\mathbb{C} = C_1 + C_2 + \ldots + C_m + (\Lambda^{(m)})^n$$

Pode-se mostrar que a distância Euclidiana quadrática entre dois pontos quaisquer $\underline{c}_1$ e $\underline{c}_2$ do código de classes laterais é limitada por:

$$d^2(\underline{c}_1, \underline{c}_2) \ge \min\{\delta_1 . d_{min}(C_1), \delta_2 . d_{min}(C_2), \ldots, \delta_m . d_{min}(C_m), \Delta_m\},$$

onde $d_{min}(C_i)$ denota a distância mínima de Hamming do i-ésimo código em (4.12), Δ_i denota a distância intra-conjunto (ou intra-reticulado) em $\Lambda^{(i)}$ e δ_i denota a distância inter-conjunto (ou inter-reticulado) entre as classes laterais de $\Lambda^{(i)}$ em $\Lambda^{(i-1)}$ [39].

* **Empacotamentos Esféricos**

Se os pontos de $\mathbb{C}$ estiverem separados por uma distância mínima de 2ρ, então esferas de raio ρ centradas nestes pontos podem ser usadas para a construção de empacotamentos esféricos [39]. O objetivo destas construções é maximizar o ***ganho de codificação*** $\gamma(\mathbb{C})$ do empacotamento esférico (de dimensão N) definido por

$$\gamma(\mathbb{C}) = d^2(\mathbb{C}) / V(\mathbb{C})^{2/N} \ ,$$

onde $d^2(\mathbb{C})$ é a distância mínima Euclidiana de $\mathbb{C}$ e $V(\mathbb{C})$ é o volume médio das ***regiões de Voronoi*** de $\mathbb{C}$. Dado um ponto em $\mathbb{C}$, a região de Voronoi do mesmo consiste de todos os pontos de $\mathbb{R}^N$ que estão mais próximos a ele do que a qualquer outro ponto em $\mathbb{C}$. Interpretações físicas a respeito de $\gamma(\mathbb{C})$ podem ser encontradas em [41].

Iremos terminar esta seção deixando para estudos futuros a investigação de reticulados em $\mathbb{R}^N$ e suas partições que possam produzir bons códigos de classes laterais e conseqüentemente bons empacotamentos esféricos. Presentemente, as partições mais conhecidas são as q-árias, i.e., aquelas em que os subconjuntos de cada nível são sempre particionados em q novos subconjuntos. Uma exceção a esta prática pode ser encontrada em [42], onde são realizadas partições ternárias e quaternárias do reticulado A_2 a fim de se obter empacotamentos esféricos densos. Métodos sistemáticos para realização de particionamentos q-ários (onde $q \ge 2$ é um inteiro) podem ser encontrados em [44] e nas referências contidas na mesma.

4.8. Conclusões

Neste capítulo foi estudada a teoria básica de códigos lineares definidos sobre grupos finitos. Foram fornecidas as definições de parâmetros tais como comprimento, dimensão, taxa, distância (todas análogas às definições de códigos sobre corpos) e ainda a introdução do conceito de conjuntos de informação que caracterizam os códigos sistemáticos.

A seguir, foram caracterizadas as condições sob as quais um código sistemático é linear, i.e., as funções que mapeiam os símbolos de informação nos símbolos de paridade devem ser homomorfismos. Foi mostrado ainda que códigos sobre grupos não-abelianos são assintoticamente ruins e com isto nenhum método de construção foi proposto. Neste caso, uma alternativa é abelianizar o grupo (não-abeliano) e construir códigos sobre os grupos abelianos obtidos [6].

Daí então analisamos a proposta de construção de códigos sobre grupos abelianos, estudada em [29]. Do ponto de vista teórico, ela é bastante interessante, no sentido em que faz uma analogia com construções de códigos sobre corpos, i.e., através do uso de uma matriz verificação de paridade. Além disso, a partir desta construção, pudemos concluir que o desempenho de códigos sobre um grupo abeliano de expoente e é sempre limitado pelo desempenho de códigos sobre anéis Z_e.

A dificuldade de se sistematizar o método para construções de códigos para múltipla correção de erros nos levou a fazer uma proposta alternativa, a qual consiste em concatenar códigos definidos sobre o grupo aditivo de anéis do tipo Z_q (onde q é uma potência de primo) para obtermos um código linear sobre um grupo abeliano. Mostramos ainda que a distância mínima de Hamming de qualquer código sistemático e linear sobre um dado grupo abeliano é limitada superiormente pela distância mínima de subcódigos (de mesma taxa) definidos sobre o grupo aditivo de anéis Z_q, onde q é uma potência de primo.

