Pâmela Joyce Silva Melo Dantas

Códigos LDPC Definidos sobre Corpos de Inteiros Finitos

Campinas
2014

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Pâmela Joyce Silva Melo Dantas

Códigos LDPC Definidos sobre Corpos de Inteiros Finitos

Dissertação de mestrado apresentada ao Programa de Pós Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Orientador: Prof. Dr. Renato Baldini Filho

Este exemplar corresponde à versão final da dissertação de mestrado defendida pela aluna Pâmela Joyce Silva Melo Dantas, e orientada pelo Prof. Dr. Renato Baldini Filho

Campinas
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

D235c

Dantas, Pâmela Joyce Silva Melo, 1985-
Códigos LDPC definidos sobre corpos de inteiros finitos / Pâmela Joyce Silva Melo Dantas. – Campinas, SP : [s.n.], 2014.

Orientador: Renato Baldini Filho.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Códigos corretores de erros (Teoria da informação). 2. Teoria da codificação. I. Baldini Filho, Renato. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: LDPC codes defined over finite integer fields
Palavras-chave em inglês:
Brokers error codes (Information theory)
Coding theory
Área de concentração: Telecomunicações e Telemática
Titulação: Mestra em Engenharia Elétrica
Banca examinadora:
Renato Baldini Filho [Orientador]
José da Silva Barros
Lee Luan Ling
Data de defesa: 31-01-2014
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Pâmela Joyce Silva Melo

Data da Defesa: 31 de janeiro de 2014

Título da Tese: "Códigos LDPC Definidos sobre Corpos de Inteiros Finitos"

Prof. Dr. Renato Baldini Filho (Presidente): _____

Prof. Dr. José da Silva Barros: _____

Prof. Dr. Lee Luan Ling: _____

Resumo

Nesta dissertação apresentamos um estudo sobre a viabilidade de construção e de utilização de códigos LDPC (Low Density Parity Check) definidos sobre corpos finitos de inteiros módulo-p, onde p é um inteiro primo. A modulação utilizada para avaliar o desempenho dos códigos obtidos é a $p - PSK$. Códigos LDPC definidos sobre corpos finito de inteiros possuem estrutura algébrica bem definida, são facilmente feitos invariantes a rotação de fase da portadora no processo de modulação e podem ser feitos mais curtos que os seus equivalentes binários.

O método de decodificação iterativa utilizada na avaliação do desempenho destes códigos é uma adaptação do algoritmo SISO (Soft Input Soft Output) proposto por P. G. Farrell e J. Moreira [1] e [2] que utiliza a distância euclidiana como parâmetro de confiabilidade dos símbolos da palavra código recebida.

Os códigos LDPC utilizados na simulação da codificação e decodificação do canal de comunicação são definidos para o corpo de inteiros $\mathbb{Z}_5$. O canal de comunicação foi modelado com um ruído gaussiano branco aditivo (AWGN - Additive White Gaussian Noise) e com um desvanecimento Rayleigh. Ambos modelos de canal utilizam a modulação $5 - PSK$.

O desempenho dos esquemas de codificação LDPC definidos sobre $\mathbb{Z}_5$ foram analisados de modo comparativo com sistemas equivalentes de codificação binários e quartenários.

Palavras-chave: Códigos LDPC não binários, corpos de inteiros módulo-p, modulação $p - PSK$, canal AWGN, desvanecimento Rayleigh.

Abstract

On this disertation we present a study on the feasibility of constructions and use of LDPC (Low Density Parity Check) codes defined over finite fields of integers modulo p, where p is a prime integer. The modulation used to evaluate the performance of the codes is obtained from a $p - PSK$. LDPC codes defined over finite field of integers have well defined algebraic structure, they can be easily made invariant to phase rotation in the carrier modulation process, and can be made shorter than its binary equivalent.

The iterative decoding method used during the evaluating the performance of these codes is an adaptation of the algorithm SISO (Soft Input Soft Output) proposed by P. G. Farrell and J. Moreira [1] e [2] that uses the Euclidean distance as the reliability of the parameter code word symbols received.

The LDPC codes used during the simulation of encoding and decoding of the communication channel are defined for the whole body of $\mathbb{Z}_5$. The communication channel was modeled as additive white Gaussian noise (AWGN) and Rayleigh fading. Both communication channel models used modulation $5 - PSK$.

The performance of LDPC coding schemes defined over $\mathbb{Z}_5$ were analyzed comparatively with equivalent systems of binary and quaternary encoding.

Key-words: LDPC codes nonbinary, Field of integers modulo-p, $p - PSK$ modulation, AWGN Channel,Rayleigh fading.

Sumário

Aos meus pais, meus irmãos e meu esposo.

Agradecimentos

Ao concluir este trabalho, agradeço,

A Deus, que me deu a vida e o presente de ter chegado até aqui, por ter sempre me abençoado.

Aos meus pais Jinaldo Melo e Francisca Melo que com muito esforço me possibilitaram sonhar e mesmo sem perceber, ao mostrar o orgulho e confiança que depositaram em mim, me davam forças para continuar, e aos meus meus irmãos Deyweson e Kamylla.

Ao meu esposo Adjário pela paciência, amor, carinho e incentivo durante este trabalho.

Em especial ao Prof. Dr. Renato Baldini Filho, pela oportunidade, pela atenção e orientação, pelo incetivo, dedicação, susgestão e correção que tornaram possível esse trabalho.

Aos Professores do Departamento de Comunicações - Decom e do Departamento de Telemática-DT, pela formação.

Aos meus amigos de Departamento, Esdras, Rodolpho, Alice, Sarah, Juan, Carol, Adailton, Roberto, Cristian, Lucas e Ana Carolina, pelos bons momentos no laboratório e pela convivência que de certa forma contribuíram para a realização desse trabalho.

Ao meu amigo Mario pela amizade no começo dessa caminhada, pelo incentivo, esclarecimentos e constante apoio durante o mestrado.

Aos Professores da banca examinadora Prof. Dr. José da Silva Barros(UFAL) e Prof. Dr. Lee Luan Ling(FEEC-UNICAMP).

Ao Cnpq pelo apoio financeiro.

”Uma das coisas mais difíceis não é mudar a sociedade; é mudar a si mesmo.”

Nelson Mandela

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos e Notação

AWGN	Additive White Gaussian Noise (Ruído Aditivo Gaussiano Branco)
BER	Bit Error Rate (Taxa de Erro de Bit)
BP	Belief Propagation
LDPC	Low Density Parity Check (Códigos de Verificação de Paridade de Baixa Densidade)
MatLab	Matrix Laboratory
PSK	Phase Shift Keying (Modulação por Deslocamento de Fase)
SPA	Sum Product Algorithm
SISO	Soft Input and Soft Output

C	Capacidade do Canal
p	Alfabeto p-ário, p é primo
q	Alfabeto q-ário
$\mathbf{m}$	Informação transmitida
$\mathbf{c}$	Palavra-código
k	Comprimento da informação
n	Comprimento da palavra-código
$\mathbf{G}$	Matriz Geradora
$\mathbf{H}$	Matriz de Verificação de Paridade
$\mathbf{H}^T$	Transposta da Matriz $\mathbf{H}$
$\mathbf{H}_5$	Matriz de verificação de paridade sistemática com elementos em $\mathbb{Z}_5$
$\mathbf{G}_5$	Matriz geradora sistemática com elementos em $\mathbb{Z}_5$
P	Matriz paridade
I_k	Matriz identidade
$\mathbb{C}$	Código de bloco linear
$\mathbb{Z}_p$	Corpo de inteiros finito módulo-p
$\mathbb{Z}_5$	Corpo de inteiros finito módulo-5
$\mathbb{Z}_4$	Anel de inteiros módulo-4
x	Elementos pertencentes ao corpo de inteiros $\mathbb{Z}_5$
$\mathbf{r(t)}$	Sinal recebido
$\mathbf{c(t)}$	Sinal modulado transmitido
$\mathbf{n(t)}$	Ruído gaussiano
$\mathbf{d(t)}$	Desvanecimento
d_x^2	Distância Euclidiana quadrada do símbolo x
D_E^2	Distância Euclidiana quadrada
W_E^2	Peso Euclidiano quadrado

R_c Taxa de Codificação
w_c Peso de coluna
w_r Peso de linha
σ Variância do ruído AWGN

Capítulo 1

Introdução

A teoria da codificação nasceu em 1948, com o famoso trabalho de Shannon [3] que definiu um limitante para a taxa de transmissão de informação confiável em um sistema de comunicações, denominada de capacidade de canal. Em linhas gerais, este resultado diz que se a transmissão de informação ocorrer abaixo de uma certa taxa C (bits por segundo), chamada de capacidade do canal, é sempre possível obter uma probabilidade de erro de bit tão pequena quanto se deseja, através da utilização de códigos corretores de erro eficientes. A capacidade para um canal com ruído gaussiano branco aditivo (AWGN) é dada por

$$C = B \log_2 \left(1 + \frac{P}{N}\right) \tag{1.1}$$

onde P e N são as potências dada em Watts do sinal enviado e do ruído branco na largura de faixa disponível B Hertz, respectivamente. O canal AWGN é um modelo simples de canal de comunicação que possui densidade espectral de potência constante e uma função de distribuição de probabilidades gaussiana.

Os códigos de verificação de paridade de baixa densidade, inventados por Gallager [4] em 1963, é uma destas classes de códigos que alcança desempenho extremamente próximo do limitante de Shannon. Os códigos LDPC são especificados por uma matriz de verificação contendo muitos zeros e pouquíssimos uns, seguindo uma estrutura pré-estabelecida de posicionamento destes poucos uns nesta matriz. Infelizmente, na época não havia disponível um processo de decodificação eficiente e processadores que realizassem os cálculos necessários para a decodificação a uma velocidade aceitável. Isto fez com que estes códigos ficassem esquecidos por mais de 30 anos. A primeira retomada do interesse sobre os códigos LDPC se deu quase 20 anos depois, em 1981, quando Tanner generalizou os códigos LDPC e criou um grafo bipartido para representá-los de forma mais efetiva.

Entretanto, somente em 1995, com Mackay [5] e Neal [6], estes códigos foram, de fato, considerados factíveis de decodificação com a introdução de processos iterativos de decodificação. O desempenho muito próximo da capacidade dos esquemas de codificação LDPC é alcançado devido ao longo comprimento do código, sua natureza (pseudo) aleatória e pelo processo de

decodificação iterativo.

1.1 Motivação

Os códigos LDPC definidos sobre corpos de inteiros $\mathbb{Z}_p$ (p um número primo) e anéis de inteiros $\mathbb{Z}_q$, apresentam algumas características interessantes que podem ser exploradas para minimizar as deficiências dos códigos binários, tais como: o perfeito casamento com os símbolos da modulação $p-PSK$, podem ser facilmente feitos invariantes a rotações de fase da portadora e possuem menor comprimento das palavras código que os códigos LDPC binários equivalentes. Além disso, os códigos LDPC definidos sobre $\mathbb{Z}_p$ permitem que a redundância esteja presente não só nos $n-k$ símbolos de paridade mas também em alguns dos símbolos de $\mathbb{Z}_p$, dando assim um grau de liberdade maior de taxa de codificação.

Como existem poucos estudos na literatura sobre códigos LDPC definidos sobre corpos de inteiros $\mathbb{Z}_p$, este trabalho de pesquisa visa propor um método de obtenção destes códigos aplicados a modulações $p-PSK$. A busca de códigos LDPC se restringiu ao corpo finito $\mathbb{Z}_5$ de inteiros módulo-5. O desempenho dos códigos LDPC obtidos foram comparados com códigos LDPC binários e quartenários equivalentes.

1.2 Organização da Dissertação

Esta dissertação está dividida em 5 capítulos, onde os conteúdos estão dispostos como descrito abaixo.

No capítulo 1, são apresentados uma breve introdução a teoria da codificação e aos códigos de verificação de paridade de baixa densidade, a motivação deste trabalho e a organização da tese.

No capítulo 2, são apresentados os conceitos básicos de Álgebra Abstrata e Linear dentre eles as definições de Grupos, Anéis, Corpos e Espaços Vetoriais. Também são abordados conceitos dos códigos de blocos lineares, sua descrição matricial e a taxa de codificação de um codificador multinível. Descrevemos ainda a modulação $p-PSK$, o calculo da distância Euclidiana entre dois sinais e os canais de transmissão utilizados.

No capítulo 3, foram introduzidos os códigos LDPC binários regulares e irregulares, definimos códigos LDPC sobre os corpos finitos de inteiros $\mathbb{Z}_p$ módulo p e a sua construção, abordamos também a decodificação dos códigos LDPC utilizando o Algoritmo SISO(Soft-Input Soft-Output) e por fim será apresentado um exemplo para ilustrar a codificação e decodificação de um código de bloco 5-ário.

No capítulo 4, são apresentados os modelos dos sistemas de comunicações, as medidas de desempenho utilizadas nas simulaçãoes de ambos os códigos LDPC: binário, quartenário e 5-

ário para várias situações de transmissão pelo canal e finalmente é apresentado os resultados das simulações.

No capítulo 5, são apresentadas as conclusões deste trabalho e algumas propostas de trabalhos futuros.

Capítulo 2

Conceitos Básicos

Neste capítulo serão introduzidos alguns conceitos básicos de Álgebra Abstrata e Álgebra Linear, conceitos matemáticos de fundamental importância para o desenvolvimento de códigos de blocos lineares. Serão introduzidas definições de Grupos, Anéis, Corpo e Espaços vetoriais. Também faremos uma introdução à Modulação $p - PSK$ e distâncias Euclidianas entre dois sinais $p-PSK$. Por fim, abordaremos os conceitos de Canais de Transmissão, tais como: Canal AWGN e Desvanecimento Rayleigh.

