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

RAPHAEL PARREIRA DE SOUZA

Análise de Desempenho de Códigos LDPC em Cenários 5G *New Radio*

São João da Boa Vista
2022

Raphael Parreira de Souza

Análise de Desempenho de Códigos LDPC em Cenários 5G *New Radio*

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

Orientadora: Profaº Dra. Cintya Wink de Oliveira Benedito

São João da Boa Vista
2022

S729a

Souza, Raphael Parreira de
Análise de desempenho de códigos LDPC em cenários 5G new radio / Raphael Parreira de Souza. -- São João da Boa Vista, 2022
67 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Telecomunicações) - Universidade Estadual Paulista (Unesp), Faculdade de Engenharia, São João da Boa Vista
Orientadora: Cintya Wink de Oliveira Benedito

1. Codificação. 2. Códigos corretores de erros (Teoria da informação). 3. Sistemas de comunicação sem fio. 4. Teoria da informação. 5. Telecomunicações. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Engenharia, São João da Boa Vista. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

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

TRABALHO DE CONCLUSÃO DE CURSO

ANÁLISE DE DESEMPENHO EM CÓDIGOS LDPC EM CENÁRIOS 5G NEW RADIO

Aluno: Raphael Parreira de Souza
Orientador: Prof.ª Dr.ª Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientadora)
- Ivan Aritz Aldaya Garde (Examinador)
- Rafael Abrantes Penchel (Examinador)

A ata da defesa com as respectivas assinaturas dos membros encontra-se no prontuário do aluno (Expediente nº 082/2022)

São João da Boa Vista, 16 de dezembro de 2022

Este trabalho é dedicado aos meus familiares e amigos, que sempre me apoiaram e nunca me deixaram desistir dos meus objetivos.

AGRADECIMENTOS

Aos meus pais Edmilson Luiz de Souza e Sonia Parreira de Souza que me moldaram e me ensinaram o caminho do bem e da busca da sabedoria.

Aos meus irmão Drielle Parreira de Souza e Jefferson Parreira de Souza que em todos os momentos de dificuldade estiveram ao meu lado e nunca soltaram minhas mãos.

À minha namorada Thayane Devecchi Matias que foi a minha maior parceira e companheira durante todo os momentos de alegrias e sofrimentos.

À minha orientadora Profª Dra. Cintya Wink de Oliveira Benedito que aceitou orientar este TCC em um curto espaço de tempo e conseguiu tirar o melhor de mim.

Aos professores Prof. Dr. Ivan Aritz Aldaya Garde e Prof. Dr. Rafael Abrantes Penchel que aceitaram participar da banca e avaliaram este trabalho.

Às minhas amigas Mariana Teixeira e Stephanie Andrade que participaram da criação do trio mais unido e já visto em uma graduação, responsável por me fazer chegar tão longe.

Aos meu primeiros colegas de casa João Victor Caldeirão, Pablo Terra, Thales Braga, e Yago Tonini por me receberem de braços abertos e tornarem os primeiros anos tão leves e divertidos.

Aos últimos colegas de casa Leonardo Mendes, Renan Bairral, e Thierry Ronchi pelos conhecimentos obtidos e não me deixarem desistir nos meus últimos momentos de graduação.

Aos meus amigos Aime Jordan, Arthur Terra, Felipe Franco e Vinicius Moraes por serem meus companheiros e estarem sempre comigo durante toda a trajetória nessa Universidade.

À toda equipe de futsal e do CS:GO que me fizeram viver grandes emoções e se tornaram famílias para mim.

Por fim, agradeço à todas as pessoas que passaram pela minha vida. Cada uma trouxe experiências, conhecimentos, e ideias diferentes que me fizeram evoluir e ser a pessoa que sou hoje.

Muito obrigado!

“Informação é a resolução da incerteza.“
(Claude E. Shannon)

RESUMO

A teoria da informação, dissertada por Shannon em 1948, apresenta conceitos que são utilizados para desenvolvimento das técnicas de codificação de canal. A partir da identificação do que são os códigos de bloco lineares, Gallager propôs um modelo de códigos do tipo bloco com matrizes controle de paridade com baixa densidade de 1's, conhecido como códigos LDPC (*Low Density Parity Check*). Os estudos sobre este código foram esquecidos na comunidade científica por muitos anos. Redescoberto por Mackay e Neal na década de 90, hoje são usados para diversos padrões como IEEE 802.11, DVB-S2, e o objeto desse estudo, o 5G NR (*New Radio*). Para a quinta geração das comunicações móveis (5G), adotou-se a codificação através de construções de matrizes base com expansões em matrizes identidades e, para a decodificação, algoritmos de decisão suave como os algoritmos Soma-Produto (SPA) e o Mínima-Soma (MS). Este trabalho, apresenta um estudo de como são construídas essas matrizes seguindo as normas definidas pelo *3rd Generation Partnership Project* (3GPP), bem como a decodificação através do algoritmo Mínima-Soma em Camadas (LMS) baseado em tangentes hiperbólicas. A partir deste estudo, simulações serão apresentadas de modo que os resultados obtidos demonstram como é a codificação de uma mensagem enviada no padrão 5G NR, e como um algoritmo de alto custo computacional (SPA) pode ser simplificado (LMS) a fim de obter uma rápida convergência sem prejudicar o desempenho. Serão apresentadas simulações para diferentes comprimentos de bloco de informação e diferentes taxas (o que resulta em diferentes matrizes bases). Como parâmetro a ser analisado, a taxa de erro de bits foi selecionada de acordo com o valor mínimo estabelecido para o 5G NR de 10^{-6}. Para todos os tamanhos e taxas foi possível atingir o valor desejado.

PALAVRAS-CHAVE: Códigos de Bloco. Codificação. Decodificação. Códigos LDPC. 5G NR.

ABSTRACT

Information theory, lectured by Shannon in 1948, presents concepts that are used for the development of channel coding techniques. From the identification of what linear block codes are, Gallager proposed a model of block-type codes with parity control matrices with low density of 1's, known as LDPC (Low Density Parity Check) codes. Studies on this code were forgotten in the scientific community for many years. Rediscovered by Mackay and Neal in the 90s, today they are used for several standards such as IEEE 802.11, DVB-S2, and the subject of this study, 5G NR (New Radio). For the fifth generation of mobile communications (5G), encoding was adopted through constructions of base matrices with expansions in identity matrices and, for decoding, soft decision algorithms such as Sum-Product (SPA) and Min-Sum (MS). This work presents a study of how these matrices are built following the rules defined by the 3rd Generation Partnership Project (3GPP), as well as decoding through the Layered Min-Sum (LMS) algorithm based on hyperbolic tangents. From this study, simulations will be presented so that the results will demonstrate how the encoding of a message sent in the 5G NR standard is, and how a high computational cost algorithm (SPA) can be simplified (LMS) in order to obtain a fast convergence without impair performance. There will be simulations for different information block lengths and different rates (which results in different bases matrix). As a parameter to be analyzed, the bit error rate was selected according to the minimum value established for the 5G NR of 10^{-6}. For all sizes and rates it was possible to reach the desired value.

KEYWORDS: Block codes. Coding. Decoding. LDPC codes. 5G NR.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

3GPP	*3rd Generation Partnership Project*
5G	*Fifth generation mobile network*
ARIB	*Association of Radio Industries and Businesses*
ATIS	*Alliance for Telecommunications Industry Solutions*
AWGN	*Additive White Gaussian Noise*
BER	*Bit Error Rate*
BPSK	*Binary Phase-Shift Keying*
CCSA	*China Communications Standards Asssociation*
DVB-S2	*Digital Video Broadcasting Satellite - Second Generation*
ETSI	*European Telecommunications Standards Institute*
FDM	*Frequency Division Multiplexing*
IEEE	*Institute of Electrical and Electronics Engineers*
LDPC	*Low Density Parity Check*
LTE	*Long Term Evolution*
LLR	*Log-likelihood Ratio*
LMS	*Layered Min-Sum*
MIMO	*Multiple-Input Multiple-Output*
MS	*Min-Sum*
NR	*New Radio*
OFDM	*Orthogonal Frequency Division Multiplexing*
QAM	*Quadrature Amplitude Modulation*
QPSK	*Quadrature Phase-Shift Keying*
SNR	*Signal-to-Noise Ratio*
SPA	*Sum-Product*
TTA	*Telecommunications Technology Association*

TTC *Telecommunication Technology Committee*

URC *Ultra-Reliable Communication*

LISTA DE SÍMBOLOS

$\|x\|$	Valor absoluto
$\prod$	Produtório
$\sum$	Somatório
$2W$	Banda do ruído
a	Fator multiplicativo fixo
b	Vetor decodificado
B	Matriz base
B_c	Banda do canal
BG1	Matriz base 1
BG2	Matriz base 2
C	Capacidade do canal
c	Vetor palavra-código
$C(n, k)$	Código de Bloco Linear
c_i	Nó de paridade
d	Densidade
d_H	Distância de Hamming
$E(x)$	Esperança
E_b	Energia por bit
E_s	Energia por símbolo
G	Matriz geradora
GF_2	Corpo de Galois com 2 elementos
H	Matriz controle de paridade
$H(X)$	Entropia
$H(X, Y)$	Entropia conjunta
$H(X\|Y)$	Entropia condicional

$H(Y)$	Entropia
$H(Y\|X)$	Entropia condicional
I	Matriz identidade
$I(X;Y)$	Informação mútua
i_{LS}	Ordem de expansão
J_a	Fator exponencial variável
k	Comprimento dos bits de informação
K_b	Número de colunas de bits de informação
L	Vetor de estimativas de probabilidade
$ln(x)$	Logaritmo natural
m	Número de bits de paridade
m	Mensagem a ser enviada
$\mathcal{M}(n)$	Conjunto de nós de paridade conectados à um nó de variável
$\mathcal{M}(n)\backslash m$	Conjunto de nós de paridade conectados à um nó de variável exceto m
$Min(x)$	Valor mínimo
M_P	Matriz de Probabilidades
n	Comprimento da palavra-código
$\mathcal{N}(m)$	Conjunto de nós de variável conectados à um nó de paridade
$\mathcal{N}(m)\backslash n$	Conjunto de nós de variável conectados à um nó de paridade exceto n
$N(\mu, \sigma^2)$	Distribuição Gaussiana com média μ e variância σ^2
$N_0/2$	Densidade espectral do ruído de dupla banda lateral
N_b	Número de bits enviados
n_e	Número de bits errados
N_r	Densidade espectral do ruído de dupla banda lateral em uma banda dupla
P_n	Potência do ruído
P_s	Potência do sinal
$p(x)$	Função densidade de probabilidade