Finalmente indicamos como a construção proposta (4.12) poderá ser usada para se obter esquemas de modulação codificada, códigos de classes laterais e empacotamentos esféricos.

APÊNDICE 4.1

Uma Nota sobre Códigos Perfeitos

Sabemos que a grande maioria dos códigos com capacidade de correção de t erros (onde $t = \lfloor (d_{min} - 1)/2 \rfloor$) pode ainda corrigir alguns padrões de erro contendo mais do que t erros. Os códigos ***perfeitos*** são aqueles que corrigem apenas os padrões com t ou menos erros e não mais. Uma interpretação geométrica disto é que se centrarmos esferas de raio t nas palavras-código (pertencentes a um código perfeito), elas preencherão todo o espaço.

Já foi mostrado que os únicos códigos perfeitos sobre corpos GF(q) (onde q é uma potência de primo) são:

i) códigos binários de repetição;

ii) códigos perfeitos q-ários com $d_{min} = 3$ e $n = (q^r - 1) / (q - 1)$, para algum inteiro r;

iii) o (23, 12, 7)-código de Golay sobre GF(2);

iv) o (11, 6, 5)-código de Golay sobre GF(3).

Para maiores detalhes, veja [45] e as referências contidas na mesma.

Iremos mostrar aqui que não existem códigos perfeitos de "Hamming" ($t = 1$) definidos sobre anéis Z_m, para m diferente de primo. Depois disto, baseados na teoria da Seção 4.5, concluiremos que não existem códigos perfeitos com $t = 1$, definidos sobre grupos cíclicos cuja ordem é diferente de primo.

Antes porém, vamos recordar algumas propriedades. Seja C um código perfeito de comprimento n, capacidade de correção t e número de palavras igual a M. Se o alfabeto A sobre o qual C está definido tem cardinalidade $|A| = m$, então é válida a seguinte relação:

$$M.\sum_{i=0}^{t}\binom{n}{i}(m-1)^i = m^n \qquad (4.14)$$

a qual é conhecida como a condição do empacotamento esférico [45].

Considere agora o caso em que C é um (n, k)-código linear sobre A e com $t = 1$. Então, (4.14) torna-se:

$$m^k . \sum_{i=0}^{1} \binom{n}{i} . (m-1)^i = m^n \tag{4.15}$$

E isto implica que

$$n = \frac{m^{n-k} - 1}{m - 1} \tag{4.16}$$

Ou seja, se n-$k = \ell$, então um (n, k)-código linear e perfeito com $t = 1$ e definido sobre um alfabeto contendo m elementos deverá ter como parâmetros:

$$n = \frac{m^{\ell} - 1}{m - 1}, \quad k = \frac{m^{\ell} - 1}{m - 1} - \ell \quad \text{e} \quad d_{min} = 3.$$

Proposição 4.9 : Sejam os vetores $\underline{h}_1$ e $\underline{h}_2 \in Z_m^{\ell}$ tais que $a.\underline{h}_1 = (1\ \ 1\ \ 1\ \ ...\ 1)^T$ e $b.\underline{h}_2 = (1\ \ 1\ \ ...\ 1\ b)^T$, com $b > 1$, para algum par $a, b \in Z_m$. Então, $\underline{h}_1 \neq \underline{h}_2$.

Prova : Seja $\underline{h}_1 = (h_{11}\ \ h_{12}\ ...\ h_{1,\ell})^T$. Como $a.\underline{h}_1 = (a.h_{11}\ a.h_{12}\ ...\ a.h_{1,\ell})^T = (1\ \ 1\ ...\ 1)^T$, então $h_{11} = h_{12} = ... = h_{1,\ell}$, já que o inverso de a (quando existe) é único [4]. Isto implica que todo múltiplo de h_1 é da forma ($r\ \ r\ \ r\ ...\ r$), com $r \in Z_m$. Portanto, $\underline{h}_1 \neq \underline{h}_2$, pois existe um múltiplo de $\underline{h}_2$ da forma (1 1 1 ... 1 b), com $b > 1$. ■

Teorema 4.4 : Não existem códigos lineares e perfeitos com $t = 1$ sobre anéis Z_m, onde m é diferente de primo.