2.1 Grupos

Definição 2.1.1 *Um grupo $(G, +)$ é um conjunto não vazio G, com uma operação binária sobre G que satisfaz as seguintes propriedades, para todo a, b e $c \in G$:*

1. *Fechamento, $a + b \in G$.*
2. *A operação binária $+$ é associativa, $(a + b) + c = a + (b + c)$.*
3. *Existe um elemento neutro, $0 \in G$, tal que $0 + a = a + 0 = a$.*
4. *Para todo elemento $a \in G$, existe um elemento $a' \in G$, tal que $a + a' = a' + a = 0$.*

Definição 2.1.2 *Um Grupo G é abeliano se a operação binária $(+)$ for comutativa, isto é, $a + b = b + a$, para todo a e $b \in G$.*

2.2 Anel

Definição 2.2.1 *Um anel $(A, +, *)$ é um conjunto A com duas operações, adição $(+)$ e multiplicação $(*)$ que satisfaz as seguintes condições, para todo $a, b \in A$:*

1. *$(A, +)$ é um grupo abeliano.*
2. *A é fechado sobre a operação de multiplicação, isto é, $a * b \in A$.*
3. *A multiplicação é associativa, isto é, $(a * b) * c = a * (b * c)$.*

4. *Para todo $a, b, c \in A$, vale a lei da distribuição, tanto à direita, $(a + b) * c = a * c + b * c$, como à esquerda $a * (b + c) = a * b + a * c$.*

Definição 2.2.2 *Um anel A é dito comutativo com unidade se sua multiplicação é comutativa e existe um elemento neutro $1 \in A$, tal que, para todo $a, b \in A$, temos $a * b = b * a$ e $1 * a = a$, para todo $a \in A$.*

2.3 Corpos

Definição 2.3.1 *Um anel A, comutativo com unidade, é chamado de corpo se todo elemento não nulo de A admite simétrico multiplicativo, ou seja, para todo $a \in A$, $a \neq 0$ implica que existe $b \in A$, tal que, $a * b = 1$.*

Definição 2.3.2 *Um anel finito de inteiros módulo-q, é definido pelo conjunto $\mathbb{Z}_q = \{0, 1, 2, \ldots, q - 1\}$ com as operações de soma e produto módulo-q.*

Definição 2.3.3 *Se q for igual a p, um inteiro primo, dizemos que esse conjunto é um corpo de inteiros módulo-p.*

Um exemplo de corpo de inteiros módulo-5 $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$ com as operações de soma e multiplicação módulo-5 é dada pelas tabelas abaixo.

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Tabela 2.1: Operação de adição sobre o corpo de inteiros módulo-5

*	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Tabela 2.2: Operação de multiplicação sobre o corpo de inteiros módulo-5

2.4 Espaços vetoriais

Definição 2.4.1 *Seja K um corpo. Um* ***espaço vetorial*** *sobre K consiste de um grupo abeliano V sob adição junto com uma operação de multiplicação por escalar de elementos de V por elementos de K à esquerda, tal que para todo $a, b \in K$ e $\alpha, \beta \in V$, valem as seguintes propriedades:*

1. $a\alpha \in V$;
2. $a(b\alpha) = (ab)\alpha$;
3. $(a+b)\alpha = (a\alpha) + (b\alpha)$;
4. $a(\alpha+\beta) = a\alpha + a\beta$;
5. $1\alpha = \alpha$, *onde* 1 *é o elemento neutro da multiplicação em* K.

Definição 2.4.2 *Seja V um espaço vetorial sobre um corpo K. Um subconjunto $W \subseteq V$, não vazio, será um subespaço vetorial de V se as seguintes propriedades forem satisfeitas:*

1. $\forall \alpha, \beta \in V, \alpha + \beta \in W$;
2. $\forall a \in K, \forall \alpha \in V, a\alpha \in W$.

Por exemplo, as n-uplas de elementos em K_q formam um espaço vetorial sobre o corpo K_q. Como veremos futuramente, um código linear de comprimento n sobre K_q nada mais é do que um subespaço vetorial do espaço de todas as n-uplas K_q^n.

Definição 2.4.3 *Seja V um espaço vetorial sobre o corpo K. Os vetores em um subconjunto $S = \{u_i | i \in I\}$ (onde I é um conjunto de índices) geram V se $\forall u \in V$, temos que:*

$$\underline{\boldsymbol{u}} = a_1 \cdot \underline{u}_{i_1} + a_2 \cdot \underline{u}_{i_2} + \cdots + a_n \cdot \underline{u}_{i_n}$$

para algum conjunto de $a_j \in K$ e $\underline{\boldsymbol{u}}_{i_j} \in S, j = 1, 2, \cdots, n$. Um vetor

$$\underline{\boldsymbol{u}} = \sum_{j=1}^{n} a_j \underline{\boldsymbol{u}}_{i_j}$$

é chamado de Combinação Linear *dos $\underline{\boldsymbol{u}}_{i_j}$*

Definição 2.4.4 *Um espaço vetorial V sobre um corpo K tem* dimensão finita *se existe um subconjunto finito de V cujas combinações lineares entre seus vetores geram V.*

Definição 2.4.5 *Os vetores em um subconjunto $S = \{u_i | i \in I\}$ de um espaço vetorial V sobre um corpo K são* linearmente independentes *sobre K se*

$$\sum_{j=1}^{n} a_j \underline{\boldsymbol{u}}_{i_j} = 0$$

implica que $a_j = 0,\ j = 1, 2, \cdots, n$. Dessa forma, se os vetores não são linearmente indepentes sobre K, dizemos que eles são linearmente dependentes *sobre K.*

Definição 2.4.6 *Se V é um espaço vetorial sobre um corpo K, os vetores em um subconjunto $B = \{u_i | i \in I\}$ de V formam uma* base *para V e são linearmente independentes.*

O conjunto da n-uplas sobre K_q forma um espaço vetorial de dimensão n, na qual a base é dada pela base canônica n dimensional.

2.4.1 Espaço Vetorias de n-uplas

Dado um corpo K, uma n-upla é uma sequência $(a_1, a_2, \cdots, a_n)$ tal que $a_i \in K, 1 \leq i \leq n$. O conjunto de todas as n-uplas sobre K_q^n forma um espaço vetorial sob "adição componente-a-componente" e "multiplicação por escalar componente-a-componente". Denota-se por K_q^n o conjunto formado por todas as n-uplas. Espaços vetoriais de dimensão finita podem ser tratados como espaços vetoriais de n-uplas.

Definição 2.4.7 *O produto interno de duas n-uplas $\underline{\boldsymbol{u}} = (u_1, u_2, \cdots, u_n)$ e $\underline{\boldsymbol{v}} = (v_1, v_2, \cdots, v_n)$ é o escalar definido por*

$$\underline{\boldsymbol{u}} \cdot \underline{\boldsymbol{v}} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n.$$

Definição 2.4.8 *Dois vetores $\underline{\boldsymbol{u}}$ e $\underline{\boldsymbol{v}}$ em K_q^n são ortogonais se $\underline{\boldsymbol{u}} \cdot \underline{\boldsymbol{v}} = 0$.*

Definição 2.4.9 *O subespaço de vetores ortogonais a um subespaço W é denominado complemento ortogonal de W e é denotado por $W^{\perp}$.*

2.5 Códigos de Blocos

Dado um corpo $\mathbb{Z}_q$. Um código linear $\mathbb{C}(n, k)$ é um sub-espaço de $\mathbb{Z}_q^n$, isto é, um código linear é um conjunto não vazio de n-uplas sobre $\mathbb{Z}_q$ chamadas *palavras-código*, tal que a soma de duas palavras-código quaisquer é uma palavra-código e o produto de qualquer palavra-código por um elemento de $\mathbb{Z}_q$ é uma palavra-código.

2.5.1 Descrição Matricial

Seja $\mathbb{Z}_p$ um corpo de inteiros com p elementos e $\mathbb{C} \subset \mathbb{Z}_p^n$ um código de bloco linear (n, k), uma palavra-código $\mathbf{c}$ é um vetor de dimensão n formada por símbolos de $\mathbb{Z}_p$,

$$\mathbf{c} = [c_0, c_1, \cdots, c_{n-1}]$$

.

A mensagem $\mathbf{m}$ a ser codificada é formada por um vetor de dimensão k, definido também com símbolos pertencentes a $\mathbb{Z}_p$,

$$\mathbf{m} = [m_0, m_1, \cdots, m_{k-1}].$$

As palavras código $\mathbf{c}$ podem ser geradas a partir de uma matriz $\mathbf{G}$, de dimensão $k \times n$, que é a *matriz geradora do código*. Os espaços das linhas de $\mathbf{G}$ é o código linear $\mathbb{C}$. As linhas da matriz $\mathbf{G}$ são linearmente independentes, onde k-linhas é a dimensão do sub-espaço. Assim, $\mathbf{c} = \mathbf{m} \cdot \mathbf{G}$, onde $\mathbf{m}$ (informação) é uma k-upla dos símbolos de informação a serem codificados e a n-upla $\mathbf{c}$ é a palavra-código.

Definição 2.5.1 *Uma matriz geradora $\mathbf{G}$, de dimensões $k \times n$, de um código $\mathbb{C}$ está na forma sistemática se tivermos $\mathbf{G} = [\boldsymbol{P}|\boldsymbol{I}_k]$, onde $\boldsymbol{I}_k$ é a matriz identidade $k \times k$ e $\boldsymbol{P}$ é a matriz de paridade de dimensão $k \times (n-k)$.*

Para toda palavra-código $\mathbf{c}$ vale a relação $\mathbf{c} \cdot \mathbf{H}^T = 0$, onde a matriz $\mathbf{H}$ de dimensão $(n-k) \times n$, é a matriz de verificação de paridade, qualquer vetor ortogonal as suas linhas pertence ao espaço vetorial das linha de $\mathbf{G}$ associada.

Definição 2.5.2 *A matriz de verificação de paridade $\boldsymbol{H}$ $(n-k) \times n$, pode ser obtida por, $\boldsymbol{H} = [\boldsymbol{I}_{(n-k)}| - \boldsymbol{P}^T]$, onde $\boldsymbol{I}_{(n-k)}$ é a matriz identidade de dimensão $(n-k) \times (n-k)$ e $-\boldsymbol{P}^T$ é a matriz transposta de $\boldsymbol{P}$ com os seus elementos trocados pelos seus respectivos inversos aditivos.*

Dessa forma, o código obtido através da matriz $\mathbf{H}$ é chamado de *código dual* $\mathbb{C}^{\perp}$ do código linear $\mathbb{C}$, ou seja, a matriz $\mathbf{H}$ é o complemento ortogonal da matriz $\mathbf{G}$, portanto, vale a seguinte relação:

$$\mathbf{G} \cdot \mathbf{H}^T = 0 \tag{2.1}$$

2.5.2 Taxa de Codificação

Neste trabalho utilizamos um codificador multinível que possui k entradas com símbolos pertencentes ao alfabeto q-ário e n saídas com símbolos do alfabeto p-ário. Assim, a taxa de codificação é dada por:

$$R_c = \frac{\log q^k}{\log p^n} = \frac{k \log q}{n \log p}. \tag{2.2}$$

Note que se a cardinalidade do alfabetos forem iguais, ou seja, $q = p$, então $R_c = \frac{k}{n}$.

.

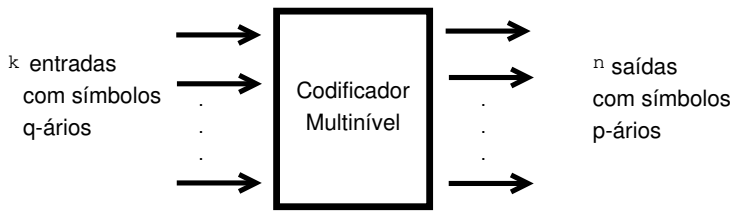

Figura 2.1: Diagrama do Codificador Multinível

2.6 Modulação $p - PSK$

Os símbolos p-ários na saída do codificador são geralmente associados a uma modulação $p-PSK$, devido ao perfeito casamento com os elementos que pertencem a um corpo de inteiros finito $\mathbb{Z}_p$ da codificação.

A constelação $p - PSK$ é representada pela Fig. 2.2 abaixo:

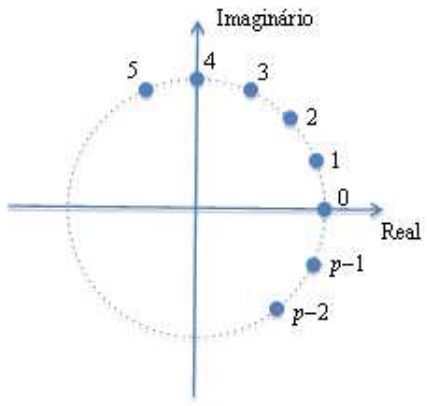

Figura 2.2: Modulação $p - PSK$

A modulação $5 - PSK$ é realizada fazendo-se o mapeamento dos símbolos $(0, 1, 2, 3, 4)$ nas coordenadas:

$$\left\{ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \right\} \approx \left\{ \begin{array}{c} \cos(0) + j\sin(0) \\ \cos(\frac{2\pi}{5}) + j\sin(\frac{2\pi}{5}) \\ \cos(\frac{4\pi}{5}) + j\sin(\frac{4\pi}{5}) \\ \cos(\frac{6\pi}{5}) + j\sin(\frac{6\pi}{5}) \\ \cos(\frac{8\pi}{5}) + j\sin(\frac{8\pi}{5}) \end{array} \right\} \tag{2.3}$$

O sistema de coordenadas $5 - PSK$ é descrito em termos de coordenadas (x, y) em que x é representado por I (em fase) , ou seja, eixo real e y é representado por Q (quadratura), eixo imaginário. Cada símbolo da constelação $5 - PSK$ pode carregar 2 bits. O mapeamento dos bits para símbolo da modulação pode ser feito utilizando o código de Gray, que garante que os símbolos vizinhos na constelação diferenciam-se em apenas um bit, minimizando assim a taxa de erro de bit do sistema de comunicação. O mapeamento de bits em símbolo de modulação é mostrado na Fig. 2.3. Note que para o símbolo 4 não é definido os bits transmitidos, uma vez que vamos utilizar este símbolo apenas como redundância, o que faz com que ele não carregue informação.

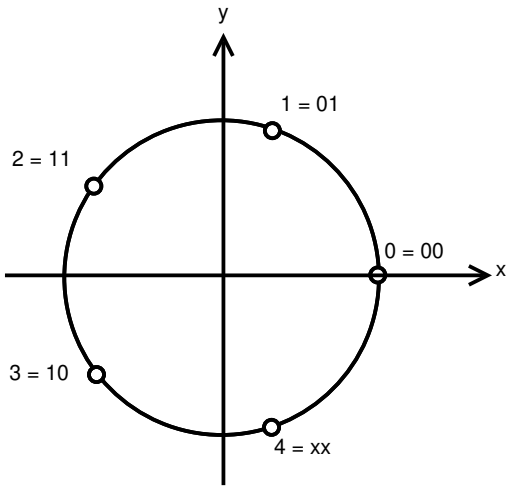

Figura 2.3: Representação dos símbolos da modulação $5 - PSK$

2.7 Distância Euclidiana entre Dois Sinais $p - PSK$

A sequência de entrada do modulador é composta por símbolos pertencentes ao corpo de inteiros $\mathbb{Z}_p$. Estes símbolos são transformados através da modulação $p-PSK$ em sinais do tipo:

$$S_i = A\exp(j\frac{2\pi i}{p}) = A(\cos(j\frac{2\pi i}{p}) + j\sin(j\frac{2\pi i}{p})), \tag{2.4}$$

onde i é um símbolo do alfabeto p-ário $\{0, 1, 2, \ldots, p-1\} \in \mathbb{Z}_p$, $j = \sqrt{(-1)}$, e A é uma amplitude (raio) do sinal, consideramos a amplitude do sinal $A = 1$, sem perda de generalidade.