$p(x, y)$	Probabilidade conjunta
$p(x\|y)$	Probabilidade condicional
$p(y\|x)$	Probabilidade condicional
$Q(x)$	Função distribuição cumulativa complementar
$q_{n\to m}$	Mensagem enviada do nó de variável para o nó de paridade
R	Taxa do código
$\mathbf{r}$	Vetor recebido modulado em BPSK sobre um canal AWGN
$r_{m\to n}$	Mensagem enviada do nó de paridade para o nó de variável
$Sign(x)$	Função sinal
T	Tempo por símbolo
$\tanh(x)$	Tangente hiperbólica
$\tanh^{-1}(x)$	Arco tangente hiperbólica
$\mathbf{u}$	Vetor mensagem enviada
v_j	Nó de variável
w	Mensagem enviada em um canal discreto
w^n	Mensagem decodificada em um canal discreto
ω_H	Peso de Hamming
X	Variável aleatória
$\mathcal{X}$	Alfabeto
x^n	Mensagem codificada em um canal discreto
Y	Variável aleatória
$\mathcal{Y}$	Alfabeto
$\mathbf{y}$	Palavra-código recebida algoritmo
y^n	Palavra-código recebida no decodificador em um canal discreto
Z_c	Fator de expansão

SUMÁRIO

1 INTRODUÇÃO

A quinta geração das redes de comunicações móveis (5G, *Fifth generation mobile network*) apresenta uma nova tecnologia de Novos Rádios (NR, *New Radio*), que vem com o objetivo de cobrir os requisitos que a sua geração anterior, a Evolução à Longo Prazo (LTE, *Long Term Evolution*), não suporta para a quinta geração. O 5G NR foi desenvolvido com objetivo de atingir altas taxas de transmissão de dados, com baixa latência, para aplicações como veículos autônomos, monitoramento de gado e plantações, realização de cirurgias à distância, transmissão de vídeos em ultradefinição, realidade virtual, entre outros [Osseiran, Monserrat e Marsch 2016]. Consegue-se operar em diversas faixas do espectro, o que cria a possibilidade de uso de células de diferentes tamanhos e taxas, inclusive em ondas milimétricas. Utiliza o método de múltiplas entradas e múltiplas saídas (MIMO, *Multiple-Input Multiple-Output*), com o objetivo de aumentar o número de canais para enviar e receber, o que permite o aumento da taxa de dados. A utilização de altas frequências, limita a distância de cobertura, e por isso, esse padrão adota a técnica da formação de feixe (*Beamforming*) para direcionar a transmissão ou recepção. Ela utiliza novos métodos de multiplexação ortogonal por divisão de frequência e novas técnicas de codificação de canal.

Para o desenvolvimento de uma tecnologia de comunicação móvel com alinhamento global, é necessária uma organização capaz de definir requerimentos, padrões e regras de utilização. Em 1998, cinco organizações, o Instituto Europeu de Padrões de Telecomunicações (ETSI, *European Telecommunications Standards Institute*), a Associação das Indústrias e Empresas de Rádio (ARIB, *Association of Radio Industries and Businesses*) e o Comitê de Tecnologia de Telecomunicações (TTC, *Telecommunication Technology Committee*) do Japão, a Associação de Padrões de Comunicações da China (CCSA, *China Communications Standards Asssociation*), a Aliança para Soluções da Indústria de Telecomunicações (ATIS, *Alliance for Telecommunications Industry Solutions*) da América do Norte, e a Associação de Tecnologia de Telecomunicações (TTA, *Telecommunications Technology Association*) da Coreia do Sul, se juntaram com o objetivo de regulamentar o desenvolvimento das gerações das comunicações móveis. Essa congregação recebeu o nome de Projeto de Parceria para a Terceira Geração (3GPP, *3rd Generation Partnership Project*) e ficou responsável por traçar as regras e definições para as gerações das comunicações móveis.

No ano de 2004, essa organização começou a desenvolver o que seria conhecido como a tecnologia LTE, utilizada nas redes de quarta geração (4G). A documentação das especificações e normas, foi finalizada no ano de 2008, o que permitiu que redes 4G fossem implantadas em 2009. Por volta do ano de 2012, começou-se as discussões sobre a quinta geração (5G). O termo 5G, costuma-se misturar com a tecnologia NR, onde muitas vezes 5G é designado para identificá-la. Porém, o 5G NR só começou a ser elaborado em 2016 como objeto de estudo do 3GPP após um *workshop* em 2015. Essa tecnologia veio com o objetivo de preencher as lacunas do LTE para implantação de todas as características e requisitos do 5G. Alguns dos benefícios do NR são:

- utilização do espectro em frequências elevadas, que permite o suporte para maiores larguras de banda de transmissão e as altas taxas de dados associadas;

- melhora de desempenho de energia e redução da interferência por meio de projetos *ultra-lean*;
- preparação para compatibilidade com tecnologias futuras desconhecidas;
- baixa latência para evolução do desempenho e aplicações não suportadas no 4G, como por exemplo veículos autônomos e cirurgias à distância;
- projeto com centralização do feixe, que permite o uso extensivo do *beamforming* e um grande número de antenas para transmissão de dados e procedimentos de plano de controle, como acesso inicial.

Os três primeiros itens citados, são considerados os principais princípios ou requerimentos de projeto do 5G NR. Considerando especificamente a codificação de canal, um dos cenários importantes para o 5G NR é apresentar uma Comunicação Ultraconfiável (URC, *Ultra-Reliable Communication*), a qual tem como objetivo oferecer uma razão de erro de bits de 10^{-6}. Para esse quadro é visto um canal fixo sem a análise de possíveis desvanecimentos [Popovski 2014].

O 3GPP então, em dezembro de 2017, emitiu a primeira especificação oficial sobre o NR, e a cada ano, essa especificação é atualizada com novas informações e normas. De acordo com o relatório técnico do 3GPP sobre o 5G NR [3GPP-138.211 2022], a Multiplexação por Divisão Ortogonal de Frequência (OFDM, *Orthogonal Frequency Division Multiplexing*) será utilizada tanto para *downlink* quanto para *uplink*. Essa técnica combina a Multiplexação por Divisão de Frequência (FDM, *Frequency Division Multiplexing*) com as modulações da portadora por chaveamento de mudança de fase binária (BPSK, *Binary Phase-Shift Keying*), $\pi/2$-BPSK, chaveamento de mudança de fase e quadratura (QPSK, *Quadrature Phase-Shift Keying*), e modulação de amplitude em quadratura (M-QAM, *Quadrature Amplitude Modulation*) com $M = [16, 64, 256, 1024]$, para obtenção de altas taxas de dados. Em relação às técnicas de codificação de canal, são utilizados os códigos polares para os canais de controle e os códigos de verificação de paridade de baixa densidade (LDPC, *Low Density Parity Check*) para os canais de dados [3GPP-138.212 2022]. Nas especificações das Estações Rádio Base, assume-se as condições de propagação como um canal AWGN [3GPP-138.104 2022].

No ano de 1948, Shannon revolucionou as telecomunicações com seu artigo precursor do que é conhecido hoje como a Teoria da Informação. Ele provou que um canal ruidoso, pode-se fazer uma transmissão livre de erros utilizando uma técnica de codificação de canal adequada. Ou seja, uma mensagem pode ser codificada utilizando uma estratégia de codificação que diminua a probabilidade de erro o quanto se queira desde que a sua taxa seja menor ou igual à capacidade do canal [Shannon 1948]. A partir dos resultados apresentados por Shannon, diversas estratégias de codificação de canal começaram a ser apresentadas. Na década de 60, Gallager iniciou seus estudos sobre uma família de códigos de bloco lineares, o código de bloco linear com verificação de paridade de baixa densidade (LDPC) [Gallager 1962]. Por muitos anos, esse código ficou estagnado e novos desenvolvimentos não ocorreram na comunidade científica. Nos anos 90, Mackay e Neal redescobriram esses códigos [MacKay e Neal 1997] com performances próximas aos limites de Shannon sobre um canal gaussiano.

Os códigos LDPC são códigos de bloco lineares com particularidades que aumentam o desempenho da codificação. Eles são compostos, essencialmente, por matrizes controle de paridade com pequenas quantidades de 1's no caso binário. A partir dessas matrizes, é possível encontrar a matriz geradora, que

permite a codificação de uma mensagem qualquer. Existe, porém, a possibilidade dessa matriz geradora ter alta densidade de elementos não-nulos em sua composição, o que dificulta a codificação. Algumas estratégias foram desenvolvidas ao longo dos anos para melhorar essa condição, como por exemplo a construção de Gallager [Gallager 1962] para códigos LDPC regulares, e construções por meio de matrizes base e fatores de expansão [Wang, Yedidia e Draper 2008]. Em relação à decodificação desses códigos, algumas técnicas vem sendo estudadas, como a técnica de *bit-flipping* [Jiang e Fan 2021] que utiliza decisão abrupta (*hard-decision*), e algoritmos de decisão suave (*soft-decision*) como o Soma-Produto (SPA, *Sum-Product*) [Hu et al. 2001] e suas evoluções e simplificações, como o Mínima-Soma (MS, *Min-Sum*) [Zarkeshvari e Banihashemi 2002] e o Mínima-Soma em Camadas (LMS, *Layered Min-Sum*) [Hocevar 2004]. O algoritmo SPA, tratado sobre um canal com ruído aditivo, branco e gaussiano (AWGN, *Additive White Gaussian Noise*), permite a caracterização das estimativas de probabilidades em relações de tangentes hiperbólicas. Isso permite que o algoritmo seja simplificado, reduza o tempo de decodificação e o custo de implementação computacional.