Prova : Recorde que os parâmetros desses códigos devem ser dados por:

$$n = \frac{m^{\ell} - 1}{m - 1}, \quad k = \frac{m^{\ell} - 1}{m - 1} - \ell, \quad d_{min} = 3 \quad \text{e} \quad \ell > 1.$$

Para demonstrarmos este teorema, iremos nos basear na matriz verificação de paridade H que possui n colunas de ℓ componentes cada. Daí concluiremos que existem duas colunas linearmente dependentes em H, o que implicaria que $d_{min} \leq 2$.

Suponha então, por absurdo, que seja dado um código com os parâmetros acima e que H seja a sua matriz verificação de paridade com as colunas $\underline{h}_1, \underline{h}_2, \ldots, \underline{h}_n$ pertencentes a Z_m^{ℓ}. A partir disto, podemos concluir que:

i) para toda coluna $\underline{h}_i$, $a.\underline{h}_i \neq \underline{0}$ $(1 \leq i \leq n)$ para todo $a \neq 0$. Se isto não fosse verdade, a distância mínima do código seria 1;

ii) para quaisquer que sejam as colunas $\underline{h}_i$ e $\underline{h}_j$, vale que $a.\underline{h}_i + b.\underline{h}_j \neq \underline{0}$, com a e b não ambos nulos. Se isto não fosse verdade, a distância mínima do código seria, no máximo, 2.

Portanto, $a.\underline{h}_i \neq b.\underline{h}_j$, $\forall$ a e b não ambos nulos. Defina então os conjuntos:

$$A_i = \{a.\underline{h}_i \ : \ a \neq 0\}, \qquad 1 \leq i \leq n.$$

Temos então que:

a) a cardinalidade $|A_i|$ de A_i é m - 1. Do contrário, existiriam a e $b \in Z_m$ $(a \neq b)$ tais que $a.\underline{h}_i = b.\underline{h}_i$. Daí $(a - b)\underline{h}_i = 0$ com $a \neq b$ e isto contradiz i);

b) por ii), $A_i \cap A_j = \phi$, para $i \neq j$;

Por a) e b) podemos concluir que o conjunto $A = A_1 \cup A_2 \cup \ldots \cup A_n$ é conjunto de todas as ℓ-uplas pertencentes a Z_m (com exceção da ℓ-upla toda nula). Note que $|A| = n.(m - 1) = m^{\ell} - 1$.

Sejam agora os vetores:

$$\underline{v}_1 = (1\ 1\ 1\ \ldots\ 1)^T \quad \text{e} \quad \underline{v}_2 = (1\ 1\ 1\ \ldots\ 1\ h)$$

com $h > 1$ e $(h - 1)$ divisor de zero. (Lembre que é sempre possível escolher um divisor de zero em Z_m, para m diferente de primo). Como o conjunto A é o conjunto de todas as

ℓ-uplas (com exceção da ℓ-upla toda nula), então $\underline{v}_1 \in A_i$ e $\underline{v}_2 \in A_j$, para algum par i, j. Daí $\underline{v}_1 = a.\underline{h}_i$ e $\underline{v}_2 = b.\underline{h}_j$. Pela Proposição 4.9, devemos ter $i \neq j$.

Sendo $(h - 1)$ divisor de zero, $\exists\ c \neq 0$ tal que $c.(h - 1) = 0$ ou equivalentemente, $\exists\ c \neq 0$ tal que $c.h = c$. Daí:

$$c.\underline{v}_1 = (c\ \ c\ \ c\ \ldots\ c) \text{ e}$$

$$c.\underline{v}_2 = (c\ \ c\ \ c\ \ldots\ c.h) = (c\ \ c\ \ c\ \ldots\ c)$$

Isto implica que $-c.\underline{v}_1 + c.\underline{v}_2 = 0$, ou seja, $(-c.a).\underline{h}_i + (c.b)\ \underline{h}_j = 0$ com $c.a$ e $c.b$ diferentes de zero (se $c.a$ e $c.b$ fossem zero, então $c.\underline{v}_1$ e $c.\underline{v}_2$ seriam zero).

Portanto, a distância mínima do código é 2, o que é uma contradição. E isto prova que não existe o código com os parâmetros dados na hipótese. ■

Da Seção 4.5, sabemos que um (n, k)-código linear e sistemático sobre um grupo abeliano G tal que

$$G \cong Z_{d_1} \times Z_{d_2} \times \ldots \times Z_{d_m}$$

tem a matriz H em (4.9) como a sua matriz verificação de paridade. Lembre que H deverá estar sobre o anel Z_{d_m} (onde d_m é o expoente de G) e que ela é do tipo $m.(n-k) \times m.n$. Então, baseados no Teorema 4.4, temos o Corolário 4.2.