A distância Euclidiana ao quadrado entre dois sinais S_i e S_r é definida por:

$$\begin{aligned} D_E^2(S_i, S_r) &= ||S_i - S_r||^2 \\ &= \left|\left|\exp\left(j\frac{2\pi i}{p}\right) - \exp\left(j\frac{2\pi r}{p}\right)\right|\right|^2 \\ &= \left|\left|\exp\left(j\frac{2\pi(i-r)}{p}\right) - 1\right|\right|^2 \\ &= \left|\left|\exp\left(j\frac{2\pi(i\ominus r)}{p}\right) - 1\right|\right|^2, \end{aligned} \tag{2.5}$$

onde $\ominus$ denota a subtração módulo-p. Desse modo, a distância Euclidiana ao quadrado entre dois pontos no espaço de sinais modulados em $p - PSK$ é dada pela expressão:

$$D_E^2(i, r) \triangleq \left|\left|\exp\left(j\frac{2\pi(i\ominus r)}{p}\right) - 1\right|\right|^2. \tag{2.6}$$

E o peso Euclidiano ao quadrado é definido por:

$$W_E^2(i) \triangleq D_E^2(i, 0) \triangleq \left|\left|\exp\left(j\frac{2\pi i}{p}\right) - 1\right|\right|^2. \tag{2.7}$$

Dessa forma, a distância Euclidiana ao quadrado entre dois sinais é igual ao peso Euclidiano ao quadrado do resultado da subtração módulo-p, entre estes dois pontos, isto é,

$$D_E^2(i,r) \triangleq W_E^2(i \ominus r). \tag{2.8}$$

Note que o índice i é equivalente a $S_i = \exp\left(j\frac{2\pi i}{p}\right)$. Logo, a relação fechada que existe entre o corpo de inteiros módulo-p e a modulação $p-PSK$ permite o perfeito casamento dos símbolos do alfabeto p-ário com a modulação $p-PSK$.

Portanto, podemos obter a distância Euclidiana ao quadrado entre n-uplas x e y, com componentes pertencentes ao corpo de inteiros sobre $\mathbb{Z}_p$ por

$$D_E^2(x,y) \triangleq \sum_{i=1}^{n} \left|\left|\exp\left(j\frac{2\pi(x_i - y_i)}{p}\right) - 1\right|\right|^2. \tag{2.9}$$

À medida em que a cardinalidade do alfabeto p-ário aumenta, aumenta também a quantidade de informação associada a cada símbolo, no entanto, diminui a distância Euclidiana entre os símbolos não idênticos mais próximos.

2.8 Canal de Transmissão

Em um sistema de comunicação móvel ocorrem algumas pertubações que podem alterar um sinal de comunicação. Neste trabalho será considerada a existência de ruído AWGN (*Additive White Gaussian Noise*) e o desvanecimento Rayleigh.

2.8.1 Canal com ruído AWGN

O canal adiciona ruído branco que está presente em toda faixa de frequências com densidade espectral bilateral de potência constante $\frac{N_0}{2}$ $\mathbf{W/Hz}$ o ruído é caracterizado por uma função de distribuição de probabilidades gaussiana de média zero e a variância σ^2 . Esse ruído é independente do sinal enviado. Assim, o sinal recebido é caracterizado por $\mathbf{r(t)} = \mathbf{c(t)} + \mathbf{n(t)}$, onde $\mathbf{c(t)}$ é o sinal transmitido e $\mathbf{n(t)}$ é o ruído aditivo gaussiano branco. Na Fig. 2.4 pode ver a representação de um canal com ruído AWGN.

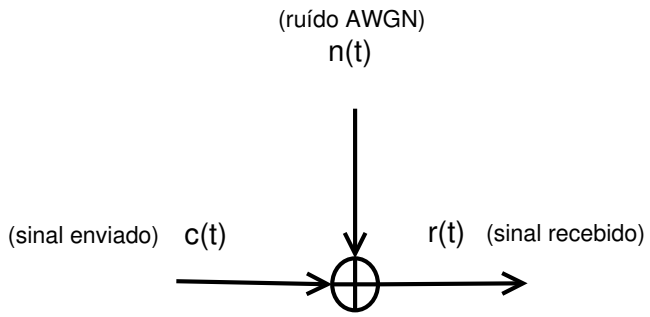

Figura 2.4: Representação de um canal com ruído AWGN

2.8.2 Canal com Desvanecimento

O sinal transmitido pode sofrer algumas flutuações na sua amplitude, porque ele percorre diferentes caminhos até chegar ao receptor. Nestes percusos ele sofre os efeitos da reflexão nas superfíceis, prédios, carros, etc., ocasionando um desvanecimento (*fading*) no sinal recebido. O desvanecimento é causado pela soma de versões do sinal transmitido que chegam ao receptor com diferentes atenuações e atrasos.

A banda de coerência explica como o canal modifica o sinal no domínio da frequência, assim se banda de coerência do canal for maior que a banda ocupada pelo sinal transmitido, ao passar pelo canal este sinal sofrerá desvanecimento plano, dessa forma, todas as componentes de frequência serão atenuadas uniformimente.

O desvanecimento é caracterizado pela amplitude com a função de densidade de probabilidade do tipo Rayleigh em ambientes abertos e sem linha de visada direta, essa será a distribuição a ser considerada neste trabalho, a função densidade de probabilidade de Rayleigh é dada por:

$$f(y) = \frac{y}{\alpha^2} \exp^{-\frac{y^2}{2\alpha^2}} \tag{2.10}$$

onde α^2 é a variância de y.

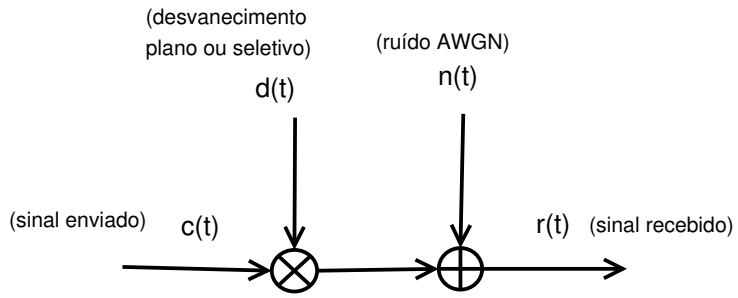

Figura 2.5: Representação de um canal com desvanecimento e ruído AWGN

Na fig. 2.5, temos que o sinal recebido é dado por: $\mathbf{r(t) = c(t) \cdot d(t) + n(t)}$, onde $\mathbf{c(t)}$ é o sinal modulado transmitido que é multiplicado pelo desvanecimento $d(t)$ e depois é somado o ruído $\mathbf{n(t)}$.

O desvanecimento $\mathbf{d(t)}$ é caracterizado por um fator Δ é a envoltória do desvanecimento que modifica a amplitude do sinal dada pela equação 2.11, formada por duas variáveis aleatórias gaussianas independentes e identicamente distribuídas.

$$\Delta = \sqrt{x_1^2 + x_2^2} \tag{2.11}$$

onde x_1^2 e x_2^2 são respectivamente as componentes em fase e quadratura do sinal recebido, essas componentes são variáveis aleatórias gaussianas com média zero e a mesma variância α^2 [7]. A

energia média da envoltória do desvanecimento está associada com a variância, em fase e em quadratura, com valor:

$$E[\Delta^2] = 2\alpha^2. \tag{2.12}$$

Capítulo 3

Códigos LDPC

Neste capítulo faremos uma introdução aos códigos LDPC binários, regulares e irregulares, definiremos códigos LDPC sobre corpos de inteiros finitos $\mathbb{Z}_p$ e a construção dos códigos LDPC definidos sobre $\mathbb{Z}_5$. Também abordaremos a decodificação desses códigos utilizando o algoritmo SISO, por fim, apresentamos um exemplo numérico da aplicação do algoritmo SISO na decodificação de um código de bloco 5-ário.

3.1 Introdução aos Códigos LDPC Binários

Um código LDPC (Low Density Parity Check) binário [8, 9] é visto como sendo um código de bloco linear binário (n, k, w_c), com o peso das colunas w_c da matriz de verificação $\mathbf{H}$ muito menor que $m = n - k$. A matriz $\mathbf{H}$ possui dimensões $(n - k) \times n$, contendo um número w_c de $1's$ por coluna e um número $w_r = w_c \cdot \frac{n}{m}$ de $1's$ por linha. Se $w_c \geq 3$, então o conjunto de códigos LDPC obtidos, em sua maioria, possuem distância de Hamming mínima elevada, o que leva estes códigos a apresentarem um desempenho muito próximo ao Limite de Shannon. Devido a essa grande distância mínima, os códigos LDPC apresentam uma matriz $\mathbf{H}$ esparsa, ou seja, esta matriz possui baixa densidade de elementos não nulos nas suas colunas e linhas.

Os códigos LDPC foram introduzidos por R. G. Gallager [4], no início dos anos 60 e são capazes de alcançar um desempenho próximo a capacidade de canal em diversos modelos de canais. Os códigos LDPC eram muito longos para a tecnologia da época, pois os computadores não eram capazes de simular o desempenho de códigos com comprimentos significativos e com baixas densidades de erro. Além disso, a elevada complexidade computacional requerida tanto na construção da matriz $\mathbf{H}$, que garantisse uma boa distância mínima do código, como no processo de codificação e de decodificação tornavam a implementação prática destes códigos proibitivas. Isso fez com que os códigos LDPC fossem deixados de lado pelos pesquisadores por muito tempo. Somente em 1981, R.M. Tanner [10] generalizou o trabalho de Gallager e introduziu uma representação gráfica dos códigos LDPC, denominada de grafos bipartidos.

Em meados da década de 90, os códigos LDPC foram novamente resgatados, por D.J.C.

Mackay [5], R.M. Neal [6] e outros [11]. Mackay mostrou que os códigos LDPC longos, quando decodificados com o algoritmo soma-produto, eram capazes de atingir um desempenho muito próximo ao limite de Shannon [3] em um canal AWGN. Com isso os códigos LDPC voltaram a ser intensamente estudados e utilizados para controle de erro em um grande número de sistemas de comunicação e armazenamento de dados.

Apesar da eficiência dos códigos LDPC binários, estes apresentam algumas deficiências, principalmente em aplicações que demandam transmissão em tempo real, devido ao seu longo comprimento e ao atraso provocado pelo processo de decodificação iterativo.

Códigos LDPC não binários, definidos sobre anéis ou corpos de inteiros, apresentam algumas características interessantes que podem ser exploradas para minimizar as deficiências dos códigos binários, tais como: o perfeito casamento com os símbolos da modulação $p - PSK$, podem ser feitos facilmente invariantes a rotações de fase da portadora e podem possuir menor comprimento das palavras código que os códigos LDPC binários equivalentes.

3.2 Grafos de Tanner

Os códigos LDPC são geralmente apresentados em uma forma gráfica, conhecida como grafos de Tanner [10], ou como um grafo bipartido. Os grafos de Tanner possuem dois tipos de nós: os nós de paridade e os nós de símbolos. Os nós de símbolos estão relacionados aos símbolos que chegam ao decodificador, ou seja, referem-se ao número de colunas da matriz **H**, enquanto que os nós de verificação estão relacionados ao número de linhas desta matriz. Estes dois tipos de nós são ligados por linhas de acordo com os elementos não nulos da matriz de verificação de paridade **H**.

Ao representarmos um código de bloco linear (n, k) por grafos, temos que em um dos lados do grafo são colocados n nós de símbolos c_j, e no outro lado são colocados $n - k$ nós de verificação de paridade f_i, o número de ligações que parte de cada nó de símbolo em direção aos nós de paridade é determinado pelo número de elementos não nulos de cada coluna da matriz **H** e o número de ligações que parte de cada nó de paridade em direção aos nós de símbolos é determinado pelo número de elementos não nulos de cada linha da matriz **H**. Dessa forma, as posições dos elementos não nulos da matriz **H** determinam as interconexões entre os nós de símbolos e os nós de paridade, o número de linhas que confluem em um dado nó é chamado de grau do nó [12].

Como exemplo, considere um código (8,4) binário cuja representação matricial é dada por:

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \tag{3.1}$$

onde $w_r = 4$ é o número de elementos não nulo de cada linha e $w_c = 2$ é o número de elementos não nulo de cada coluna. A representação gráfica desse código $(8, 4)$ é dada pela Fig. 3.1.

Um código de bloco linear (n, k), pode ser representado por um sistema de $n - k$ equações lineares homogêneas obtidas a partir da matriz de verificação de paridade. Um grafo bipartido pode então ser construído a partir dessas equações. As equações de paridade obtidas da matriz **H** em 3.1 são:

$$\begin{array}{rcrcrcrcl} x_{12} & + & x_{14} & + & x_{15} & + & x_{18} & = & 0 \\ x_{21} & + & x_{22} & + & x_{23} & + & x_{26} & = & 0 \\ x_{33} & + & x_{36} & + & x_{37} & + & x_{38} & = & 0 \\ x_{41} & + & x_{44} & + & x_{45} & + & x_{47} & = & 0 \end{array} \tag{3.2}$$

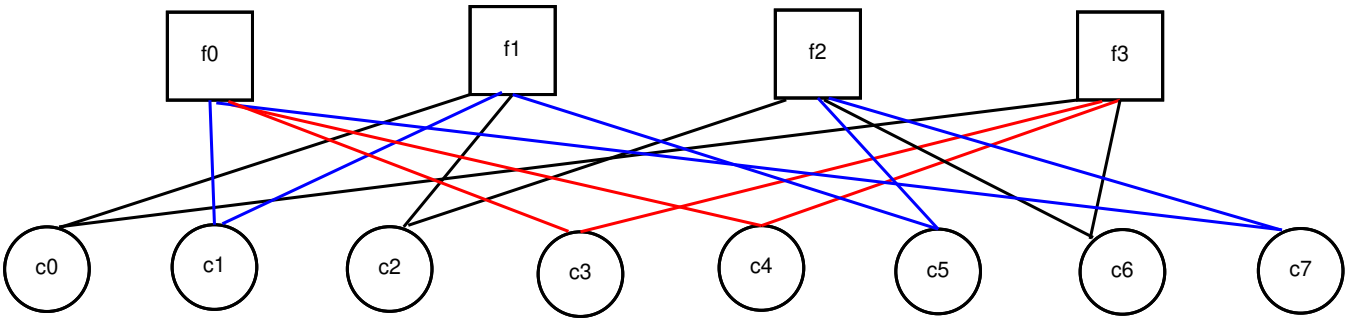

Figura 3.1: Grafo de Tanner para um código de bloco $(8, 4)$

O grafo possui n nós de símbolo e $n - k$ nós de paridade, assim um nó de paridade f_i é conectado por uma aresta a um nó de símbolo c_j se o elemento h_{ij} da matriz de verificação de paridade for não nulo.

Os grafos de Tanner possuem algumas propriedades específicas relacionadas ao desempenho de correção de erro do código, como:

- *Ciclo* é uma sequência de nós e arestas conectadas, no qual o primeiro nó é também o nó final e não existe repetição de nó dentro da sequência.
- O tamanho de um ciclo é determinado pelo número de arestas que o ciclo possui.
- *Girth* é o menor comprimento de todos os ciclos existentes num grafo de Tanner.