Neste trabalho, os códigos LDPC serão apresentados com o objetivo de analisar o seu desempenho em cenários 5G NR URC. Para isso, será apresentado como a estratégia da construção por matrizes bases foi desenvolvida para codificar mensagens nos padrões do 5G NR. Em relação à decodificação, será descrito como é realizado o SPA baseado em tangentes hiperbólicas, a simplificação para o algoritmo de MS, e a técnica utilizada para melhorar a performance e tempo de processamento (LMS). As simulações serão realizadas utilizando a modulação BPSK, sobre um canal AWGN, que são as características mais simples permitidas para uso no 5G NR, com o objetivo de atingir taxas de erro de bits menores que 10^{-6} como estabelecido para esse padrão [Popovski 2014, Lema et al. 2017].

Este trabalho será divido da seguinte forma: no Capítulo 2 serão abordados os conceitos básicos da teoria da informação e de códigos de bloco lineares. No Capítulo 3 serão introduzidas as definições e particularidades sobre os códigos LDPC, como grafos de Tanner, códigos regulares, construção de Gallager e decodificação por decisão abrupta *bit-flipping*. No Capítulo 4 serão aplicados os conceitos do capítulo anterior para os padrões do 5G NR, com a apresentação das construções das matrizes base e dos algoritmos de decodificação de decisão suave SPA, MS e LMS. No Capítulo 5, serão apresentadas as simulações realizadas com os resultados e discussões pertinentes ao cenário 5G NR. E por fim, no Capítulo 6 as conclusões acerca do trabalho serão apresentadas.

2 CONCEITOS BÁSICOS

Neste capítulo serão apresentados os conceitos básicos necessários para o desenvolvimento dos demais capítulos. A Seção 2.1, apresentará os principais conceitos de teoria da informação, pois para chegar à capacidade de um canal, é preciso conhecer o que é a informação mútua e a entropia de uma variável aleatória. Na Seção 2.2, serão apresentados os códigos de bloco lineares, em especial as matrizes geradora e controle de paridade que são utilizadas para codificação e decodificação destes códigos, respectivamente. A Subseção 2.2.1, exemplificará os códigos de bloco lineares através do código de Hamming (7,4). Estes conceitos são importantes, pois os códigos LDPC são um tipo de código de bloco linear.

2.1 TEORIA DA INFORMAÇÃO E CODIFICAÇÃO

A teoria da informação traz consigo conceitos básicos como entropia, entropia conjunta e informação mútua, que são indispensáveis para a definição sequente de capacidade do canal. Esta seção apresentará esses conceitos para o caso discreto e para o caso contínuo, que permitirá caracterizar um canal gaussiano. Com base para os estudos, serão utilizados definições e conceitos dos livros [Cover e Thomas 2006, Goldsmith 2005].

Um evento aleatório possui uma certa probabilidade de acontecer. Caso esse evento seja certo, não há incerteza associada à ele. Para situações em que a probabilidade seja diferente da certeza, existe uma informação associada à quantidade de incerteza que envolve a variável aleatória representativa do evento. Essa medida de incerteza da ocorrência de um evento, recebe o nome de entropia. Seja X uma variável aleatória com alfabeto $\mathcal{X}$ e função densidade de probabilidade $p(x) = Pr\left\{X = x\right\} x \in \mathcal{X}$.

Definição 2.1.1 *A entropia $H(X)$ de uma variável aleatória discreta X é definida por*

$$H(X) = E\left\{\log \frac{1}{p(x)}\right\} = \sum_{x\in\mathcal{X}} p(x) \log \frac{1}{p(x)}, \tag{1}$$

onde o operador $E\left\{x\right\}$ representa o valor esperado de x e $p(x)$ representa a probabilidade da variável aleatória X ser igual à x.

Os logaritmos utilizado na Equação 1 e nas próximas equações, podem ser expressos em base 2. Assim, as entropias serão dadas em bits. Também, em cálculos de entropia, convenciona-se que $0 \log 0 = 0$. Isso se explica devido à continuidade $x \log x \to 0$ quando $x \to 0$.

Sejam duas variáveis aleatórias X com alfabeto $\mathcal{X}$ e Y com alfabeto $\mathcal{Y}$ com distribuição conjunta de probabilidades $p(x, y)$ e distribuição de probabilidades condicionais $p(x|y)$ e $p(y|x)$.

Definição 2.1.2 *A entropia conjunta $H(X, Y)$ de duas variáveis aleatórias (X,Y) é dada por*

$$H(X, Y) = E\left\{\log \frac{1}{p(x, y)}\right\} = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x, y) \log \frac{1}{p(x, y)}, \tag{2}$$

onde o operador $E\{(x,y)\}$ representa o valor esperado de (x,y) e $p(x,y)$ representa a probabilidade conjunta do par de variáveis aleatórias (X,Y) ser igual à (x,y).

Definição 2.1.3 *A entropia condicional $H(Y|X)$ de duas variáveis aleatórias (X,Y) é dada por*

$$H(Y|X) = E\left\{\log \frac{1}{p(y|x)}\right\} = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(y|x)\log\frac{1}{p(y|x)}, \tag{3}$$

onde o operador $E\{(y|x)\}$ representa o valor esperado de $(y|x)$ e $p(y|x)$ representa a probabilidade condicional da variável aleatória Y dado um valor específico da variável aleatória X.

A definição de informação é a medida da quantidade de incerteza de um processo que ocorra com uma certa probabilidade. Ao considerar duas variáveis aleatórias X e Y, a quantidade de informação que uma dessas variáveis contém sobre a outra recebe o nome de informação mútua. Ela reduz a incerteza de uma variável aleatória devido o conhecimento da outra.

Definição 2.1.4 *Sejam duas variáveis aleatórias X com alfabeto $\mathcal{X}$ e Y com alfabeto $\mathcal{Y}$, com distribuição conjunta de probabilidades $p(x,y)$ e distribuições marginais $p(x)$ e $p(y)$, respectivamente. A informação mútua $I(X;Y)$ é dada por*

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}. \tag{4}$$

Por meio da probabilidade condicional

$$p(x|y) = \frac{p(x,y)}{p(y)}, \tag{5}$$

a Equação 4 pode ser reescrita como

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x|y)}{p(x)}, \tag{6}$$

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\frac{1}{\log p(x)} - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\frac{1}{p(x|y)}. \tag{7}$$

Finalmente, ao substituir (1) e (3) em (7), a informação mútua pode ser escrita por meio da subtração entre a entropia de uma das variáveis aleatórias e a entropia condicional da mesma variável aleatória dada a outra. Ou seja,

$$I(X;Y) = H(X) - H(X|Y). \tag{8}$$

A Equação 8 pode ser escrita em função da entropia de qualquer uma das variáveis. Isso significa que

$$H(X) - H(X|Y) = H(Y) - H(Y|X), \tag{9}$$

então, a informação mútua pode ser descrita por meio da soma entre as entropias de cada uma das variáveis aleatórias e sua entropia conjunta

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{10}$$

As definições das entropias e informação mútua para variáveis aleatórias discretas, se estendem para variáveis aleatórias contínuas. A entropia recebe o nome de entropia diferencial e possui muitas semelhanças com o caso discreto.

Definição 2.1.5 *A entropia diferencial $h(X)$ de uma variável aleatória contínua X com distribuição de densidade de probabilidade $f(x)$ é definida por*

$$h(X) = -\int_X f(x)\log f(x)dx. \tag{11}$$

Tanto a entropia discreta quanto a entropia diferencial, dependem apenas da densidade de probabilidade da sua respectiva variável aleatória. Também, as entropias conjuntas, condicionais, e a informação mútua seguem as mesmas definições do caso discreto. Sejam duas variáveis aleatórias contínuas X e Y, com distribuição de densidade de probabilidade $f(x)$ e $f(y)$, respectivamente, distribuição conjunta $f(x,y)$ e distribuição condicional $f(y|x)$, tem-se as seguintes definições.

Definição 2.1.6 *A entropia conjunta $h(X,Y)$ de duas variáveis aleatórias contínuas X e Y será*

$$h(X) = -\int_X\int_Y f(x,y)\log f(x,y)dxdy. \tag{12}$$

Definição 2.1.7 *A entropia condicional $h(X,Y)$ de duas variáveis aleatórias contínuas X e Y será*

$$h(Y|X) = -\int_X\int_Y f(x,y)\log f(y|x)dxdy. \tag{13}$$

Definição 2.1.8 *A informação mútua $I(X;Y)$ de duas variáveis aleatória contínuas X e Y será*

$$I(X;Y) = \int_X\int_Y f(x,y)\log\frac{f(x,y)}{f(x)f(y)}dxdy. \tag{14}$$

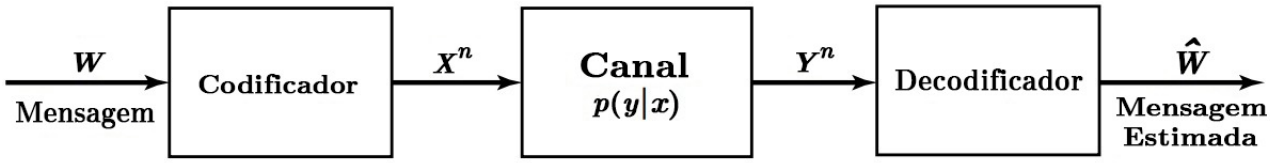
A mesma relação obtida em (10), descreve a informação mútua para o caso contínuo, ou seja,

$$I(X;Y) = h(X) + h(Y) - h(X,Y). \tag{15}$$

2.1.1 Canal de Comunicação

Em um sistema de comunicações, o canal é definido como o meio físico de transporte da informação de um ponto à outro. Ele causa degradação no sinal, devido à fatores como ruídos e imperfeições em sua estrutura. Para cada informação que deseja-se enviar, há uma probabilidade correspondente de, após passar pelo canal, chegar de maneira correta ao destino. Algumas estratégias são utilizadas para otimizar essas probabilidade afim de reduzir os erros de comunicação.