Corolário 4.2 : Não existem códigos lineares e sistemáticos que são perfeitos com $t = 1$ sobre grupos cíclicos cuja ordem é diferente de primo.

Prova : Imediata. ■

CAPÍTULO 5

CONCLUSÕES

Neste trabalho foi dado um primeiro passo rumo à caracterização de códigos lineares sobre grupos finitos. A principal motivação deste estudo foi a construção de códigos de classes laterais generalizados a partir de códigos (sinais) geometricamente uniformes "elementares", conforme sugerido na Seção IV de [2].

O estudo de códigos sobre grupos finitos nos permitiu concluir vários aspectos com relação ao seu desempenho (taxa e distância mínima) e dentre eles os mais relevantes foram:

i) códigos lineares e sistemáticos sobre um grupo abeliano G qualquer (finito) possuem desempenho necessariamente limitado por um subcódigo definido sobre um subgrupo de G, o qual é isomorfo ao grupo aditivo de um corpo finito GF(q) ou de um anel de inteiros Z_q, onde q é uma potência de primo;

ii) códigos lineares e sistemáticos sobre grupos não-abelianos (também finitos) são assintoticamente ruins, i.e., a razão entre a distância mínima e o comprimento dos códigos vai a zero à medida que o comprimento dos códigos aumenta (isto foi mostrado por Biglieri e Elia em [29]). Uma sugestão dada por Forney [6] foi a de se *abelianizar* os grupos não-abelianos de interesse e então construir códigos sobre os grupos (abelianos) provenientes desta *abelianização*.

Optamos por investigar somente códigos sobre grupos abelianos, considerando a geração e a decodificação dos mesmos.

A identificação de códigos sobre grupos abelianos com códigos sobre o grupo aditivo de anéis Z_q nos permitiu tomar vantagem da estrutura de anel de Z_q, i.e., ainda foi possível - nos casos de interesse - caracterizar os códigos em termos de matrizes geradoras e verificação de paridade e conseqüentemente tratá-los de forma bastante análoga a códigos sobre corpos GF(q).

Nesse sentido, iniciamos o trabalho com a investigação das principais classes de códigos sobre anéis Z_q que possuem os correspondentes em corpos, i.e., códigos cíclicos, de Hamming, Reed-Solomon e BCH. A seguir foram derivados os respectivos algoritmos de decodificação para estes códigos, procurando sempre manter analogia com os algoritmos eficientes de decodificação dos correspondentes códigos definidos sobre corpos.

Uma contribuição importante foi a generalização do algoritmo de Berlekamp-Massey (BM) para a decodificação de códigos Reed-Solomon e BCH quando definidos sobre anéis Z_q. Uma aplicação imediata desta generalização foi a síntese de registros de deslocamento que geram seqüências prescritas cujos elementos pertencem a um anel comutativo com identidade.

Ainda dentro deste escopo foi considerado o problema da síntese de registros de deslocamento para a geração de multiseqüências (com elementos também pertencentes a um anel comutativo com identidade) o qual foi resolvido a partir de uma outra generalização do algoritmo de BM. Esta generalização foi obtida a partir de um refinamento de um algoritmo mais geral, conhecido como Algoritmo Iterativo Fundamental (AIF), o qual resolve o problema da determinação do menor conjunto inicial de colunas linearmente dependentes em uma matriz cujas entradas pertencem também a um anel comutativo com identidade. Mostramos ainda mais que a generalização do algoritmo de BM para multiseqüências podia também ser utilizada para a decodificação de certas classes de códigos cíclicos sobre Z_q até o limitante de Hartmann-Tzeng (HT), onde múltiplas seqüências de síndromes são disponíveis.

Terminado então o estudo de códigos lineares sobre anéis Z_q, passamos a descrever os códigos lineares sobre grupos. Inicialmente discorremos sobre as principais contribuições dadas para este tópico, e.g., as contidas no trabalho por Biglieri e Elia [29] no qual os autores propuseram construir códigos lineares sobre grupos abelianos através do uso de uma matriz verificação de paridade a qual possui entradas em um anel Z_q.

Vimos que o principal obstáculo apresentado por esta construção é que a referida matriz verificação de paridade impõe certas restrições aos seus elementos, as quais dificultam a sistematização do processo de construção de novas classes de códigos.