Dessa forma, no exemplo acima, temos que a matriz **H**, possui um *Girth* de tamanho igual a 4. Na prática ao se projetar um código LDPC busca-se evitar a existência de ciclos curtos de forma a melhorar o desempenho do algoritmo de decodificação denominado *Soma-Produto*.

3.3 Códigos LDPC Binários Regulares e Irregulares

O método de construção de códigos LDPC binários, proposto por Gallager, consiste em formar uma matriz de verificação de paridade esparsa **H** de modo aleatório.

Um código LDPC regular é um código com uma matriz **H** de verificação de paridade esparsa que possui número fixo de 1's por coluna e por linha. Já um código LDPC irregular possui números distintos de 1's por linhas e ou por colunas. Em geral, o desempenho, em termos de taxa de erro de bit, dos códigos LDPC irregulares são melhores do que as dos códigos LDPC regulares [13].

3.4 Códigos LDPC Definidos sobre Corpos Finitos de Inteiros $\mathbb{Z}_p$

Os processos de codificação e decodificação de códigos LDPC definidos sobre corpo de inteiros módulo-p (p é primo) segue basicamente os mesmos procedimentos presentes em qualquer sistema de comunicação.

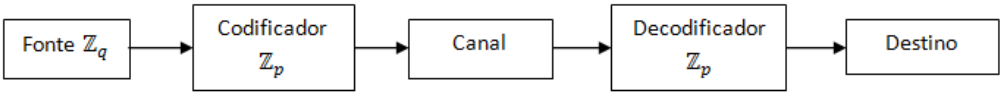

Figura 3.2: Sistema de Comunicação

A mensagem gerada por uma fonte não binária é uma sequência de símbolos pertencentes a $\mathbb{Z}_q$, que são agrupadas em blocos de comprimento fixo igual a k símbolos de informação pelo codificador. A saída do codificador é uma sequência de símbolos em $\mathbb{Z}_p$, denominada palavra código, de comprimento n, onde $n > k$. Estes símbolos são mapeados em sinais adequados para a transmissão pelo canal que adiciona ruído. A sequência de símbolos corrompida pelo ruído entra no decodificador que entrega ao destinatário uma estimativa da mensagem transmitida.

Portanto, para um código de bloco linear (n, k), os parâmetros n e k são respectivamente, o comprimento da palavra código e da mensagem, onde a diferença $(n - k)$ representa o número de símbolos de redundância introduzidos pelo codificador para detecção e correção de erro. A taxa de codificação é definida pela Equação 2.2.

3.5 Construção de Códigos LDPC sobre o Corpo $\mathbb{Z}_p$

Em [14] Gonçalves descreve um método simples para a construção de códigos LDPC definidos sobre o anel de inteiros $\mathbb{Z}_4$, onde são utilizadas substituições dos 1's da matriz de verificação de paridade **H** de códigos LDPC binários pelos símbolos 1 e 3 do anel. Neste trabalho utilizaremos o mesmo procedimento de obtenção da matrix **H** desenvolvidos em [14], onde os 1's são agora substituídos pelos símbolos não nulos de $\mathbb{Z}_5$. A decodificação dos códigos LDPC sobre $\mathbb{Z}_5$ é feita adaptando-se o Algoritmo Soft-Input Soft-Output (SISO) proposto por Moreira e Farrell [1].

Em [6] Mackay e Neal apresentam um conjunto de estratégias para gerar códigos LDPC binários que são enumeradas de 1 a 6 e neste trabalho utilizamos o item 3, ou seja, a matriz **H** é gerada com colunas de peso de Hamming w_c, linhas de peso de Hamming w_r, e não possuindo quaisquer duas colunas com mais de um 1 em comum.

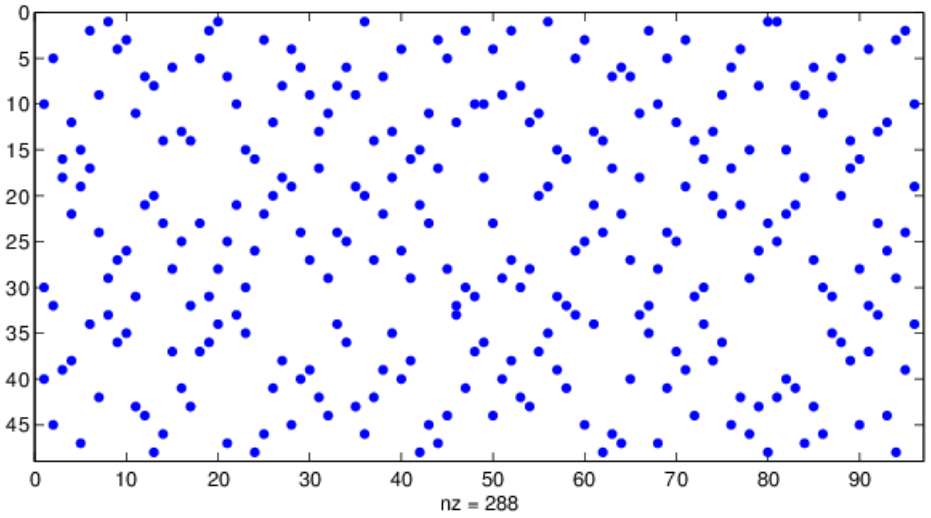

Figura 3.3: Matriz do código de Mackay H(48,96)

A Fig. 3.3 ilustra um exemplo de matriz **H** utilizando essa estratégia. Neste exemplo foi considerado o código de Mackay regular de peso de coluna w_c igual a 3 e peso de linha w_r igual a 6, obtida em [15]. Este código LDPC binário possui comprimento n=96 e taxa de codificação igual R_c a $\frac{1}{2}$. Os elementos nulos são representados pelos espaços em branco, e os pontos escuros são os elementos diferentes de zero.

Para a obtenção de códigos LDPC sobre $\mathbb{Z}_5$ as matrizes **H** binárias de verificação de paridade, disponibilizadas em [15] por Mackay, são modificadas através do seguinte procedimento:

- Identificação dos elementos 1's da matriz binária **H**.
- Substituição aleatória destes elementos diferentes de zero, por elementos não nulos do corpo de inteiros $\mathbb{Z}_5$, ou seja, $(1, 2, 3, 4)$.

Uma alternativa ao método proposto acima para a obtenção da matriz **H** sobre $\mathbb{Z}_p$ é a utilização da construção de **H** baseada em submatrizes definidas através de *latim squares* como proposta por Shu Lin em [16].

3.6 Decodificação dos Códigos LDPC

Os códigos LDPC binários possuem uma baixa densidade de elementos não nulos na sua matriz de verificação de paridade **H**, equivalentemente, o grafo de Tanner [10] também possui

uma baixa densidade de ligações. Assim, o pequeno número de ligações entre os nós do grafo bipartido permite um processo iterativo de decodificação simples, tendo em vista que a complexidade do algoritmo de decodificação LDPC está diretamente ligada a esta densidade. A densidade de um código é dada por: $d_s = \frac{w_c}{n}$.

A decodificação iterativa permite que o vetor recebido seja analisado diversas vezes até que se encontre um vetor considerado decodificado ou que seja declarado um erro. Essa decodificação iterativa ocorre com trocas sucessivas de informações entre os nós de símbolos e os nós de paridade, através das ligações feitas no grafo de Tanner.

3.6.1 Algoritmo de Decodificação Soft-Input Soft-Output (SISO)

O Algoritmo Soft-Input Soft-Output (SISO), trabalha com distâncias Euclidianas para a decodificação de códigos LDPC, reduzindo sensivelmente os efeitos de *overflow* e *underflow* dos processos iterativos. Ele foi proposto primeiramente por Moreira e Farrell [1] em abril de 2008. O desempenho alcançado por este algoritmo é similar ao do algoritmo proposto por Mackay e Neal [6]. Uma contribuição importante do algoritmo proposto por Farrell e Moreira [2] é que não é preciso ter conhecimento antecedente da informação do canal para executar a decodificação, isto é, não é preciso conhecer as probabilidades a priori dos símbolos transmitidos pelo canal, como acontece no algoritmo *Soma Produto* (SPA).

O algoritmo SISO apresenta apenas adições e comparações. A base dos algoritmos de decodificação dos códigos LDPC é descrita em [4], [5] e [6]. Entretanto, o algoritmo SISO faz uso da distância Euclidiana ao quadrado como a métrica para realizar a decodificação iterativa dos códigos LDPC. O algoritmo SISO pode ser facilmente adaptado para decodificar códigos LDPC definidos sobre $\mathbb{Z}_p$.

Inicialização

No passo inicial, são calculadas as distâncias Euclidianas quadradas entre cada símbolo do vetor recebido e os possíveis valores de cada símbolo na posição i, referente às posições da modulação adotada, no caso a modulação $5-PSK$. Dessa forma, i pode admitir as seguintes posições:

- $i\,(0) = cos\,(0) + jsen\,(0)$ posição do símbolo 0 na constelação $5 - PSK$.
- $i\,(1) = cos\left(\frac{2\pi}{5}\right) + jsen\left(\frac{2\pi}{5}\right)$ posição do símbolo 1 na constelação $5 - PSK$.
- $i\,(2) = cos\left(\frac{4\pi}{5}\right) + jsen\left(\frac{4\pi}{5}\right)$ posição do símbolo 2 na constelação $5 - PSK$.
- $i\,(3) = cos\left(\frac{6\pi}{5}\right) + jsen\left(\frac{6\pi}{5}\right)$ posição do símbolo 3 na constelação $5 - PSK$.
- $i\,(4) = cos\left(\frac{8\pi}{5}\right) + jsen\left(\frac{8\pi}{5}\right)$ posição do símbolo 4 na constelação $5 - PSK$.

Admitindo que $\mathbf{r} = [\mathbf{r}_n]$ é o vetor recebido pelo decodificador, calcula-se as 5 distâncias Euclidianas quadradas correspondente a cada símbolo do vetor $\mathbf{r}$ uma para cada posição que o símbolo pode assumir na constelação da modulação empregada, ou seja,

$$\begin{aligned}
\mathbf{d}_0^2 &= ((r_1 - i(0))^2, (r_2 - i(0))^2, \ldots, (r_n - i(0))^2) = \left(d_0^2(1), d_0^2(2), \ldots, d_0^2(n)\right), \\
\mathbf{d}_1^2 &= ((r_1 - i(0))^2, (r_2 - i(0))^2, \ldots, (r_n - i(0))^2) = \left(d_1^2(1), d_1^2(2), \ldots, d_1^2(n)\right), \\
\mathbf{d}_2^2 &= ((r_1 - i(0))^2, (r_2 - i(0))^2, \ldots, (r_n - i(0))^2) = \left(d_2^2(1), d_2^2(2), \ldots, d_2^2(n)\right), \\
\mathbf{d}_3^2 &= ((r_1 - i(0))^2, (r_2 - i(0))^2, \ldots, (r_n - i(0))^2) = \left(d_3^2(1), d_3^2(2), \ldots, d_3^2(n)\right), \\
\mathbf{d}_4^2 &= ((r_1 - i(0))^2, (r_2 - i(0))^2, \ldots, (r_n - i(0))^2) = \left(d_4^2(1), d_4^2(2), \ldots, d_4^2(n)\right). \qquad (3.3)
\end{aligned}$$

A inicialização deste algoritmo consiste em fixar os valores dos coeficientes das matrizes de inicialização $\mathbf{Q}_{ij}^0$, $\mathbf{Q}_{ij}^1$, $\mathbf{Q}_{ij}^2$, $\mathbf{Q}_{ij}^3$ e $\mathbf{Q}_{ij}^4$, com os valores dos componentes das equações 3.3 correspondentes. Todas as matrizes $\mathbf{Q}_{ij}^0$, $\mathbf{Q}_{ij}^1$, $\mathbf{Q}_{ij}^2$, $\mathbf{Q}_{ij}^3$ e $\mathbf{Q}_{ij}^4$ possuem as mesmas características da matriz de verificação de paridade $\mathbf{H}$, onde a posição dos elementos não nulos da matriz $\mathbf{H}$ é conservada nas matrizes produzidas na inicialização. Dessa forma, as matrizes de inicialização são dadas por:

$$\begin{aligned}
\mathbf{Q}_{ij}^0 &= d_0^2(j), \\
\mathbf{Q}_{ij}^1 &= d_1^2(j), \\
\mathbf{Q}_{ij}^2 &= d_2^2(j), \\
\mathbf{Q}_{ij}^3 &= d_3^2(j), \\
\mathbf{Q}_{ij}^4 &= d_4^2(j). \qquad (3.4)
\end{aligned}$$

3.6.2 Passo Horizontal

O passo horizontal leva em consideração que as equações de verificação de paridade correspondentes aos códigos LDPC tem de ser satisfeita.

Depois da inicialização, o processo de decodificação começa a troca de informações entre os nós de símbolo e os nós de verificação de paridade. A informação $\mathbf{Rd}_{ij}^x$ enviada por cada nó de verificação f_i para seus respectivos nós de símbolos c_j, a variável x correponde aos símbolos de $\mathbb{Z}_5$, e a informação $\mathbf{Rd}_{ij}^x$ é calculada através das seguintes equações:

$$\mathbf{Rd}_{ij}^{0} = -\log_2 \sum_{d:c_j=0} \left[2^{-\left(\sum\limits_{k \in N(i)\backslash j} Q_{ik}^{d_k} \right)} \right],$$
$$\mathbf{Rd}_{ij}^{1} = -\log_2 \sum_{d:c_j=1} \left[2^{-\left(\sum\limits_{k \in N(i)\backslash j} Q_{ik}^{d_k} \right)} \right],$$
$$\mathbf{Rd}_{ij}^{2} = -\log_2 \sum_{d:c_j=2} \left[2^{-\left(\sum\limits_{k \in N(i)\backslash j} Q_{ik}^{d_k} \right)} \right],$$
$$\mathbf{Rd}_{ij}^{3} = -\log_2 \sum_{d:c_j=3} \left[2^{-\left(\sum\limits_{k \in N(i)\backslash j} Q_{ik}^{d_k} \right)} \right],$$
$$\mathbf{Rd}_{ij}^{4} = -\log_2 \sum_{d:c_j=4} \left[2^{-\left(\sum\limits_{k \in N(i)\backslash j} Q_{ik}^{d_k} \right)} \right]. \tag{3.5}$$

Nas fórmulas acima, c_j é o j-ésimo nó de símbolo e $N(i)$ representa o conjunto de índices de todos os nós de símbolos ligados ao nó de paridade f_i, onde $N(i)\backslash j$ representa o mesmo conjunto com exceção do nó de símbolo c_j.