Figura 1 – Sistema de Comunicações Discreto



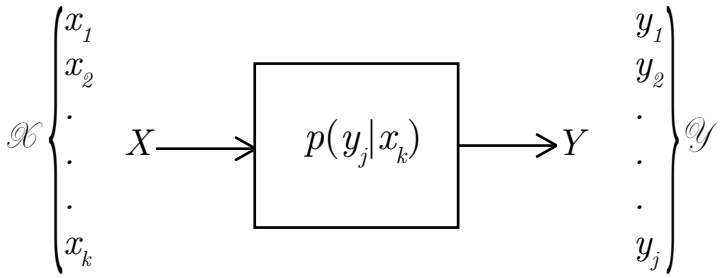
Fonte: Produção do próprio autor.

Considere as partes que o compõe visto na Figura 1. Uma mensagem w que deseja ser enviada à um determinado destinatário é codificada e se transforma em um conjunto de símbolos x^n. Ao ser transmitida por um canal, ela sofre interferência de ruídos. Por essa razão, a palavra y^n recebida e decodificada em uma mensagem $\hat{w}$, não é necessariamente igual à mensagem enviada.

Se define canal discreto o conjunto de símbolos de entrada representados por um alfabeto finito $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$, o conjunto de símbolos de saída por um alfabeto finito de saída $\mathcal{Y} = \{y_1, y_2, \ldots, y_j\}$ e a transmissão por uma lei de transição composta de uma matriz de probabilidades $p(y|x)$. Um canal no qual os símbolos enviados não exercem influência na próxima saída, é definido como canal sem memória.

Definição 2.1.9 *Um canal discreto sem memória é aquele onde a saída depende exclusivamente de uma respectiva entrada. Não é condicional aos símbolos transmitidos anteriormente.*

Figura 2 – Canal discreto e sem memória



Fonte: Produção do próprio autor.

Dados os alfabetos de entrada $\mathcal{X}$ e saída $\mathcal{Y}$, as probabilidades de transições podem ser escritas como a probabilidade condicional. Em outras palavras, a probabilidade de observar o símbolo de saída y_j para um conhecido símbolo enviado x_k,

$$p(y_j|x_k) = P(\mathcal{Y} = y_j|\mathcal{X} = x_k) \; \forall \; 0 \leq p(y_j|x_k) \leq 1. \tag{16}$$

Através de (16) é possível montar uma matriz que apresenta todas as probabilidades condicionais, ou seja, para cada uma das saídas apresentar quais são as probabilidades de cada um dos símbolos de

entrada isoladamente ter sido enviado

$$M_P = \begin{bmatrix} p(y_1|x_1) & p(y_2|x_1) & \cdots & p(y_j|x_k) \\ p(y_1|x_2) & p(y_2|x_1) & \cdots & p(y_j|x_k) \\ \vdots & \vdots & \ddots & p(y_j|x_k) \\ p(y_1|x_k) & p(y_2|x_k) & \cdots & p(y_j|x_k) \end{bmatrix}. \tag{17}$$

Esta matriz terá dimensões $k \times j$, onde as linhas representam cada uma das entradas independentes do canal e as colunas cada uma das saídas independentes. A partir dessa matriz pode-se definir as probabilidades a priori e posteriori, sendo elas:

- $p(x_k)$: probabilidades a priori
- $p(y_j)$: probabilidades a posteriori

Em um canal discreto, existe um limite de envio de mensagem para que não ocorram erros de transmissão devido ao uso do canal. Esse limite é caracterizado por reduzir as probabilidades condicionais para que cada uma das saídas correspondam à apenas um símbolo de entrada.

Definição 2.1.10 *Seja um canal discreto com alfabeto de entrada $\mathcal{X}$ e alfabeto de saída $\mathcal{Y}$, a capacidade do canal é dada por*

$$C = \max_{p(x)} I(X;Y). \tag{18}$$

No caso binário, a capacidade do canal representa o limite máximo de bits capazes de serem transmitidos sem que ocorra erro por uso do canal. Matematicamente, é dado pela máxima informação mútua, como visto em (4), sobre a distribuição de probabilidade $p(x)$.

Um caso particular de um canal de comunicação, é o canal binário gaussiano contínuo. Dada uma variável aleatória binária contínua X com distribuição gaussiana $N(\mu, \sigma^2)$, com média μ e variância σ^2, a entropia será

$$h(x) = \frac{1}{2}\log_2(2\pi e\sigma^2)\ bits, \tag{19}$$

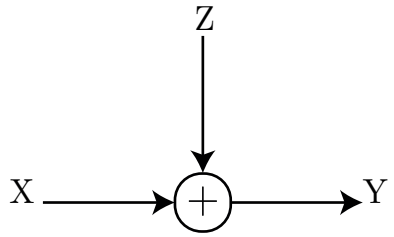
ou seja, a entropia de uma variável aleatória gaussiana não depende da sua média, é apenas determinada através da sua variância.

O canal Gaussiano recebe esse nome, porque é inserido um ruído AWGN, que representa um ruído aditivo, branco (se refere à todas as faixas de frequência), e com distribuição gaussiana. Caso a variância desse ruído seja igual a 0, pode-se dizer que o sistema é sem ruído e toda a informação enviada será recebida sem erros. Para valores de variância cada vez maiores, significa uma maior probabilidade da mensagem recebida conter erros, mas ainda assim, existem conjuntos de entradas que podem ser selecionadas para que a saída possa distingui-las com menor chance de erro quanto se queira.

Se a variância for diferente de zero, significa que o sistema possui restrição de potência. De acordo com a Figura 3, dado que $E[X^2] \leq P_s$ e $E[Z^2] = \sigma^2 = N_r$, a saída é $Y = X + Z$, e que a entrada do canal e o ruído são independentes, então

$$E[Y^2] = E[(X+Z)^2] = E[X^2] + 2E[XZ] + E[Z^2] = E[X^2] + E[Z^2] \leq P_s + N_r, \tag{20}$$

Figura 3 – Canal gaussiano



Fonte: Produção do próprio autor.

como X e Z são independentes, a informação mútua pode ser reduzida a

$$I(X;Y) = h(Y) - h(Y|X) = h(Y) - h(X+Z|X) = h(Y) - h(Z|X) = h(Y) - h(Z). \quad (21)$$

As entropias relacionadas à capacidade do canal gaussiano

$$\begin{aligned} h(Y) &= \frac{1}{2}\log_2(2\pi e(P_s + N_r)), \\ h(Z) &= \frac{1}{2}\log_2(2\pi e N_r), \end{aligned} \quad (22)$$

são relacionadas através da subtração

$$h(Y)-h(Z) = \frac{1}{2}\log_2(2\pi e(P_s+N_r)) - \frac{1}{2}\log_2(2\pi e N_r) = \frac{1}{2}\log_2(\frac{P_s+N_r}{N_r}) = \frac{1}{2}\log_2(1+\frac{P_s}{N_r}). \quad (23)$$

Logo a capacidade do canal gaussiano será

$$C = \max I(X;Y) = \frac{1}{2}\log_2\left(1+\frac{P_s}{N_r}\right). \quad (24)$$

E, se o canal é limitado em banda $2B_c$ será

$$C = 2B_c\frac{1}{2}\log_2\left(1+\frac{P_s}{N_r}\right) = B_c\log_2\left(1+\frac{P_s}{N_r}\right). \quad (25)$$

2.2 CÓDIGOS DE BLOCO LINEARES

A princípio nesta seção será definido o que são os códigos de bloco lineares, com a apresentação de sua matriz geradora e matriz controle de paridade, nas formas sistemáticas. Será apresentado como uma palavra-código é verificada através do vetor síndrome e serão exemplificados, o peso e a distância de Hamming, conceitos utilizados na construção dos códigos LDPC no Capítulo 3. A Seção 2.2.1 apresentará um exemplo de código de bloco linear chamado código de Hamming (7,4), com a definição do que é a taxa de um código e os conceitos de decisões abruptas e suave, que serão utilizadas nos algoritmos de decodificação que serão apresentados na Seção 4.2 [Baldi 2014]. Os conceitos dos códigos de bloco lineares envolvidos nesse capítulo, se limitarão ao caso binário, ou seja, sobre o corpo de Galois ou corpo finito com 2 elementos GF_2 [Lin e Costello 2001].

Definição 2.2.1 *Seja o corpo finito GF_2^k que indica um espaço vetorial de dimensão k sobre GF_2. Um código de bloco de comprimento n e dimensão k, com 2^k palavras-códigos é chamado código de bloco linear se, e somente se, essas 2^k palavras-códigos formam um subespaço vetorial de dimensão k do corpo binário GF_2^n.*

Um código de bloco com dimensão k e comprimento n é representado por $C(n,k)$, de modo que a aplicação linear

$$C : GF_2^k \rightarrow GF_2^n, \tag{26}$$

associa cada vetor binário de informação de comprimento k a um vetor binário de palavra-código de comprimento n. Se $C(n,k)$ é um código de bloco linear, existem k vetores $\{\mathbf{g}_0, \ldots, \mathbf{g}_{k-1}\}$ linearmente independentes que formam uma base desse subespaço. Cada uma das palavras-códigos $\mathbf{c} = [c_0, \ldots, c_{n-1}] \in C$ pode ser expressa como uma combinação linear destes vetores

$$\mathbf{c} = u_0\mathbf{g}_0 + \ldots + u_{k-1}\mathbf{g}_{k-1}, \tag{27}$$

com o vetor de informação $\mathbf{u} = [u_0, \ldots, u_{k-1}]$. Em forma matricial, a Equação 27 pode ser escrita como

$$\mathbf{c} = \mathbf{u} \cdot G, \tag{28}$$

com a matriz geradora G com dimensões $k \times n$, onde k é o comprimento do vetor de informação e n é o comprimento da palavra-código. Essa matriz G, é utilizada para codificar os vetores de informações.