Finalmente apresentamos uma proposta alternativa em que os códigos sobre grupos abelianos são obtidos via concatenação generalizada de códigos sobre anéis de inteiros residuais. O processo de decodificação consistiu basicamente de se construir decodificadores para cada código componente na concatenação. Foram indicadas também aplicações desta construção em esquemas de modulação codificada, códigos de classes laterais e empacotamentos esféricos.

- **Tópicos para pesquisa futura**

Aparentemente existem vários tópicos abordados neste trabalho os quais poderiam ser pesquisados em maior profundidade. A lista abaixo relaciona alguns deles:

1) generalização de outras classes de códigos com símbolos em corpos finitos, e.g., códigos alternantes, para códigos com símbolos em anéis de inteiros residuais Z_q;

2) investigação de classes de códigos definidos sobre anéis com estrutura diferente de Z_q, e.g., anéis inteiros algébricos;

3) adaptação dos algoritmos de decodificação - baseados na distância de Hamming - para outras distâncias, e.g., a distância de Lee;

4) o método de obtenção de códigos sobre grupos abelianos depende basicamente da disponibilidade de códigos sobre anéis de inteiros residuais e isto acabou nos levando naturalmente a fazer uso das duas operações de anel, a "adição" e a "multiplicação", quando da geração e decodificação dos códigos sobre grupos. Uma sugestão seria a de se usar *apenas* uma operação, a operação "aditiva" do grupo abeliano em questão.

BIBLIOGRAFIA

[1] C.E. Shannon, "A Mathematical Theory of Communication", *Bell Syst., Tech. J.*, Vol. 27 (1948), pp. 397-423 and 623-656.

[2] G.D. Forney, Jr., "Geometrically Uniform Codes", *IEEE Trans. Inform. Theory*, Vol. 37, pp. 1241-1260, September 1991.

[3] A.R. Hammons, Jr., P.V. Kumar, A.R., Calderbank, N.J.A. Sloane and P. Solé, "The Z_4-Linearity of Kerdock, Preparata, Goethals and Related Codes", *IEEE Trans. Inform. Theory*, Vol. 40, No. 2, pp. 301-319, March 1994.

[4] J.B. Fraleigh, *A First Course in Abstract Algebra*, Addison-Wesley Publishing Co., 1982.

[5] C.F. Gardiner, *Algebraic Structures*, Chinchester: Ellis Horwood Limited, 1986.

[6] G.D. Forney, Jr., "On the Hamming Distance Properties of Group Codes", *IEEE Trans. Inform. Theory*, Vol. 38, No. 6, pp. 1797-1801, November 1992.

[7] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error Correcting Codes*, North-Holland Publishing Company, 1977.

[8] V. Pless, *Introduction to the Theory of Error Correcting Codes*, New York: John Wiley & Sons, 1989.

[9] S. Lin and D.J. Costello, Jr., *Error Control Coding*, Prentice-Hall, Inc., 1983.

[10] I.F. Blake, "Codes over Certain Rings", *Inform. Contr.* 20, pp. 396-404, 1972.

[11] I.F. Blake, "Codes over Integer Residue Rings", *Inform. Contr.* 29, pp. 295-300, 1975.

[12] P. Shankar, "On BCH Codes over Arbitrary Integer Rings", *IEEE Trans. Inform. Theory*, Vol. IT-25, No. 4, pp. 480-483, July 1979.

[13] B.R. McDonald, *Finite Rings with Identity*, New York: Marcel Dekker, 1974.

[14] J.C. Interlando and R. Palazzo, Jr., "Modified Berkekamp-Massey Algorithm for Decoding BCH Codes Defined over Rings", *Proceedings of the IEEE International Symposium on Inform. Theory* (ISIT), 27 June - 1 July 1994 - Trondheim, Norway, pp. 94.

[15] W.W. Peterson and E.J. Weldon, Jr., *Error Correcting Codes*, 2nd. ed., MIT Press, Cambridge, Mass., 1972.

[16] B.R. McDonald, *Linear Algebra over Commutative Rings*, New York: Marcel Dekker, Inc., 1984.

[17] W.C. Brown, *Matrices over Commutative Rings*, New York: Marcel Dekker, 1993.

[18] N.H. McCoy, *Rings and Ideals*, Carus Math. Monograph No. 8, The Mathematical Association of America, Washington, D.C., 1948.

[19] G.D. Forney, Jr., "On Decoding BCH Codes", *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 549-557, October 1965.