3.6.3 Passo Vertical

No passo vertical, o nó de símbolo c_j envia para o nó de paridade f_i a estimativa $\mathbf{Q}_{ij}^{x}$, a qual é a estimativa do nó ser o símbolo x ($x = (0, 1, 2, 3, 4)$), de acordo com a informação dada pelos outros nós de verificação ligados a ele. As informações que o nó de símbolo c_j enviam para o nó de paridade f_i são dados pelas seguintes equações:

$$\mathbf{Q}_{ij}^{0} = d_0^2(j) + \sum_{k \in M(j)\backslash i} \mathbf{Rd}_{kj}^{0},$$
$$\mathbf{Q}_{ij}^{1} = d_1^2(j) + \sum_{k \in M(j)\backslash i} \mathbf{Rd}_{kj}^{1},$$
$$\mathbf{Q}_{ij}^{2} = d_2^2(j) + \sum_{k \in M(j)\backslash i} \mathbf{Rd}_{kj}^{2},$$
$$\mathbf{Q}_{ij}^{3} = d_3^2(j) + \sum_{k \in M(j)\backslash i} \mathbf{Rd}_{kj}^{3},$$
$$\mathbf{Q}_{ij}^{4} = d_4^2(j) + \sum_{k \in M(j)\backslash i} \mathbf{Rd}_{kj}^{4}. \tag{3.6}$$

Onde $M(j)$ representa o conjunto de índices de todos os nós de paridade f_i ligados aos nós de símbolo c_j, onde $M(j)\backslash i$ representa o mesmo conjunto com exceção do nó de paridade f_i.

Dessa forma, o cálculo dos coeficientes das matrizes $\mathbf{Q}_{ij}^{x}$ permite o cálculo dos coeficientes das matrizes $\mathbf{Rd}_{ij}^{x}$ que em seguida pode ser utilizada para apresentar uma estimativa de cada valor do índice j.

Por fim, uma estimativa de cada símbolo x é obtida. Essa estimativa é calculada pelo argumento mínimo das distâncias suaves acumuladas, dadas por:

$$\hat{d}_j^2 = \underset{x}{argmin}[d_x^2(j) + \sum_{k \in M(j)} \mathbf{Rd}_{kj}^{x}] \tag{3.7}$$

Assim, o decodificador pode tomar uma decisão em um dado passo do processo de decodificação, fundamentando-se na ocorrência das seguintes situações:

- Se $\hat{d}_0^2$ for a menor distância suave acumulada obtida, então $x_j = 0$.
- Se $\hat{d}_1^2$ for a menor distância suave acumulada obtida, então $x_j = 1$.
- Se $\hat{d}_2^2$ for a menor distância suave acumulada obtida, então $x_j = 2$.
- Se $\hat{d}_3^2$ for a menor distância suave acumulada obtida, então $x_j = 3$.
- Se $\hat{d}_4^2$ for a menor distância suave acumulada obtida, então $x_j = 4$.

Como pode-se observar nas expressões acima, não é necessário o conhecimento das probabilidades a priori dos símbolos nem o valor da variância do ruído σ presente no canal, pois no cálculo decodificador SISO utiliza-se as distâncias euclidianas quadráticas, ao contrário do tradicional algoritmo Soma-Produto (SPA) que requer as informções de probabilidades a priori e variância do ruído do canal para detectar uma melhor performance (BER) na decodificação de um dado código LDPC.

3.6.4 Uma simplificação dos Cálculos do Passo Horizontal

À medida em que a decodificação progride no número de iterações, podem ocorrer problemas numéricos nos cálculos, porque as distâncias Euclidianas quadráticas num dado momento da simulação computacional são estimadas como a somatória de valores prévios. Os valores das distâncias torna-se números grandes, de tal forma que nas equações 3.5 os termos da forma $2^{(-d_x^2)}$ tendem a zero, ocorrendo *underflow* que faz o decodificador tomar decisões erradas. No entanto, problemas com as operações das Equações 3.5 podem ser solucionados usando a seguinte aproximação:

$$log_2(2^{\gamma} + 2^{\delta}) = max(\gamma, \delta) + log_2(1 + 2^{-|\gamma - \delta|}), \tag{3.8}$$

no caso binário, define-se $\gamma = -d_{x_1}^2$ e $\delta = -d_{x_2}^2$, em que γ e δ são os negativos das distâncias euclidianas ao quadrado, onde o termo de correção $log_2(1 + 2^{-|\gamma - \delta|})$ é excluído, fazendo com que a aproximação seja simplificada:

$$-log_2(2^{-d_{x_1}^2} + 2^{-d_{x_2}^2}) \approx min(d_{x_1}^2, d_{x_2}^2). \tag{3.9}$$

Essa aproximação simplificada é valida para códigos LDPC pequenos, mas para códigos LDPC extensos o fator de correção $log_2(1 + 2^{-|d^2_{x_1} - d^2_{x_2}|})$ é acrescentado, obtendo a seguinte expressão:

$$-log_2(2^{-d^2_{x_1}} + 2^{-d^2_{x_2}}) = min(d^2_{x_1}, d^2_{x_2}) - log_2(1 + 2^{-|d^2_{x_1} - d^2_{x_2}|}). \quad (3.10)$$

Dessa forma, podemos utilizar esta aproximação para implementar o algoritmo de decodificação, simplificando as implementações práticas, onde as operações envolvidas utilizam apenas adições e comparações.

A equação 3.10, pode ser adequada para códigos LDPC definidos sobre corpos de inteiros $\mathbb{Z}_p$, assim, temos:

$$-log_2(2^{-d^2_{x_1}} + 2^{-d^2_{x_2}} + \cdots + 2^{-d^2_{x_p}}) = min(d^2_{x_1}, d^2_{x_2}, \cdots, d^2_{x_p}) - log_2(1 + \sum_{i \neq j} 2^{-|d^2_{x_i} - d^2_{x_j}|}), \quad (3.11)$$

o valor de $d^2_{x_j}$ é alcançado através de $d^2_{x_j} = min(d^2_{x_1}, d^2_{x_2}, \cdots, d^2_{x_p})$.

3.7 Exemplo

Vamos construir o código $\mathbb{C}(8, 4)$ de taxa $R_c = \frac{1}{2}$ sobre o corpo de inteiros finito $\mathbb{Z}_5$. A matriz de verificação de paridade do código binário é dada por:

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

A matriz $\mathbf{H}$ sobre $\mathbb{Z}_5$ usando substituição dos símbolos fica:

$$\mathbf{H} = \begin{bmatrix} 1 & 2 & 0 & 4 & 0 & 0 \\ 0 & 4 & 1 & 0 & 0 & 3 \\ 2 & 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

Colocando na forma sistemática, temos:

$$\mathbf{H}_5 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 4 \\ 0 & 1 & 0 & 4 & 2 & 3 \\ 0 & 0 & 1 & 4 & 2 & 1 \end{bmatrix}$$

E sua matriz Geradora $\mathbf{G}$ sistemática é dada por:

$$\mathbf{G}_5 = \begin{bmatrix} 4 & 1 & 1 & 1 & 0 & 0 \\ 4 & 3 & 3 & 0 & 1 & 0 \\ 1 & 2 & 4 & 0 & 0 & 1 \end{bmatrix}$$

Seja a informação $\mathbf{m} = [3\ 2\ 0]$, que gera, a partir de $\mathbf{c} = \mathbf{m} \cdot \mathbf{G}$, a seguinte palavra código $\mathbf{c} = [0\ 4\ 4\ 3\ 2\ 0]$. Como a modulação utilizada é a $5-PSK$, a palavra código modulada transmitida será $\mathbf{c}_{5-PSK} = [1,\ 0.3090-0.9510i,\ 0.3090-0.9510i,\ -0,8090-0,5877i,\ -0.8090+0.5877i,\ 1]$, a palavra código modulada adicionada do ruído complexo produzido pelo canal AWGN com desvio padrão $\sigma = 0.6324$, implica na mensagem recebida: $\mathbf{r}$=[1.1296+0.0037i, 0.5414-0.9483i, 0.6181-0.8818i, -0.4832-0.3746i, -0.3241+0.9814i, 1.2097+0.2272i].

Inicialização

Na inicialização, as distâncias Euclidianas são calculadas. Dessa forma, a tabela 3.1 mostra os valores calculados para cada símbolo enviado:

w	**c**	$\mathbf{c}_{5-PSK}$	**r**	d_0^2	d_1^2	d_2^2	d_3^2	d_4^2
1	0	1	1.1296+0.0037i	0.0168	1,5708	4.0996	4.1084	1,5851
2	4	0.3090 - 0.9510i	0.5414-0.9483i	1.1096	3,6618	4.1835	1.9537	0.0540
3	4	0.3090 - 0.9510i	0.6181-0.8818i	0.9235	3.4551	4.1966	2.1232	0.1003
4	3	-0.8090 - 0.5877i	-0.4832-0.3746i	2.3403	2.3852	1.0325	0.1515	0.9598
5	2	-0.8090 + 0.5877i	-0.3241+0.9814i	2.7166	0.4018	0.3900	2.6975	4.1354
6	0	1	1.2097+0.2272i	0.0956	1.3352	4.2055	4.7398	2.1998

Tabela 3.1: Distâncias da mensagem enviada

Os valores das distâncias calculdas são usados para fixar os valores dos coeficientes das matrizes de inicialização $\mathbf{Q}_{ij}^0$, $\mathbf{Q}_{ij}^1$, $\mathbf{Q}_{ij}^2$, $\mathbf{Q}_{ij}^3$ e $\mathbf{Q}_{ij}^4$, onde essas matrizes possuem as mesmas características da matriz de verificação de paridade $\mathbf{H}_5$, onde os elementos nulos dos coeficientes não são utilizados na decodificação. Dessa forma, temos as seguintes Tabelas 3.2, 3.3, 3.4, 3.5 e 3.6:

Passo Horizontal

No passo horizontal iniciamos com a construção das equações de paridade da matriz $\mathbf{H}$, onde cada linha de $\mathbf{H}$ possui uma equação de verificação de paridade, dessa forma, para a matriz $\mathbf{H}$, dada neste exemplo, temos:

$$\begin{aligned} c_{11} + c_{12} + c_{14} &= 0 \\ c_{22} + c_{23} + c_{26} &= 0 \\ c_{31} + c_{33} + c_{35} &= 0 \end{aligned} \tag{3.12}$$

w	1	2	3	4	5	6
1	0.0168	1.1096	0	2.3403	0	0
2	0	1.1096	0.9235	0	0	0.0956
3	0.0168	0	0.9235	0	2.7166	0

Tabela 3.2: Valores de $\mathbf{Q}_{ij}^0$

w	1	2	3	4	5	6
1	1.5708	3.6618	0	2.3852	0	0
2	0	3.6618	3.4551	0	0	1.3352
3	1.5708	0	3.4551	0	0.4018	0

Tabela 3.3: Valores de $\mathbf{Q}_{ij}^1$

w	1	2	3	4	5	6
1	4.0996	4.1835	0	1.0325	0	0
2	0	4.1835	4.1966	0	0	4.2055
3	4.0996	0	4.1966	0	0.3900	0

Tabela 3.4: Valores de $\mathbf{Q}_{ij}^2$

w	1	2	3	4	5	6
1	4.1084	1.9537	0	0.1515	0	0
2	0	1.9537	2.1232	0	0	4.7398
3	4.1084	0	2.1232	0	2.6975	0

Tabela 3.5: Valores de $\mathbf{Q}_{ij}^3$

w	1	2	3	4	5	6
1	1.5851	0.0540	0	0.9598	0	0
2	0	0.0540	0.1003	0	0	2.1998
3	1.5851	0	0.1003	0	4.1354	0

Tabela 3.6: Valores de $\mathbf{Q}_{ij}^4$

onde o subíndice de cada elemento da palavra-código representa a posição do elemento diferente de zero da matriz $\mathbf{H}$.

A tabela 3.7 mostra as combinações lineares entre os símbolos do corpo de inteiros $\mathbb{Z}_5$, em que cada soma de cada elemento $\mathbb{Z}_5$ é igual ao seu correspondente símbolo. Essa tabela 3.7 foi construída aos pares, pois os cálculos das combinações lineares é igual ao número de elementos diferentes de zero da linha da matriz $\mathbf{H}$ menos 1, portanto, temos na matriz $\mathbf{H}$ 3 elementos diferentes de zero menos 1, obtemos 2.

No passo seguinte, calculamos as distâncias suaves, para que no passo vertical possamos determinar os valores das matrizes $\mathbf{Rd}_{ij}^x$, onde x são os elementos de $\mathbb{Z}_5$.

Assim, como exemplo, iremos calcular o valor de Rd_{12}^1 que está relacionado com a equação de verificação de paridade $c_{11} + c_{12} + c_{14} = 0$, que é a distância suave estimada, que o nó de

0	1	2	3	4
0+0	0+1	0+2	0+3	0+4
1+4	1+0	1+1	1+2	1+3
2+3	2+4	2+0	2+1	2+2
3+2	3+3	3+4	3+0	3+1
4+1	4+2	4+3	4+4	4+0

Tabela 3.7: Soma dois-a-dois dos símbolos do corpo de inteiros $\mathbb{Z}_5$

paridade 1 envia para o nó de símbolo 2, na qual o nó de símbolo está no estado 1. Como o símbolo enviado é 1, para este caso, existem 5 tipos de combinações lineares que utilizam os coeficientes da equação de paridade $c_{11} + c_{12} + c_{14} = 0$ excluindo o coeficiente c_{12} a qual é a posição da informação calculada, onde a soma dos índices das distâncias suaves são iguais a 1, dessa forma, calculamos essas distâncias:

$$\begin{aligned} d^2_{01} &= Q^0_{11} + Q^1_{14} = 0.0168 + 2.3852 = 2.4020 \\ d^2_{10} &= Q^1_{11} + Q^0_{14} = 1.5708 + 2.3403 = 3.9111 \\ d^2_{24} &= Q^2_{11} + Q^4_{14} = 4.0996 + 0.9598 = 5.0594 \\ d^2_{33} &= Q^3_{11} + Q^3_{14} = 4.1084 + 0.1515 = 4.2599 \\ d^2_{42} &= Q^4_{11} + Q^2_{14} = 1.5851 + 1.0325 = 2.6176 \end{aligned} \tag{3.13}$$

os subíndices ij das matrizes de inicialização $\mathbf{Q}^x_{ij}$ correspondem aos subíndices dos coeficientes das equações de paridade, excluindo-se Q^1_{12}.