Um código de bloco é chamado sistemático se todas as palavras-códigos contém o vetor de informação associado à ela. Um código de bloco linear na forma sistemática apresenta uma matriz geradora G na forma

$$G = [I|P], \tag{29}$$

onde I é uma matriz identidade de dimensões $k \times k$, e P uma matriz de dimensões $k \times m$ que representa o conjunto de equações de verificação de paridade. Neste caso, cada palavra-código será escrita inicialmente com o vetor de informação $\mathbf{u}$ seguido dos $m = n - k$ bits de redundância, como segue

$$\mathbf{c} = [u_0, \ldots, u_{k-1}|p_0, \ldots, p_{m-1}], \tag{30}$$

onde cada bit p_i é expressado em termos da função de verificação de paridade.

O conjunto $n - k$ de vetores de comprimento n linearmente independentes $\{\mathbf{h}_0, \ldots, \mathbf{h}_{n-k-1}\}$ que formam uma base ortogonal ao conjunto das palavras-códigos, podem ser escrito na forma matricial:

$$H = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{n-k-1}. \end{bmatrix}. \tag{31}$$

O teorema de Fredholm mostra que o complemento ortogonal do espaço de H é o espaço nulo de H. Portanto, é possível afirmar que a matriz H, também conhecida como matriz controle de paridade

de um código $C(n,k)$ deve satisfazer a relação:

$$H \cdot \mathbf{c}^T = \mathbf{0}, \tag{32}$$

onde $\mathbf{0}$ representa um vetor nulo de comprimento $n-k$. Os vetores resultantes dessa relação $\mathbf{s} = H \cdot \mathbf{c}^T$ são chamados de síndrome de $\mathbf{c}$ através de H. Para qualquer palavra-código pertencer ao código de bloco linear C, a síndrome deve ser um vetor nulo. Essa matriz controle de paridade, é utilizada para realizar a decodificação de um código de bloco linear.

As matrizes geradora e controle de paridade descrevem por completo um código de bloco linear. Elas são responsáveis por realizar a codificação de um vetor de informação e a decodificação de uma palavra-código para encontrar o vetor de informação correspondente. Ao substituir a Equação 28 na Equação 32, obtém-se:

$$H \cdot \mathbf{c}^T = H \cdot G^T \cdot \mathbf{u}^T = \mathbf{0},\ \forall \mathbf{u} \in GF_2^k, \tag{33}$$

e então, expressa-se a relação entre a matriz geradora e a controle de paridade para representar o código de bloco linear,

$$H \cdot G^T = \mathbf{0}. \tag{34}$$

Ainda, ao retomar (29), é possível escrever a matriz controle de paridade H na sua forma sistemática:

$$H = [P^T | I]. \tag{35}$$

Duas métricas importantes dos códigos de bloco lineares são o peso e a distância de Hamming. Esta apresenta a relação da diferença entre os bits em cada posição de duas palavras-código, e aquela apresenta o número de elementos não-nulos de uma palavra-códdigo.

Definição 2.2.2 *O número de bits diferentes, em suas respectivas posições, entre dois vetores de comprimento n, é denominado como distância de Hamming, ou seja,*

$$d_H(u,v) = \#\left\{i | u_i \neq v_i, 1 \leq i \leq n\right\}. \tag{36}$$

A menor distância encontrada entre as palavras-códigos de um código de bloco linear C é chamada de distância mínima de Hamming, $d_H(C)$, do código C .

Definição 2.2.3 *A quantidade de valores não-nulos de um vetor ou palavra-código é definido como $\omega_H(v)$, peso de Hamming. Em um código de bloco linear C, o peso mínimo de Hamming, $\omega_H(C)$ é o menor peso entre todas as palavras-códigos.*

O peso envolve o cálculo que utiliza apenas um vetor. Sendo assim, para encontrar a distância mínima de Hamming, utiliza-se o Teorema 2.2.1.

Teorema 2.2.1 *Seja C um código de bloco linear e $u, v \in C$,*

1. $d_H(u,v) = \omega_H(u-v)$.

2. $d_H(C) = \omega_H(C)$.

Por meio desse Teorema, observa-se que para um código de bloco linear $C(n, k)$ a distância mínima de Hamming entre duas palavras-código coincide com o peso mínimo.

Exemplo 2.2.1 *Seja o código de bloco linear* $C = \{0000, 1000, 1110, 1111, 0110, 0111, 0001, 1001\}$. *Têm-se as distâncias de Hamming*

$$d_H(1000, 1110) = 2,$$
$$d_H(0000, 1000) = 1,$$
$$d_H(1111, 0001) = 3,$$

e o peso de Hamming correspondente a cada um desses vetores será

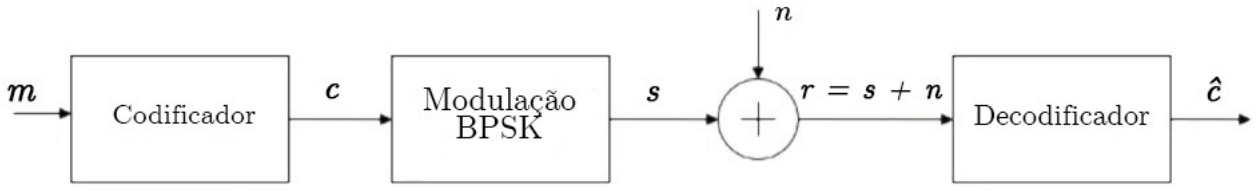
$$\omega_H(1000) = 1,$$
$$\omega_H(1110) = 3,$$
$$\omega_H(0000) = 0,$$
$$\omega_H(1000) = 1,$$
$$\omega_H(1111) = 4,$$
$$\omega_H(0001) = 1.$$

Logo, $d_H(C) = w_H(C) = 1$.

2.2.1 Códigos Hamming

Para exemplificar as características dos códigos de bloco lineares, nesta seção será apresentado o código de Hamming com parâmetros $(7, 4)$. Também, será apresentado os conceitos de decisões suaves (*soft-decision*) e abruptas (*hard-decision*) em algoritmos de decodificação.

Figura 4 – Diagrama Código de Hamming



Fonte: Produção do próprio autor.

A mensagem **m** tem comprimento de 4 bits. Após a codificação, a mensagem se torna uma palavra-código de 7 bits. Em sequência, ela será mapeada em uma modulação BPSK sobre um canal AWGN, que receberá um ruído $N(0, \sigma^2)$. Essa modulação tem como característica, o mapeamento dos bits $0 \rightarrow +1$ e $1 \rightarrow -1$. Na entrada do decodificador, será recebido o vetor $\mathbf{r} = [r_1\ r_2\ \ldots\ r_n]$ que será a soma da palavra-código mapeada em BPSK com o ruído gaussiano. Na saída do decodificador, apresentará a palavra-código interpretada no final do sistema de comunicação. As palavras código relacionadas às mensagens são dadas na Tabela 1.

Tabela 1 – Mensagem e palavra-Código equivalente

Mensagem	Palavra-Código
0000	0000000
0001	0001011
0010	0010110
0011	0011101
0100	0100111
0101	0101100
0110	0110001
0111	0111010
1000	1000101
1001	1001110
1010	1010011
1011	1011000
1100	1100010
1101	1101001
1110	1110100
1111	1111111

Fonte: Produção do Próprio Autor.

Esse código possui algumas características importantes. Primeiramente, é possível citar que é um código na forma sistemática, visto que todas as 16 palavras-códigos apresentam o vetor informação seguido dos bits de paridade. Também, nota-se que esse código é cíclico e, efetuar a soma(mod2) de quaisquer duas palavras-códigos, retornará outra palavra-código válida. É possível observar que o código possui uma taxa de $R = 4/7$.

Definição 2.2.4 *Dado um código de bloco linear com parâmetros $C(n, k)$, a taxa do código é dada pela razão entre o número de bits de informação e o número de bits da palavra-código.*

$$R = \frac{k}{n}. \tag{37}$$

A taxa de um código se relaciona com a capacidade de um canal descrita na Definição 2.1.10. Ela é sempre menor que a capacidade do canal em que ocorrerá a transmissão, e caso exceda o limite, nenhuma informação útil será transmitida através desse sistema. Com isso, encontra-se o Teorema de Shannon.

Teorema 2.2.2 *Se em um sistema de comunicação a taxa do código é menor que a capacidade do canal*

$$R < C, \tag{38}$$

existe uma técnica de codificação que reduz a probabilidade de erro na recepção o quanto se queira.

A etapa da decodificação pode ser realizada através de dois tipos de decisões: a decisão abrupta (*hard-decision*), e a decisão suave (*soft-decision*). Com a decisão abrupta, ele interpreta cada um dos bits recebidos e compara com o limiar de decisão se $r_i \geq 0$, $b_i = 0$, e se $r_i < 0$, $b_i = 1$. Para o código de Haming $(7, 4)$, o vetor decodificado $\mathbf{b} = [b_1\ b_2\ \ldots\ b_n]$, pode ser qualquer uma das $2^k = 2^7 = 128$

possibilidades. Este será relacionado à todas as palavras-códigos **c** para encontrar qual minimiza a distância de Hamming entre eles, e então tomar a decisão da palavra-código recebida.

Exemplo 2.2.2 *Dado o vetor decodificado* $\boldsymbol{b} = [1\ 0\ 1\ 0\ 1\ 0\ 1]$*, comparamos este vetor com cada uma das palavras-código apresentadas na Tabela 1. Têm-se que a menor distância de Hamming é*

$$d_H(1010101, 1000101) = 1.$$

Portanto, a palavra-código interpretada como sendo a palavra-código enviada será $\boldsymbol{c} = [1\ 0\ 0\ 0\ 1\ 0\ 1]$.