[20] L. Childs, *A Concrete Introduction to Higher Algebra*, Springer-Verlag, 1979.

[21] H.E. Rose, *A Course in Number Theory*, Clarendon Press, Oxford, 1988.

[22] E. Grosswald, *Topics from the Theory of Numbers*, 2nd. ed., Birkhäuser, Boston, 1984.

[23] E. Weiss, *first course in Algebra and Number Theory*, Academic Press, 1971.

[24] E.R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[25] J.L. Massey, "Shift Register Synthesis and BCH Decoding", *IEEE Trans. Inform. Theory*, Vol. IT-15, pp. 122-127, January 1969.

[26] J.A. Reeds and N.J.A. Sloane, "Shift-Register Synthesis (modulo *m*)", *Siam J. Comput.*, Vol. 14, No. 3, pp. 505-513, August 1985.

[27] G.-L. Feng and K.K. Tzeng, "A Generalization of the Berlekamp-Massey Algorithm for Multisequence Shift-Register Synthesis with Applications to Decoding Cyclic Codes", *IEEE Trans. Inform. Theory*, Vol. 37, No. 5, pp. 1274-1287, September 1991.

[28] C.R.P. Hartmann and K.K. Tzeng, "Generalizations of the BCH Bound", *Inform. Contr.*, 20, No. 5, pp. 489-498, June 1972.

[29] E. Biglieri and M. Elia, "Construction of Linear Block Codes over Groups". Submitted to *IEEE Trans. Inform. Theory*.

[30] D. Slepian, "Group Codes for the Gaussian Channel", *Bell Syst. Tech. J.*, Vol. 47, pp. 575-602, 1968.

[31] R. Palazzo, Jr., J.C. Interlando and C. de Almeida, "Construction of Signal Sets Matched to to Abelian and Non-Abelian Groups", *Proceedings of the IEEE International Symposium on Inform. Theory* (ISIT), 27 June - 1 July 1994 - Trondheim, Norway, pp. 488.

[32] C.W. Curtis and J. Reiner, *Representation Theory of Finite Groups and Associative Algebras*, John Wiley & Sons, Inc., 1962.

[33] A.G. Kurosh, *The Theory of Groups*, Vol. 1, New York: Chelsea Publishers, 1960.

[34] J.S. Rose, *A Course on Group Theory*, Cambridge Univ. Press, 1978.

[35] J.J. Rotman, *An Introduction to the Theory of Groups*, Dubuque, Iowa: Wm.C. Brown Publishers, 1988.

[36] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error-Correcting Codes", *IEEE Trans. Inform. Theory*, Vol. IT-23, pp. 371-377, March 1977.

[37] V.V. Ginzburg, "Multidimensional Signals for a Continuous Channel", *Prob. Inform. Trans.*, Vol. 20, No. 1, pp. 20-34, 1984 (Translated from *Prob. Peredach. Inform.*, Vol. 20, No. 1, pp. 28-46, 1984).

[38] S.J. Sayegh, "A Class of Optimum Block Codes in Signal Space", *IEEE Trans. Commun.*, Vol. COM-34, pp. 1043-1045, October 1986.

[39] G.D. Forney, Jr., "Coset Codes I: Introduction and Geometrical Classification", *IEEE Trans. Inform. Theory*, Vol. 34, pp. 1123-1151, September 1988.

[40] G.D. Forney, Jr., "Coset Codes II: Binary Lattices and Related Codes", *IEEE Trans. Inform. Theory*, Vol. 34, pp. 1152-1187, September 1988.

[41] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, New York: Springer-Verlag, 1988.

[42] F.R. Kschischang and S. Pasupathy, "Some Ternary and Quaternary Codes and Associated Sphere Packings", *IEEE Trans. Inform. Theory*, Vol. 38, No. 2, March 1992.

[43] J. Portugheis, *Generalized Concatenated Codes for M-PSK Modulation*, Doctoral Thesis, T.H. Darmstadt, Germany, August 1992.

[44] R. Palazzo, Jr. and J.C. Interlando, "On the Relationship between the Diophantine Equations over the Ring of Algebraic Integers and the Mapping by Set Partitioning", *Coded modulation and bandwidth-efficient transmission: proceedings of the Fifth Tirrenia International Workshop on Digital Communications*, Tirrenia, Italy, September 8-12, 1991 / edited by E. Biglieri and M. Luise, pp. 79-90.

[45] I.F. Blake and R.C. Mullin, *The Mathematical Theory of Coding*, New York: Academic Press, 1975.