Para calcularmos a informação transmitida por Rd^1_{12}, precisamos substituir os valores das distâncias suaves na equação 3.5, temos:

$$\begin{aligned} Rd^1_{12} &= -log_2[2^{-d^2_{01}} + 2^{-d^2_{10}} + 2^{-d^2_{24}} + 2^{-d^2_{33}} + 2^{-d^2_{42}}] \\ &= -log_2[2^{-2.4020} + 2^{-3.9111} + 2^{-5.0594} + 2^{-4.2599} + 2^{-2.6176}] \\ &= -log_2(0.50079) \\ &= 0.9977. \end{aligned} \tag{3.14}$$

As Tabelas a seguir mostram os valores dos coeficientes das matrizes $\mathbf{Rd}^x_{ij}$.

w	1	2	3	4	5	6
1	1.0225	0.9538	0	0.2742	0	0
2	0	0.1560	0.2366	0	0	1.2089
3	0.0364	0	1.0738	0	0.1810	0

Tabela 3.8: Valores de $\mathbf{Rd}^0_{ij}$.

w	1	2	3	4	5	6
1	0.2585	0.9977	0	1.6277	0	0
2	0	1.4527	1.5930	0	0	2.0929
3	0.9870	0	-0.6073	0	1.0284	0

Tabela 3.9: Valores de $\mathbf{Rd}_{ij}^{1}$.

w	1	2	3	4	5	6
1	-0.3248	0.1696	0	2.0005	0	0
2	0	2.3173	2.3206	0	0	0.9339
3	0.65171	0	-0.1011	0	2.0236	0

Tabela 3.10: Valores de $\mathbf{Rd}_{ij}^{2}$.

w	1	2	3	4	5	6
1	-0.0180	-0.3201	0	0.6752	0	0
2	0	1.1073	1.0153	0	0	-0.2021
3	1.8478	0	1.0552	0	0.7562	0

Tabela 3.11: Valores de $\mathbf{Rd}_{ij}^{3}$.

w	1	2	3	4	5	6
1	0.7960	0.1255	0	-0.2645	0	0
2	0	-0.1178	-0.1513	0	0	0.0220
3	1.3173	0	2.0160	0	-0.2430	0

Tabela 3.12: Valores de $\mathbf{Rd}_{ij}^{4}$.

Passo Vertical

Os valores de $\mathbf{Rd}_{ij}^{x}$ são utilizados para calcular os valores dos coeficientes das matrizes $\mathbf{Q}_{ij}^{x}$, que são atualizados utilizando as equações 3.6, assim, atualizando o valor de Q_{12}^{1} que é determinado sem nenhuma normalização, temos:

$$\begin{aligned} Q_{12}^{1} &= d_{1}^{2}(2) + R_{22}^{1} + R_{32}^{1} \\ &= 3.6618 + 1.4527 + 0 \\ &= 5.1145. \end{aligned} \tag{3.15}$$

As tabelas abaixo mostram os valores atualizados para os coeficientes das matrizes $\mathbf{Q}_{ij}^{x}$, onde x são os símbolos de $\mathbb{Z}_5$.

w c	1	2	3	4	5	6
1	0.0532	1.2656	0	2.3403	0	0
2	0	2.0634	1.9973	0	0	0.0956
3	1.0393	0	1.1601	0	2.7166	0

Tabela 3.13: Valores de $\mathbf{Q}_{ij}^{0}$ atualizados.

w c	1	2	3	4	5	6
1	2.5578	5.1145	0	2.3852	0	0
2	0	4.6595	2.8478	0	0	1.3352
3	1.8293	0	5.0481	0	0.4018	0

Tabela 3.14: Valores de $\mathbf{Q}_{ij}^{1}$ atualizados.

w c	1	2	3	4	5	6
1	4.7313	6.5008	0	1.0325	0	0
2	0	4.3531	4.0955	0	0	4.2055
3	1.5851	0	0.1003	0	4.1354	0

Tabela 3.15: Valores de $\mathbf{Q}_{ij}^{2}$ atualizados.

w c	1	2	3	4	5	6
1	5.9562	3.0610	0	4.1084	0	0
2	0	1.6336	3.1784	0	0	4.7398
3	4.0904	0	3.1385	0	2.6975	0

Tabela 3.16: Valores de $\mathbf{Q}_{ij}^{3}$ atualizados.

w c	1	2	3	4	5	6
1	2.9024	-0.0638	0	0.9598	0	0
2	0	0.1795	2.1163	0	0	2.1998
3	2.3811	0	-0.0510	0	4.1354	0

Tabela 3.17: Valores de $\mathbf{Q}_{ij}^{4}$ atualizados.

Neste momento, podemos fazer uma estimativa da palavra-código enviada, através da equação 3.7 das distâncias suaves acumuladas para cada símbolo, como estamos considerando que o sím-

bolo enviado foi 1, temos as seguintes estimativas das distâncias suaves.

$$\begin{aligned}
\hat{d}_0^2(1) &= d_0^2(1) + Rd_{11}^0 + Rd_{21}^0 + Rd_{31}^0 = 0.0168 + 1.0225 + 0 + 0.0364 = 1.0757 \\
\hat{d}_1^2(1) &= d_1^2(1) + Rd_{11}^1 + Rd_{21}^1 + Rd_{31}^1 = 1.5708 + 0.2585 + 0 + 0.9870 = 2.8163 \\
\hat{d}_2^2(1) &= d_2^2(1) + Rd_{11}^2 + Rd_{21}^2 + Rd_{31}^2 = 4.0996 - 0.3248 + 0 + 0.6517 = 4.4265 \\
\hat{d}_3^2(1) &= d_3^2(1) + Rd_{11}^3 + Rd_{21}^3 + Rd_{31}^3 = 4.1084 - 0.0180 + 0 + 1.8478 = 5.9382 \\
\hat{d}_4^2(1) &= d_4^2(1) + Rd_{11}^4 + Rd_{21}^4 + Rd_{31}^4 = 1.5851 + 0.7960 + 0 + 1.3173 = 3.6984
\end{aligned} \tag{3.16}$$

A tabela 3.18, mostra os valores das distâncias suaves acumuladas para cada símbolo na primeira iteração.

w	**c**	d_0^2	d_1^2	d_2^2	d_3^2	d_4^2	$\hat{c}$
1	0	1.0757	2.8163	4.4265	5.9382	3.6984	0
2	4	2.2194	6.1122	6.6704	2.7409	0.0617	4
3	4	2.2339	4.4408	6.4161	4.1937	1.9650	4
4	3	2.6145	4.0129	3.0330	0.8267	0.6953	3
5	2	2.8976	1.4302	2.4136	3.4537	3.8924	1
6	0	1.3045	3.4281	5.1394	4.5377	2.2218	0

Tabela 3.18: Distâncias suaves acumuladas de cada símbolo

Dessa forma, a palavra-código estimada $\hat{\mathbf{c}} = [0\ 4\ 4\ 3\ 1\ 0]$, que para este caso não é a palavra-código, pois o elemento da posição 5 não é compatível com a palavra código enviada $\mathbf{c} = [0\ 4\ 4\ 3\ 2\ 0]$, observe que na posição 4 a menor distância d_4^2 pertence ao símbolo 4, porém o decodificador descarta essa distância, pois na informação não tem o símbolo 4, assim ele escolhe a segunda melhor distância d_3^2 que correnponde ao símbolo enviado 3.

Capítulo 4

Resultados

Neste capítulo é apresentada uma análise comparativa do desempenho de códigos LDPC definidos sobre $\mathbb{Z}_5$, em relação aos códigos binários e quartenários equivalentes.

As simulações dos códigos LDPC foram realizadas utilizando o software MatLab®, através do método Monte Carlo para a obtenção dos gráficos de desempenho da taxa de erro de bit. Também foram utilizadas as matrizes geradoras dos códigos para diferentes comprimentos, como proposto por Mackay [15], porém, adaptadas à construção de códigos sobre o corpo de inteiros $\mathbb{Z}_5$. As palavras-código transmitidas pelo canal foram decodificadas pelo algoritmo SISO, proposto por Farrell e Moreira [11], sendo considerado o comprimento da sequência de informação, o número de iterações e o tipo de canal de propagação.

4.1 Modelo do Sistema de Comunicação

Os códigos LDPC sobre $\mathbb{Z}_5$ foram obtidos modificando-se as matrizes binárias de Mackay conforme o método descrito na seção 3.5.

Os códigos binários utilizados como base para a construção dos códigos LDPC sobre $\mathbb{Z}_5$ são reproduzidos abaixo e estão disponíveis em [14]:

- LDPC (Número de referência: 96.3.963). Comprimento de bloco $n = 96$.
- LDPC (Número de referência: 204.33.486). Comprimento de bloco $n = 204$.
- LDPC (Número de referência: 408.3.834). Comprimento de bloco $n = 408$.
- LDPC (Número de referência: 252.252.3.252). Comprimento de bloco $n = 504$.
- LDPC (Número de referência: 816.3.174). Comprimento de bloco $n = 816$.
- LDPC (Número de referência: 504.504.3.504). Comprimento de bloco $n = 1008$.

O método utilizado para a construção das matrizes $\mathbf{H}_5$ e $\mathbf{G}_5$ sistemáticas sobre o corpo de inteiros $\mathbb{Z}_5$, são obtidas através do algoritmo descrito pela figura 4.1.

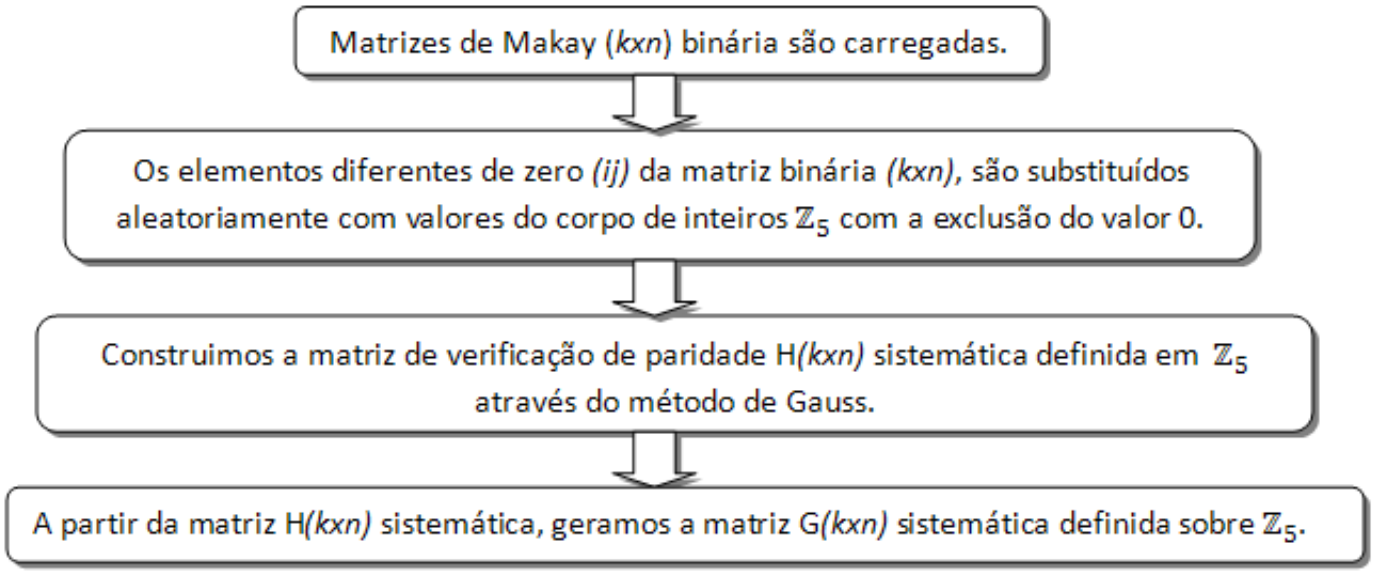

Figura 4.1: Algoritmo de construção das matrizes LDPC sobre o corpo de inteiro $\mathbb{Z}_5$.

Após a geração das matrizes $\mathbf{H}_5$ e $\mathbf{G}_5$, inicializamos o processo de codificação dos códigos LDPC definidos sobre corpo de inteiros módulo-5. A mensagem $\mathbf{m}$ é gerada aleatoriamente por uma fonte de símbolos pertencentes a $\mathbb{Z}_4$, que são agrupadas em blocos de comprimento fixo igual a k símbolos pelo codificador. A mensagem é codificada através da matriz $\mathbf{G}_5$, $\mathbf{c} = \mathbf{m} \cdot \mathbf{G}_5$, produzindo uma sequência de símbolos pertencentes a $\mathbb{Z}_5$, denominada palavra-código, de comprimento n. Como a modulação considerada neste trabalho é a $5 - PSK$ esses símbolos são mapeados em sinais adequados para a transmissão como descrito na seção 2.5, e repetida aqui,

$$\left\{ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \right\} \approx \left\{ \begin{array}{c} \cos(0) + j\sin(0) \\ \cos(\frac{2\pi}{5}) + j\sin(\frac{2\pi}{5}) \\ \cos(\frac{4\pi}{5}) + j\sin(\frac{4\pi}{5}) \\ \cos(\frac{6\pi}{5}) + j\sin(\frac{6\pi}{5}) \\ \cos(\frac{8\pi}{5}) + j\sin(\frac{8\pi}{5}) \end{array} \right\} \tag{4.1}$$

O sinal entra no canal que adiciona ruído gaussiano branco aos símbolos. Quando a palavra código corrompida entra no decodificador SISO, as distâncias Euclidianas ao quadrado são calculadas e a partir delas são calculadas as matrizes de inicialização $\mathbf{Q}_{ij}^{x}$, utilizadas no passo horizontal.

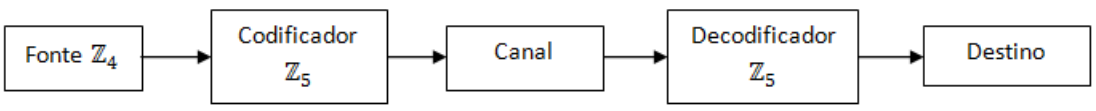

Figura 4.2: Modelo do sistema de comunicação LDPC

Após o processo de decodificação iterativa ser completado, uma estimativa $\hat{\mathbf{m}}$ da mensagem é entregue ao destinatário. Para a obtenção da taxa de erro de bit, esta estimativa é comparada com a mensagem original e os possíveis erros são contados.

4.2 Medidas de desempenhos

Existem diversos tipos de medidas para ilustrar o desempenho de um sistema de transmissão digital, a taxa de mensagens erradas (MER), a taxa de erro de bit (BER), a taxa de messagens erradas não detectadas e a taxa de erros não detectados, dentre eles se destaca a BER (Bit Error Rate) na saída do decodificador de canal.

Para comparar os códigos LDPC de forma equitativa usamos a sequinte definição da taxa de codificação:

$$R_c = \frac{\log q^k}{\log p^n} = \frac{k \log q}{n \log p}, \tag{4.2}$$

onde k é o número de símbolos de informação, n é o comprimento da palavra código e q e p são as cardinalidades dos alfabetos de entrada e saída do codificador respectivamente.