O segundo decodificador a ser utilizado, será o de máxima verossimilhança. É um decodificador de decisão suave que ao invés de encontrar o vetor decodificado e fazer as comparações de distância de Hamming, ele utiliza os vetores de número reais recebidos para realizar o cálculo da distância euclidiana $|\mathbf{r} - (1 - 2\mathbf{c})|^2$ com as palavras-códigos e encontrar a menor. É muito mais complexo do que o decodificador de decisão abrupta, porém estritamente mais preciso.

Exemplo 2.2.3 *Dado o vetor de números reais recebido* $\boldsymbol{b} = [1,1\ 0,5\ -0,8\ 1,5\ -0,1\ 0,6\ -0,1]$*, comparamos novamente este vetor com cada uma das palavras-código apresentadas na Tabela 1, porém neste caso, mapeadas em BPSK. Têm-se que a menor distância euclidiana entre elas é dada por* $\boldsymbol{c} = [0\ 1\ 1\ 0\ 0\ 0\ 1]$*,*

$$|\boldsymbol{r} - \boldsymbol{s}| = (1,1-1)^2 + (0,5-(-1))^2 + \ldots + ((-0,1)-(-1))^2 = 4,73.$$

Portanto, a palavra-código interpretada como sendo a palavra-código enviada será $\boldsymbol{c} = [0\ 1\ 1\ 0\ 0\ 0\ 1]$.

3 CÓDIGOS LDPC

Os códigos de verificação de paridade de baixa densidade (LDPC, *Low-Density Parity-Check*) são códigos da família dos códigos de bloco lineares. Inicialmente, foi descoberto por Gallager [Gallager 1962] no início do anos 60, porém não foi valorizada pela comunidade científica. Gallager apesar de caracterizá-los, ele não propôs estratégias ótimas para a construção e decodificação desses códigos. Deixou apenas a proposta de criação de classes pseudoaleatórias dos códigos LDPC. Os trabalhos sobre algoritmos de decodificação por decisão suave, através de estimativas de probabilidades, permitiram encontrar um caminho para obter taxas de transmissão próximas à capacidade do canal e com taxas de erros próxima aos limites de Shannon [MacKay e Neal 1997]. Recentemente, através de técnicas de codificação e decodificação eficientes esses códigos retornaram a aparecer em grandes números de estudos [Richardson e Urbanke 2003].

A Seção 3.1 apresentará as definições básicas dos códigos LDPC que serão úteis para o desenvolvimento do Capítulo 4. Na Seção 3.2 será identificado o que são códigos LDPC regulares e exemplificado uma técnica de construção da matriz controle de paridade proposta por Gallager. Por fim, na Seção 3.3, um método clássico de decodificação para essa família de códigos será descrito. Os conceitos desse capítulo foram baseados em [Baldi 2014, Lin e Costello 2001].

3.1 DEFINIÇÃO DE CÓDIGOS LDPC

Os códigos LDPC binários são caracterizados por possuírem uma matriz controle de paridade, de grandes dimensões, esparsa. Isto é, tem uma baixa concentração de 1's e uma alta concentração de 0's. A definição de matriz esparsa se baseia na sua densidade. Caso a densidade de 1's seja menor ou igual à 0,5, ou seja, $d \leq 0,5$, a matriz é considerada esparsa.

O estudo dos grafos bipartidos, também conhecido como Grafo de Tanner em aplicações de códigos de bloco lineares, é de suma importância para os códigos LDPC. Eles descrevem códigos de bloco lineares, em diagramas representados por nós, que se relacionam através de conexões chamadas arestas, e assim, podem ser obtidas matrizes H de controle de paridade [Tanner 1981]. A aplicação do uso desses grafos, pode ser observado no Capítulo 4, na Seção 4.2.

Definição 3.1.1 *Um grafo bipartido é definido como um grafo onde os nós são divididos em duas classes. Entre os nós de uma mesma classe, não há conexão, ou seja, não possuem arestas que os interligam.*

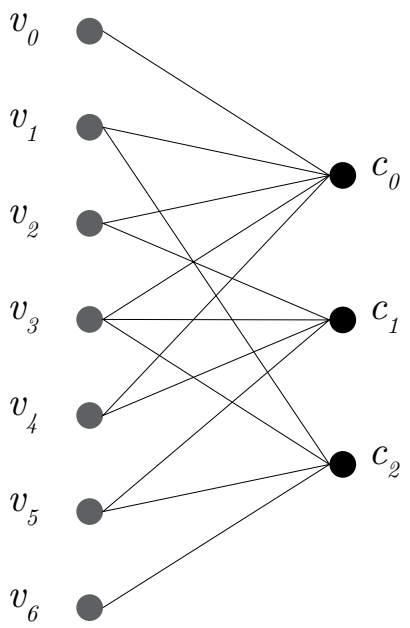
Para códigos de bloco lineares, as duas classes recebem nomes específicos. São eles os nós de variáveis que representam os n bits das palavras-códigos, e os nós de paridade que representam os $m = n - k$ bits de paridade. Para o desenvolvimento da apresentação desses grafos, assume-se que os nós de de variáveis serão definidos como v_j, e os nós de paridade c_i. A matriz controle de paridade H, gerada através desses grafos, será composta por elementos h_{ij}, onde i representa o número de linhas dessa matriz e j o número de colunas.

Definição 3.1.2 *A matriz controle de paridade gerada a partir de um grafo bipartido, tem cada uma de suas posições h_{ij} representada por 1's se e somente se existe uma aresta que conecta o nó v_j ao nó c_i. Caso não haja conexão, a posição h_{ij} é representada por 0.*

De acordo com a Definição 3.1.2, a partir da observação de um grafo bipartido, é possível construir a matriz controle de paridade. Da, mesma forma, se observar uma matriz controle de paridade, é possível encontrar o grafo bipartido correspondente. Assim, será exemplificado o primeiro caso, onde é dado um determinado grafo e encontrada a sua respectiva matriz H.

Exemplo 3.1.1 *Seja um código de bloco linear com parâmetros $C(7, 4)$, com a representação do seu grafo bipartido na Figura 5. Construir a matriz controle de paridade que representa esse grafo.*

Figura 5 – Grafo de Tanner de um código $C(7, 4)$



Fonte: Adaptado de [Baldi 2014].

A partir da definição de como gerar a matriz controle de paridade ao observar um grafo bipartido, nota-se que para o nó v_0, só há conexão com o nó c_2. Ou seja, na primeira coluna só há 1 na linha 3 que representa h_{20}. Adota-se a mesma análise para todos os outros nós, então a matriz controle de paridade será

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

O grau de um nó, é definido a partir do número de arestas conectados à ele. Consequentemente, o grau do nó de variável coincide com o peso de Hamming da coluna associada, e o grau do nó de paridade com o peso da linha da matriz controle de paridade.

O conceito de ciclos associado aos grafos também é importante para a compreensão da qualidade de um código LDPC. Ele será responsável por determinar o que recebe o nome de *girth*.

Definição 3.1.3 *Em um grafo de Tanner, o ciclo de comprimento l é o caminho das l arestas partindo de um nó e voltando para ele mesmo, passando por outros nós não repetidos.*

Isso significa que ao observar um nó de paridade c_i, ele irá se conectar à um nó de variável v_j, que em sequência irá se conectar à outro nó c_k, com $k \neq i$, e assim sucessivamente até que retorne ao nó de paridade c_i. A partir dos ciclos, chega-se à um conceito importante do *girth*.

Definição 3.1.4 *Em um grafo de Tanner, o menor comprimento entre todos os ciclos que o compõe, é denominado girth.*

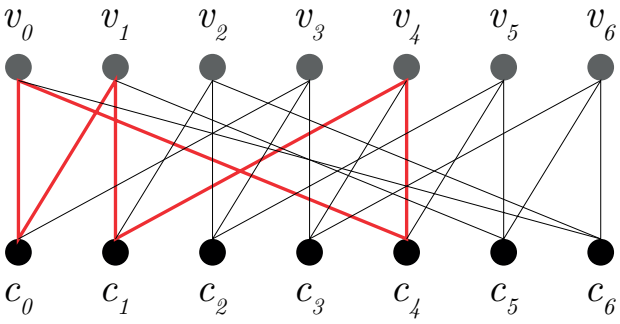
Essa definição tem impacto no bom desempenho de um código LDPC. Para esse tipo de código, é desejável *girth* pelo menos igual à 6. Para isso, a matriz controle de paridade não pode possuir dois 1's em comum entre duas colunas. Isso significa que ao tomar duas colunas vizinhas quaisquer, somente existe um ou nenhum 1 na mesma linha. Para decodificações iterativas baseadas em propagação de crença vista no Capítulo 4, na Seção 4.2, é ainda mais importante que seja maior que 6. A seguir, será demonstrado um exemplo onde vai ser definido como encontrar o *girth* de um código LDPC.

Exemplo 3.1.2 *Seja um código de bloco linear LDPC com parâmetros $C(7,3)$, com a matriz controle de paridade*

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

o seu grafo de Tanner é obtido a partir da matriz H e representado na Figura 6.

Figura 6 – Grafo de Tanner do código $C(7,3)$



Fonte: Adaptado de [Lin e Costello 2001].

Através da matriz controle de paridade, nota-se que não há dois 1's na mesma linha em colunas vizinhas. Portanto, o girth é maior que 4. Através da linha mais forte em vermelho no grafo de Tanner da Figura 6 e de acordo com a Definição 3.1.3, é possível notar um ciclo de comprimento 6. Esse ciclo se inicia em v_0, passa por c_0, v_1, c_1, v_4, c_4, e retorna para v_0. Logo, o girth desse código de bloco linear é 6.

3.2 CÓDIGOS LDPC REGULARES

Os códigos LDPC, quando possuem um número fixo de peso de Hamming em todas as colunas e nas linhas, não necessariamente o peso das linhas deve ser igual ao das colunas, recebem o nome de códigos LDPC regulares. Esse código, têm como parâmetros $C(n, \omega_c, \omega_r)$. Para essa especificação de código, convenciona-se utilizar $\omega_c \geq 3$ e $\omega_r \geq \omega_c$.