A energia por símbolo é expressa pela seguinte equação: $E_s = E_b \cdot \log_2 p \cdot R_c$, onde E_b é a energia de bit, R_c é a taxa de codificação e $\log_2 p$ fornece quantos bits é enviado por símbolo, onde p é a cardinalidade do alfabeto usado na modulação $p-PSK$. Para a modulação $5-PSK$, o símbolo 4 é utilizado apenas como redundância, assim, a relação entre bits por símbolos fica $\frac{8}{5}$, dessa forma, a energia por símbolo para essa modulação fica:

$$E_s = \frac{8}{5} \cdot E_b \cdot R_c. \tag{4.3}$$

Deste modo, considerando o caso de um canal Gaussiano, os gráficos de desempenho são expressos em termos de $\frac{Eb}{N_0}$, que está diretamente relacionado com a SNR (Relação Sinal Ruído) do canal,

$$SNR = \frac{E_s}{N_0} = \frac{\frac{8}{5} \cdot E_b \cdot R_c}{N_0}, \tag{4.4}$$

onde $\frac{N_0}{2}$ é a densidade espectral de potência bilateral do ruído, a energia de símbolo E_s é normalizada para 1. Assim, a variância do ruído Gaussiano é

$$\sigma^2 = \frac{5}{16 \cdot R_c \cdot \frac{E_b}{N_0}}. \tag{4.5}$$

que é adicionado a cada símbolo da palavra-código enviada.

Portanto, o desvio padrão necessário para a simulação do canal Gaussiano é dado por:

$$\sigma = \sqrt{\frac{5}{16 \cdot R_c \cdot \frac{E_b}{N_0}}}. \tag{4.6}$$

Para as modulações BPSK e $4-PSK$ a energia por símbolo é dada por: $E_{sBPSK} = E_b \cdot R_c$ e $E_{s4-PSK} = 2 \cdot E_b \cdot R_c$ respectivamente, assim, os desvios padrões respectivos são,

$$\sigma_{BPSK} = \sqrt{\frac{1}{2 \cdot R_c \cdot \frac{E_b}{N_0}}} \tag{4.7}$$

e

$$\sigma_{4-PSK} = \sqrt{\frac{1}{4 \cdot R_c \cdot \frac{E_b}{N_0}}}. \tag{4.8}$$

4.3 Condições de Simulações

Todas as simulações foram desenvolvidas no Software MatLab®. Os códigos binários utilizados como base para a construção dos códigos sobre $\mathbb{Z}_5$, foram descritos na seção 4.1.

Para fazer as comparações entre os códigos 5-ários e binários, utilizamos o mesmo processo na geração das matrizes geradora **G** sistemática desenvolvida na seção 4.1, só que neste caso utilizamos apenas as matrizes binárias, sem substituição dos elementos diferentes de zero, usamos o seguinte processo na geração da matriz **G**:

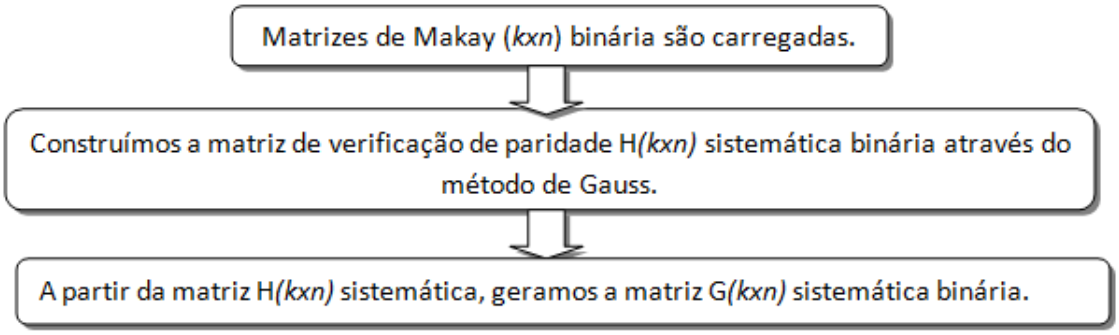

Figura 4.3: Algoritmo de construção das matrizes LDPC binária

Na decodificação iterativa utilizamos o algoritmo SISO para o código binário.

Para o código quartenário utilizamos somente a matriz teste de paridade **H** *(kxn)* não sistemática, usando o seguinte processo com todas as etapas do algoritmo de construção das matrizes sobre o anel $\mathbb{Z}_4$ [14] na Fig. 4.4. A modulação utilizada foi a $4-PSK$.

Para gerar o ruído Gaussiano utilizamos a função do MatLab® *normrnd* com média zero e desvio padrão σ e o desvanecimento Rayleigh foi gerado a partir da raíz quadrada da soma dos quadrados de duas variáveis aleatórias com distribuição Gaussiana com média nula e variâncias iguais.

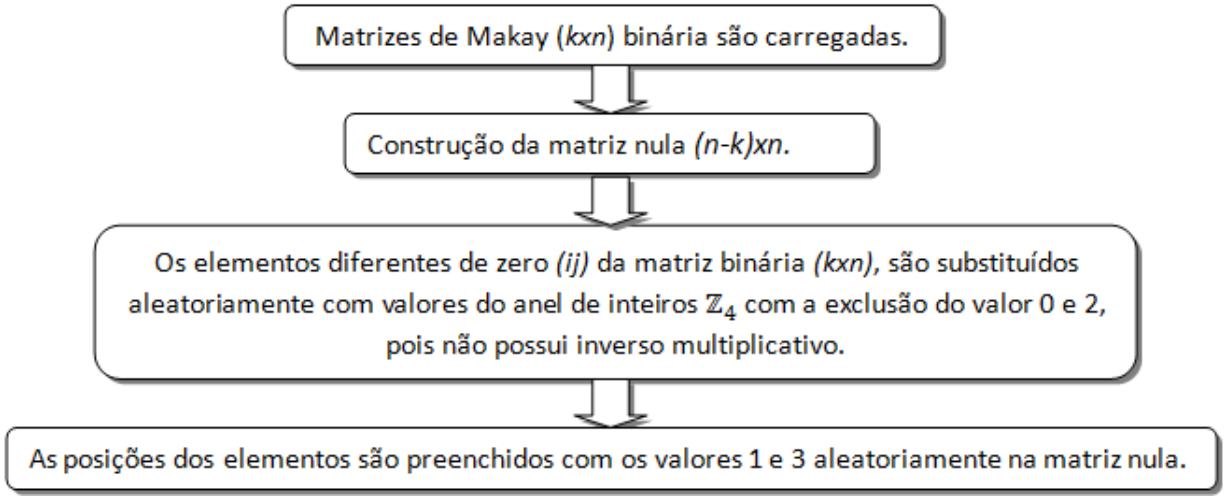

Figura 4.4: Algoritmo de construção das matrizes LDPC quartenária.

As curvas de desempenho dos códigos LDPC de todos os esquemas de codificação foram utilizando o algoritmo Monte Carlo, onde o número de palavras testadas para cada um dos códigos referidos, foi escolhido de tal forma, que após a decodificação garanta que o número de palavras-código erradas sejam suficientemente grande para ter um valor de BER confiável. O número de iterações do decodificador iterativo foi fixado em 3 nas comparações de desempenho.

4.4 Resultados das Simulações

Os códigos LDPC sobre $\mathbb{Z}_5$ analisados nas simulações foram descritos na seção 4.2. O desempenho destes códigos foram analisados para dois cenários: em canais com ruído gaussiano aditivo branco (AWGN) e em canais com desvanecimento Rayleigh.

A figura 4.5 apresenta o desempenho dos códigos LDPC *(n, k)* definidos sobre $\mathbb{Z}_5$ com $n = 2k$ e taxa de codificação igual a 0.43, para os comprimentos $n = 96$, 204, 408, 504, 816 e 1008 símbolos. O número de iterações para a decodificação iterativa foi fixada em 3 e o canal foi modelado com ruído AWGN. Observe que há uma melhora do desempenho dos códigos com o aumento de n. Assim, para uma BER de 10^{-3}, o código de comprimento $n = 1008$ símbolos apresenta desempenho em torno de 1.5dB melhor do que o código de comprimento $n = 96$.

Na figura 4.6 são apresentados o desempenho dos códigos LDPC definidos sobre $\mathbb{Z}_5$ e $\mathbb{Z}_4$ com comprimento $n = 96$ símbolos e o código binário de comprimento $n = 204$ bits, em um canal AWGN. Comparando o desempenho dos 3 códigos para BER igual a 10^{-3}, tem-se que o código definido sobre $\mathbb{Z}_5$ mostra um desempenho, em termos de $\frac{Eb}{N_0}$, em torno de 2.7dB pior do que os códigos binário e 6.0dB pior do que o quartenário. Este desempenho do código LDPC sobre $\mathbb{Z}_5$ já era esperado, devido às distâncias euclidianas entre os pontos da modulação $5 - PSK$ serem menores que as utilizadas nas outras duas codificações e devido ao pequeno comprimento dos códigos.

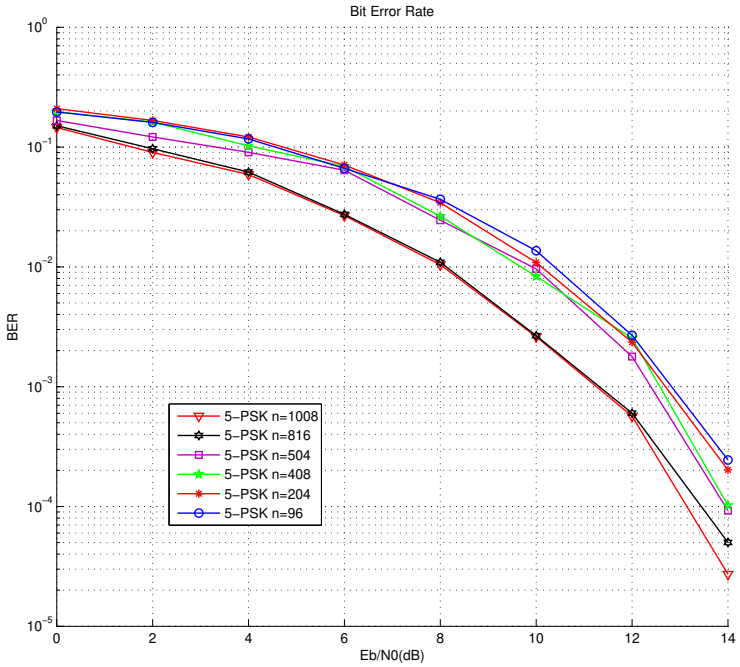

Figura 4.5: Desempenho dos códigos LDPC 5-ário para vários comprimentos.

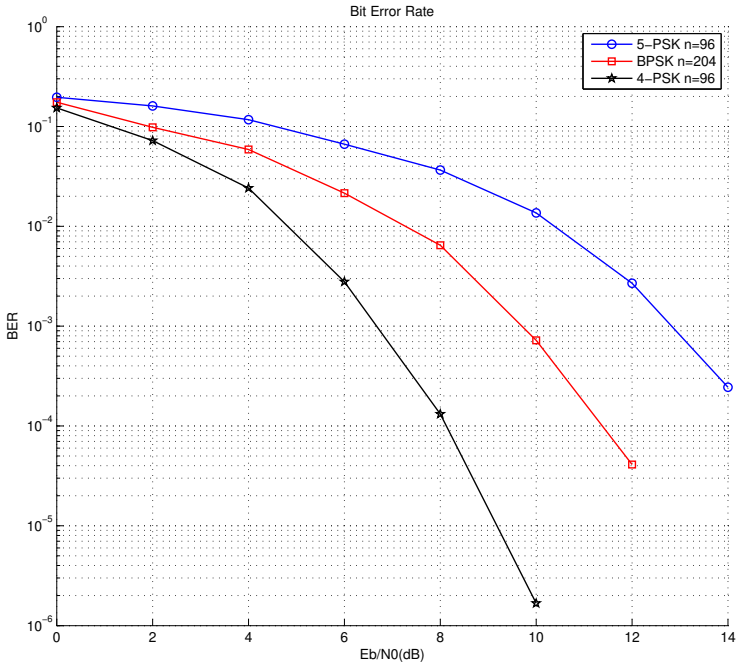

Figura 4.6: Desempenho dos códigos LDPC para as modulações BPSK, 4-PSK e 5-PSK, com n = 96 símbolos para os códigos não binários.

Na figura 4.7 são apresentados o desempenho dos códigos LDPC definidos sobre $\mathbb{Z}_5$ e $\mathbb{Z}_4$ com comprimento $n = 504$ símbolos e o código binário de comprimento $n = 1008$ bits, em um canal AWGN. Observe que o desempenho do código definido sobre $\mathbb{Z}_5$ para BER igual a 10^{-4}, está em torno de 2.7dB pior do que o código binário e 6.3dB pior do que o código quartenário.

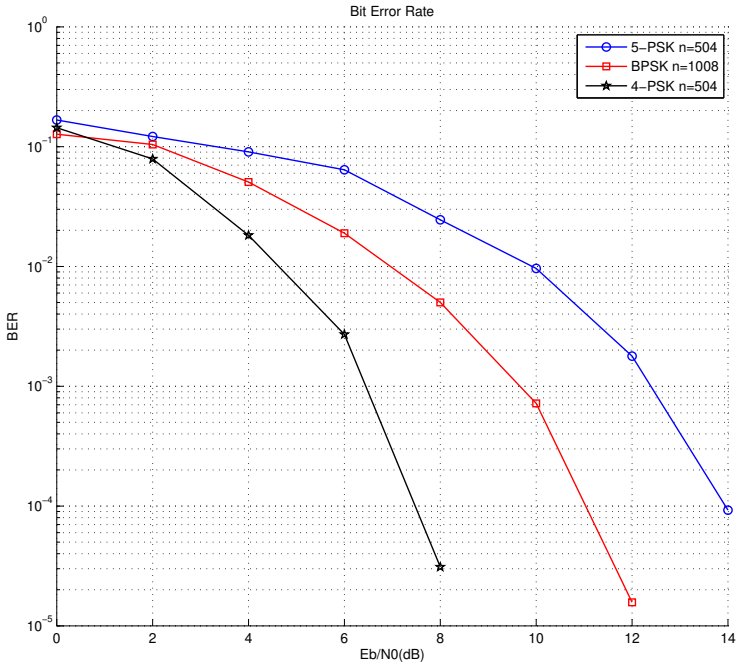

Figura 4.7: Desempenho dos códigos LDPC para as modulações BPSK, 4-PSK e 5-PSK, com n = 504 símbolos para os códigos não binários.

Na figura 4.8 temos o desempenho do código definido sobre $\mathbb{Z}_5$ de comprimento $n = 408$ símbolos para o canal com apenas ruído AWGN e outro com desvanecimento Rayleigh mais ruído AWGN. Para uma BER de 10^{-3} o desempenho do código com desvanecimento Rayleigh sofre uma degradação de 11 dB em relação à curva com apenas ruído AWGN.

Note que é possível melhorar o desempenho dos códigos LDPC definidos sobre $\mathbb{Z}_5$ realizando uma pequena alteração no processo de modulação. O símbolo 4 do alfabeto $\mathbb{Z}_5$ não é utilizado na parte de informação que sai do codificador da figura 4.9, este símbolo só aparece na parte de redundância da palavra código formada. Assim, mapeando os símbolos de informação na modulação $4 - PSK$ e os de redundância na modulação $5 - PSK$, podemos obter um melhor desempenho dos códigos definidos sobre $\mathbb{Z}_5$.

Assim, a figura 4.10 mostra o desempenho do código LDPC binário com $n = 204$ bits e modulação BPSK, do código LDPC definido sobre o anel $\mathbb{Z}_4$ com modulação $4 - PSK$, do código LDPC definido sobre $\mathbb{Z}_5$ com modulação $5 - PSK$ e com modulação $4 - 5 - PSK$.

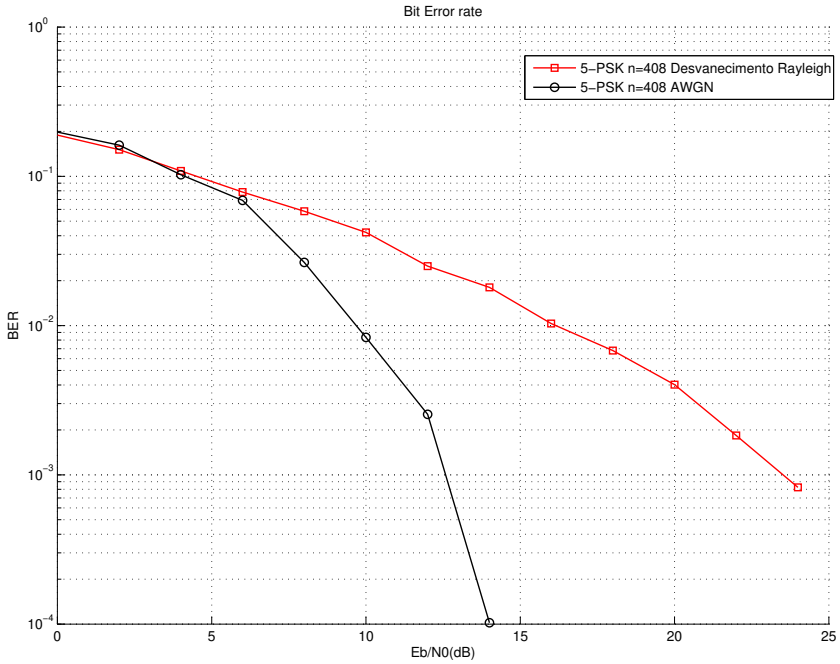

Figura 4.8: Desempenho do código LDPC definido sobre $\mathbb{Z}_5$ com comprimento $n = 408$, em um canal com ruído AWGN e com desvanecimento Rayleigh mais ruído AWGN.

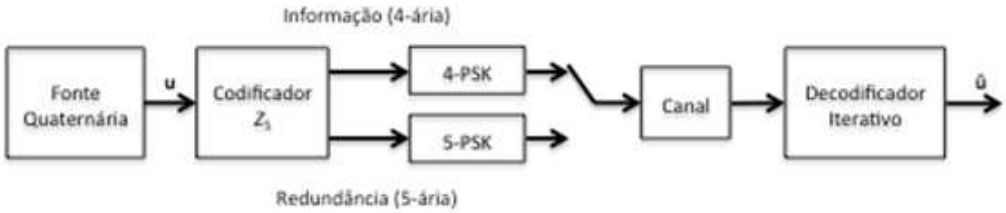

Figura 4.9: Esquema de codificação com modulação $4-5-PSK$.

A figura 4.11 mostra que, para uma BER de 10^{-3}, o código LDPC com modulação $4-5-PSK$ possui um desempenho, em termos de $\frac{Eb}{N_0}$, 1.5 dB melhor do que este código com modulação $5-PSK$, para $n = 408$ símbolos.

Todos os códigos não binários possuem comprimento $n = 96$ símbolos. O canal é AWGN. Note que no esquema de modulação $4-5-PSK$, o código definido sobre $\mathbb{Z}_5$ apresenta desempenho significativamente melhor que para a modulação $5-PSK$. Para BER igual a 10^{-3}, o código com modulação $4-5-PSK$ tem um desempenho de 1.8dB melhor do que este mesmo código com modulação $5-PSK$, ficando apenas a 4.3dB do código com modulação $4-PSK$ e 1.1dB do código binário.

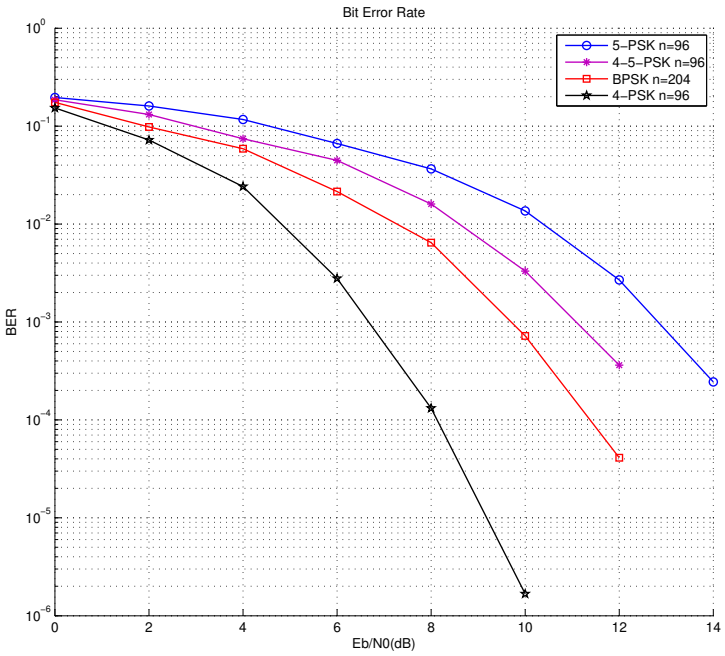

Figura 4.10: Desempenho dos códigos LDPC para as modulações BPSK, 4-PSK, 5-PSK e 4-5-PSK, em canal AWGN.

Os resultados apresentados nesta seção mostram que os códigos LDPC definidos sobre corpos de inteiros $\mathbb{Z}_5$ aparentemente possuem desempenho inferior aos códigos LDPC binários e quaternários ($\mathbb{Z}_4$) equivalentes. Note entretanto que o processo de obtenção dos códigos LDPC sobre corpos não foi otimizado, nem estruturado para extrair o máximo de desempenho dentro desta estrutura algébrica. Os códigos LDPC sobre $\mathbb{Z}_5$ apresentados neste trabalho estão, para uma mesma taxa de erro de bit, com desempenho, em termos de $\frac{Eb}{N0}$, piores que o esperado. Isto se deve aos comprimentos reduzidos dos códigos e ao baixo número de iterações realizadas no processo de decodificação. Entretanto, nosso intuito foi fazer uma análise comparativa de desempenho entre diversos tipos de códigos ($\mathbb{Z}_2$, $\mathbb{Z}_4$, $\mathbb{Z}_5$), sem nos preocuparmos com a otimização dos desempenhos destes. A escolha de um número de iterações reduzido na decodificação foi para que os tempos das simulações não se tornassem inviáveis na prática.

Portanto, há ainda bastante variantes a serem exploradas de modo a melhorar o desempenho destes códigos. Seria necessário um número de iterações por volta de 50 e comprimentos de códigos significativamente maiores. A estrutura algébrica dos corpos finitos $\mathbb{Z}_p$ também pode ser melhor explorada de modo a fornecer um processo de decodificação iterativo mais eficiente e rápido.

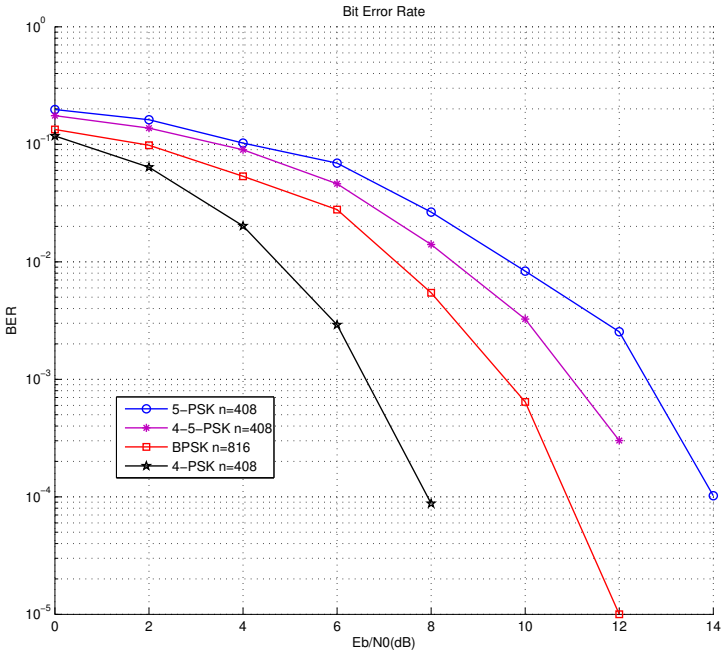

Figura 4.11: Desempenho dos códigos LDPC para as modulações BPSK, 4-PSK, 5-PSK e 4-5-PSK.

Capítulo 5

Conclusões

Poucos estudos sobre códigos LDPC definidos sobre os corpos finitos de inteiros $\mathbb{Z}_p$ são encontrados na literatura especializada. Assim, neste trabalho buscou-se a obtenção e a análise de desempenho de códigos LDPC definidos sobre o corpo de inteiros $\mathbb{Z}_5$. A análise de desempenho foi feita de modo comparativo em relação a códigos LDPC binários e códigos definidos sobre anéis de inteiros $\mathbb{Z}_4$ equivalentes.

Primeiramente construímos os códigos LDPC sobre $\mathbb{Z}_5$ modificando as matrizes $\mathbf{H}$ de verificação de paridade binárias obtidas por Mackay [15] seguindo o algoritmo descrito na Fig. 4.1. Na decodificação iterativa desses códigos foi utilizado o algoritmo SISO proposto em Castañeda e Farrell [1], que utiliza distâncias Euclidianas em vez de probabilidades. Os canais de transmissão utilizados foi o AWGN e o com desvanecimento Rayleigh.

A análise de desempenho foi realizada para códigos LDPC sobre $\mathbb{Z}_5$ com vários comprimentos em um canal perturbado por ruído AWGN. Como era de se esperar constatou-se que estes códigos possuem um melhor desempenho à medida que se aumenta o seu comprimento. Em seguida, o desempenho destes códigos foram comparados com os códigos LDPC quartenários e binários equivalentes, para os comprimentos $n = 96$ e 504 símbolos. Verificou-se que para ambos os comprimentos, os códigos LDPC sobre $\mathbb{Z}_5$ possuem desempenho, em termo de $\frac{Eb}{N_0}$, 2.5dB pior que os códigos equivalentes binários de referência.

Os códigos LDPC sobre $\mathbb{Z}_5$ também foram analisados em um canal com desvanecimento Rayleigh somado com o ruído aditivo gaussiano branco. Verificou-se que o desempenho foi 11dB inferior que para o caso de ter o canal com apenas ruído gaussiano, para a BER em 10^{-3}.

Finalmente, alteramos o processo de modulação dos símbolos da parte de informação da palavra código para o $4 - PSK$ de modo a aumentar a distância euclidiana entre esses símbolos, mantendo a modulação $5 - PSK$ para a parte de paridade da palavra código. Os códigos LDPC sobre $\mathbb{Z}_5$ com a modulação $4 - 5 - PSK$ para os comprimentos $n = 96$ e 408 símbolos, apresentaram um desempenho melhor de 1.8dB e 1.5dB, respectivamente, para uma de 10^{-3}, em relação à codificação com modulação $5 - PSK$.

Este trabalho mostrou que códigos LDPC definidos sobre corpos de inteiros $\mathbb{Z}_p$ podem ser uma alternativa aos códigos binários e quaternários equivalentes. Para tanto é necessária a criação de métodos mais eficientes de obtenção desses códigos sobre $\mathbb{Z}_5$. A substituição aleatória dos símbolos 1 da matriz **H** binária por símbolos não nulos do corpo $\mathbb{Z}_p$ não produziu códigos LDPC sobre $\mathbb{Z}_p$ com bom desempenho. Portanto, ainda existem vários parâmetros não explorados que devem produzir códigos LDPC melhores que os obtidos neste estudo.

5.1 Trabalhos Futuros

Os resultados apresentados nesta dissertação deslumbram vários caminhos que podem ser explorados para a obtenção de códigos LDPC sobre $\mathbb{Z}_p$ mais eficientes. Entre eles podemos citar:

- Obtenção de um método mais eficiente de construção de códigos LDPC definidos sobre $\mathbb{Z}_p$.
- Construção e análise de códigos LDPC definidos sobre outros corpos de inteiros, tais como: $\mathbb{Z}_7$, $\mathbb{Z}_{11}$ e $\mathbb{Z}_{13}$.
- Utilização de matrizes irregulares e matrizes regulares com outras taxas de codificação.
- Utilização de outro decodificador iterativo.
- Aplicação de códigos LDPC definidos sobre corpos finitos de inteiros para outros tipos de modulações (QAM).

Bibliografia

[1] P. G. Farrell. Decoding error-control codes with soft distance as the metric. In *Workshop on Mathematical Techniques in Coding Theory*, Edinburgh, UK, April 2008.

[2] P.G. Farrell and J.Castiñeira Moreira. "soft-input soft-output euclidean distance metric iterative decoder for ldpc codes". In *Argentine Symposium on Computing Technology (AST)*, Santa Fé, Argentina, September 2008.

[3] Shannon C.E. A mathematical theory of communications. *BSTJ*, 27:379–423, September 1948.

[4] R.G. Gallager. *Low-density parity-check codes. Ph.D Thesis*, Massachusetts Inst. of Technol, 1960.

[5] D. Mackay. Good error correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, March 1999.

[6] D. J. C. MacKay and R. M. Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 33(6):457–458, March 1997.

[7] M. K. Simon and M. S. Alouini. *Digital Communication over Fading Channels*. EUA, 2a edition, 2005.

[8] S. Lin. and J. Costello Jr. *Error Control Coding*. 2a edition, 2004.

[9] P.G. Farrell and J. Castiñeira Moreira. *Essentials of Error-Control Coding*. 2006.

[10] R.M Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, pages 533–547, September 1981.

[11] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman. Analysis of low density codes and improved designs using irregular graphs. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC), Maio 1998.*, Palo Alto-CA, May 1998.

[12] S. A. Abrantes. Descodificação iterativa de códigos ldpc por transferência de mensagens em gráficos de factores. Technical report, Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia, Universidade do Porto, Portugal, July 2005.

[13] T. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, February 2001.

[14] M. L. M. N. S. Gonçalves. Códigos ldpc quaternários aplicados à técnica de transmissão ofdm. Tese de mestrado, Unicamp, Campinas-SP, 2010.

[15] D. J. C. Mackay. *Online database of low-density parity-check codes, Website*. Disponível em: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html, 2012.

[16] Z. Li, H. Qin, S. Lin, K. Abdel, and B. F. Ian. "quasi-cyclic ldpc codes on latin squares and the ranks of their parity-check matrices.". *IEEE Trans. Commun.*, 58(11):3126–3129, September 2010.