Para que um código LDPC regular tenha uma baixa densidade de 1's em sua matriz controle de paridade, é desejado que o peso das colunas seja muito menor que o número de bits de paridade, ou seja, $\omega_c << n - k$. Também, é desejado que o peso das linhas seja muito menor que o comprimento da palavra-código, ou seja, $\omega_r << n$. Isso caracteriza a necessidade das matrizes possuírem grandes dimensões.

A quantidade de bits 1's na matriz H será dado tanto por $\omega_r m$ que é a multiplicação entre o peso das linhas e o número $m = n - k$ de bits de paridade, quanto por $\omega_c n$ que é a multiplicação entre o peso das colunas e o comprimento da palavra-código . A sua taxa, será dada a partir da Equação 37,

$$R = \frac{n}{k} = 1 - \frac{\omega_c}{\omega_r}, \tag{1}$$

e por consequência, a densidade será dada por

$$d = \frac{\omega_r}{n} = \frac{\omega_c}{m}. \tag{2}$$

Exemplo 3.2.1 *Dada a matriz H controle de paridade de um código LDPC regular* $(12, 3, 6)$

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

A taxa desse código será

$$R = 1 - \frac{\omega_c}{\omega_r} = 1 - \frac{3}{6} = 0,5,$$

e a densidade será

$$d = \frac{\omega_r}{n} = \frac{6}{12} = 0,5.$$

Os códigos LDPC regulares possuem estratégias utilizadas para a construção da matriz controle de paridade afim de reduzir a sua complexidade. O método que será apresentado, foi proposto por Gallager que utiliza a técnica da divisão da matriz em submatrizes [Gallager 1962]. Para realizar a construção segue-se os passos:

1. Dividir a matriz em ω_c submatrizes de ordem $m/\omega_c \times n$, onde cada uma dessas submatrizes tem peso da coluna igual à 1;

2. A primeira submatriz contém os 1's dispostos em cada linha de forma decrescente. Ou seja, a i-ésima linha contém 1's nas colunas $(i-1)\omega_r + 1$ até a $i\omega_r$;

3. As outras submatrizes são geradas através de permutações das colunas da primeira submatriz.

Para demonstrar como é realizada a construção, utiliza-se um exemplo para uma matriz de dimensões pequenas que segue a orientação dos passos descritos anteriormente.

Exemplo 3.2.2 *Dado um código LDPC com parâmetros (*12, 3, 6*), ou seja,* $n = 12$, $\omega_c = 3$ *e* $\omega_r = 6$, *temos*

$$R = 1 - \frac{\omega_c}{\omega_r} = 1 - \frac{3}{6} = \frac{1}{2} = \frac{k}{n}.$$

Assim, este código tem $k = 6$ *e* $m = n - k = 6$. *E,*

$$H = \left[\begin{array}{cccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}\right],$$

é a matriz controle de paridade deste código obtida a partir da estratégia de Gallager.

3.3 DECODIFICAÇÃO

Nesta seção, será apresentada uma estratégia de decodificação para códigos LDPC. Esse algoritmo recebe o nome de *bit-flipping*. Também proposto por Gallager, utiliza a decisão abrupta para encontrar a palavra-código final.

O algoritmo, consiste em utilizar o grafo de Tanner, visto no início do capítulo, e as equações de paridade da matriz H. Ele analisa as equações que falham e, entre elas, encontra o bit que mais se repete e troca o seu valor. O *bit-flipping* segue as seguintes etapas:

1. Seja um vetor **r** recebido, definir todas as equações de paridade;

2. Consultar para cada bit, o número de equações de paridade que falham;

3. O bit com maior quantidade de equações que falham, executar um *flip* na sua posição no vetor **r**, com as duas hipóteses $0 \to 1, 1 \to 0$;

4. Verificar se todas as equações de paridade são satisfeitas;

5. Caso não sejam satisfeitas, repetir todos os passos anteriores.

Para exemplificar o uso desse algoritmo, será mostrado as etapas descritas anteriormente para um exemplo pequeno com uma matriz controle de paridade de um código LDPC regular criada a partir da construção de Gallager.

Exemplo 3.3.1 *Seja a matriz controle de paridade dada em (3.2.2), com parâmetros* $(12, 3, 6)$. *Dado um vetor recebido* $\boldsymbol{r} = [0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0]$, *utilizar o decodificador bit-flipping para encontrar a palavra-código recebida. As equações de paridade dessa matriz serão*

$$\begin{cases} c_1 + c_2 + c_3 + c_4 + c_5 + c_6 = 0, \\ c_7 + c_8 + c_9 + c_{10} + c_{11} + c_{12} = 0, \\ c_1 + c_3 + c_5 + c_7 + c_9 + c_{11} = 0, \\ c_2 + c_4 + c_6 + c_8 + c_{10} + c_{12} = 0, \\ c_1 + c_2 + c_5 + c_6 + c_9 + c_{10} = 0, \\ c_3 + c_4 + c_7 + c_8 + c_{11} + c_{12} = 0. \end{cases}$$

A partir das equações, verifica-se qual bit que falha na maioria das equações. Observa-se que as equações que falham são

$$\begin{cases} c_1 + c_2 + c_3 + c_4 + c_5 + c_6 = 0, \\ c_2 + c_4 + c_6 + c_8 + c_{10} + c_{12} = 0, \\ c_3 + c_4 + c_7 + c_8 + c_{11} + c_{12} = 0. \end{cases}$$

Através dessas equações, é possível notar que o bit em comum entre todas as equações, o único que se repete, é o bit c_4. *O próximo passo é trocar o bit* $0 \to 1$ *e verificar se é uma palavra que satisfaça todas as equações de paridade. Então, a nova palavra-código será* $\boldsymbol{r} = [0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0]$. *Verifica-se que todas as equações são satisfeitas, portanto, o algoritmo para e essa será a palavra-código decodificada.*

4 CÓDIGOS LDPC EM 5G NR

Códigos LDPC possuem uma notável capacidade de correção de erros, o que o levou a ser utilizado em diversos padrões, como o IEEE 802.11 (Wifi) [Perahia e Stacey 2013] e o DVB-S2 (segunda geração de transmissão de vídeo via satélites) [EBU-UER 2009]. Para cada aplicação, diferentes técnicas de codificação e decodificação podem ser utilizadas. No 5G NR, estes códigos são utilizados para a codificação de canal de dados. A codificação neste caso é realizada através das chamadas matrizes base BG1 (*Base Graph 1*) e BG2 (*Base Graph 2*) [3GPP-138.212 2022], e os algoritmos de decodificação desejados são os de decisão suave por possuírem melhor desempenho que os de decisão abrupta.

A Seção 4.1 apresentará o que são as matrizes base BG1 e BG2, e como elas são utilizadas para a construção da matriz controle de paridade do código LDPC associado. Também será desenvolvido como são escritas e como são selecionadas estas matrizes de acordo com a taxa e o comprimento do bloco de informação. Na Seção 4.2 serão apresentadas as técnicas de decodificação. Inicia-se com a demonstração da razão logarítmica de verossimilhança (LLR) aplicada à um código de verificação de paridade único (SPC). A partir destes conceitos apresenta-se o algoritmo de decodificação Soma-Produto (SPA) baseado em tangentes hiperbólicas. Em sequência, descreve-se uma aproximação desse algoritmo para a redução do custo computacional chamado algoritmo Mínima-Soma (MS). E por fim, será descrito uma técnica que melhora a performance e reduz o tempo de processamento do algoritmo MS, que é o Mínima-Soma em Camadas (LMS). As simulações do Capítulo 5 serão realizadas a partir de códigos LDPC construídos a partir das matrizes bases BG1 e BG2, e o algoritmo de decodificação utilizado será o LMS.

4.1 MATRIZES BG1 E BG2

Os códigos LDPC para os padrões do 5G NR são construídos a partir de matrizes controle de paridade H, com dimensões $M \times N$, utilizando matrizes base H_b, com dimensões $M_b \times N_b$, juntamente com uma técnica de expansão por meio de um fator Z_c [3GPP-138.212 2022].

As matrizes bases utilizadas podem ser separadas em dois tipos: BG1 que é uma matriz com dimensão 46×68 e BG2 com dimensão 42×52. As duas matrizes tem algumas características em comum, como por exemplo, possuir uma estrutura de submatrizes em sua composição que permitem a identificação por blocos de estruturas, ou seja,

$$H_b = \begin{bmatrix} A & E & O \\ B & C & I \end{bmatrix}. \tag{1}$$

Cada uma das submatrizes apresenta diferentes dimensões, o que permite uma construção de duas matrizes controle de paridade finais de diferentes tamanhos. Para a matriz base BG1 tem-se

- A: 4×22, E: 4×4, O: 4×42 são matrizes nulas.

- B: 42×22, C: 42×4, I: 42×42 são matrizes identidades.

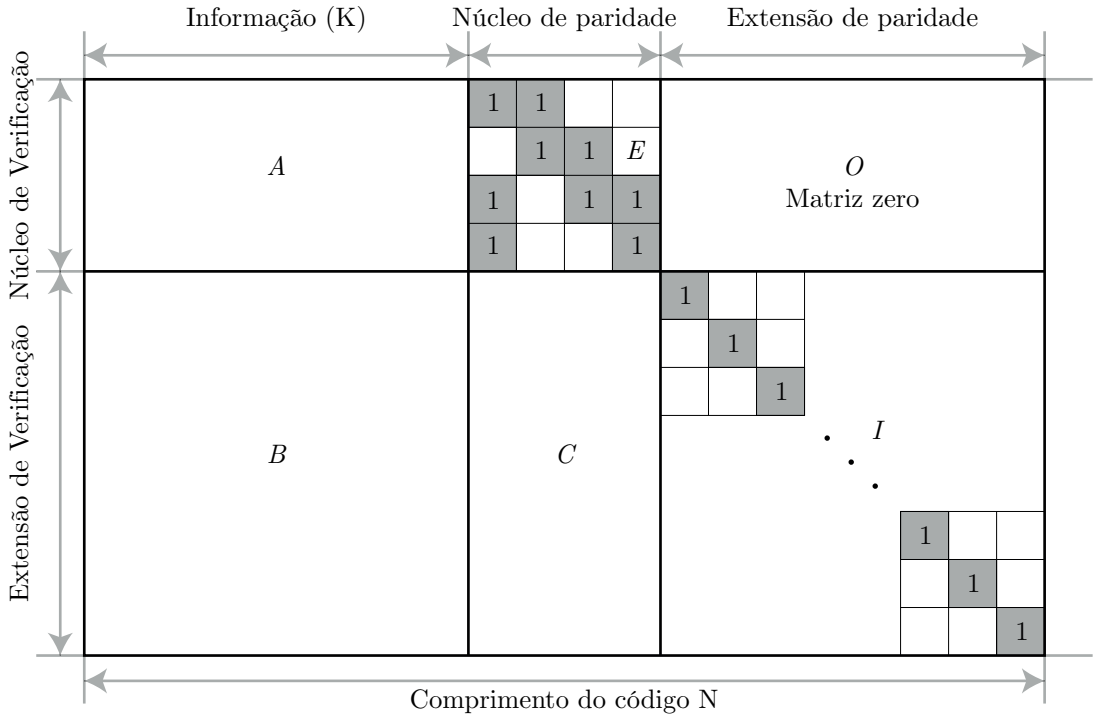
E, para a matriz base BG2 tem-se

- A: 4×10, E: 4×4, O: 4×38 são matrizes nulas.
- B: 38×10, C: 38×4, I: 38×38 são matrizes identidades.

Todos os elementos dessas submatrizes são baseados em construções de matrizes quadradas de tamanho Z_c nulas (como a submatriz O) ou identidades que podem, ou não, ser rotacionadas.

O diagrama da matriz base construída através desses blocos de estruturas no padrão 3GPP designado para o 5G NR, podem ser observados na Figura 7. Observa-se que parte dessa matriz considera os bits de informação, e a outra parte apresenta os bits de paridade e sua extensão.

Figura 7 – Estrutura da matriz base.



Fonte: Adaptado de [Nguyen, Tan e Lee 2019].

Em relação ao fator de expansão Z_c, parte importante na construção da matriz controle de paridade H, existem algumas variáveis importantes. A primeira é o índice i_{LS} que apresenta a ordem da expansão da matriz. Em sequência, têm-se um par de fatores a e J_a e o valor da expansão real Z_c. Esses valores, definem qual a matriz será utilizada para cada um dos casos que serão especificados nessa seção. São eles

- i_{LS} : $0, 1, 2, 3, 4, 5, 6, 7$.
- a : $2, 3, 5, 7, 9, 11, 13, 15$.
- J_a : $7, 7, 6, 5, 5, 5, 4, 4$.
- Z_c : $a \times 2^j, j = 0, 1, 2, ..., J_a$.

Para cada valor de i_{LS}, é representado um par a e J_a, que permite criar diferentes fatores de expansão, com limite em $Z_c = 384$. A Tabela 2 apresenta quais fatores de expansão são possíveis em cada índice.

Tabela 2 – Relação entre o índice e possíveis fatores de expansão para matrizes base.

Índice (i_{LS})	Fator de Expansão (Z_c)
0	$2, 4, 8, 16, 32, 64, 128, 256$
1	$3, 6, 12, 24, 48, 96, 192, 384$
2	$5, 10, 20, 40, 80, 160, 320$
3	$7, 14, 28, 56, 112, 224$
4	$9, 18, 36, 72, 144, 288$
5	$11, 22, 44, 88, 176, 352$
6	$13, 26, 52, 104, 208$
7	$15, 30, 60, 120, 240$

Fonte: Produção do Próprio Autor.

A construção da matriz controle de paridade final é dada seguindo alguns passos. Cada um dos elementos da matriz base H_b é substituído por uma matriz de dimensão $Z_c \times Z_c$. O valor dos elementos, variam de -1 a $Z_c - 1$, ou seja, para um fator de expansão de 384, o valor varia de -1 a 383. Para cada um desses valores, segue-se as seguintes regras:

- Os elementos de valor -1 são substituídos por matrizes nula de dimensão $Z_c \times Z_c$.
- Os elementos de valor 0 são substituídos por matrizes identidade de dimensão $Z_c \times Z_c$.
- Os elementos diferentes de -1 e 0, são substituídos por matrizes identidade com uma permutação circular de acordo com o número do elemento de dimensão $Z_c \times Z_c$.

Essa construção, permite a redução da memória requerida para implementar um código decodificador, visto que as matrizes analisadas serão de tamanhos muito menores que a matriz controle de paridade completa. Devido as altas dimensões das matrizes base BG1 e BG2, para exemplificar a construção de uma matriz H obtida a partir das regras acima citadas, será apresentado a seguir um exemplo a partir de uma matriz base H_b e um fator de expansão Z_c quaisquer.

Exemplo 4.1.1 *Considere a matriz base H_b com dimensão 3×5 e fator de expansão $Z_c = 4$*

$$H_b = \begin{bmatrix} 2 & 1 & 3 & -1 & 0 \\ 3 & 0 & 2 & -1 & 0 \\ 1 & 3 & 1 & 2 & -1 \end{bmatrix}.$$

De acordo com as regras acima citadas, os números -1's são substituídos por matrizes nula, os números 0's são substituídos por matrizes identidade, e os números diferentes de -1 e 0 são substituídos por matrizes identidade com permutações circulares. Observa-se que as cores dos elementos da matriz base, são substituídos por matrizes 4×4 de acordo com as regras descritas anteriormente. Por exemplo, para o elemento 2 a matriz identidade sofre uma permutação à direita de duas colunas. Da mesma

forma ocorre para os numerais 1 *e* 3*, que permutam uma coluna e três colunas, respectivamente.*

$$H = \left[\begin{array}{cccccccccccccccccccc}
0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0\\
0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1\\
0&0&0&1&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&1\\
0&1&0&0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0\\
0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&0&1&0&1&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
1&0&0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0&0
\end{array}\right].$$

No caso do padrão 5G NR, para selecionar qual matriz base utilizar, segue-se a documentação do 3GPP que define as características e necessidades de cada uma das matrizes [3GPP-138.212 2022]. A matriz que será utilizada para a construção da matriz controle de paridade é definida através do tamanho do bloco a ser enviado e da taxa do código. Seja um bloco de mensagem com comprimento K e uma taxa $R = K/N$. Para o uso da matriz BG2, tem-se:

- $K \leq 3824$ e $R \leq 0,67$, ou
- $K \leq 292$, ou
- $R \leq 0,25$.

Caso nenhuma das condições seja satisfeita, então será selecionada a matriz BG1. As definições dos limites para uso de cada uma podem ser observados na Figura 8. Os blocos tem um comprimento máximo permitido a ser enviado. Para a matriz base BG1, o valor máximo do comprimento do bloco é 8448, e para a matriz base BG2, o valor máximo do comprimento do bloco é 3840.

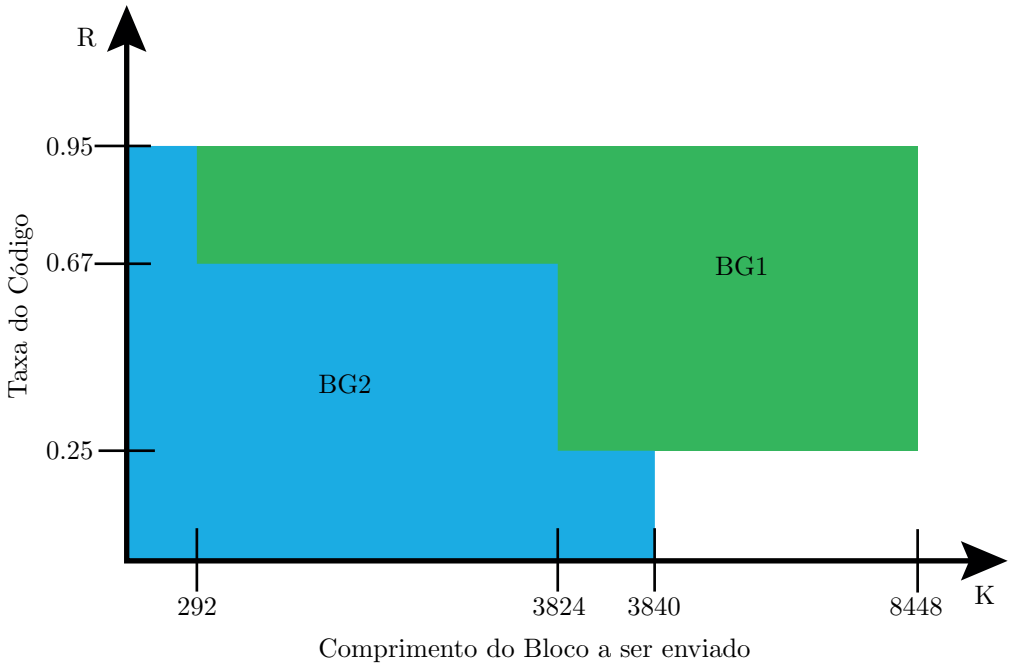
A próxima etapa é definir o número de bits K_b que denota o número de colunas de bits de informação para expansão Z_c, do bloco a ser enviado, de acordo com o tamanho K e da taxa R. São especificados para a matriz BG1:

- $K_b = 22$,

e para a matriz BG2:

- $K > 640 \rightarrow K_b = 10$,
- $560 < K \leq 640 \rightarrow K_b = 9$,
- $192 < K \leq 560 \rightarrow K_b = 8$,
- $K \leq 192 \rightarrow K_b = 6$.

Figura 8 – Seleção da Matriz Base para o 5G NR.



Fonte: Adaptado de [Chelikani 2019].

Em sequência, é encontrado o valor mínimo Z_c que cumpre o requisito de $K_b \times Z_c \geq K$, através da Tabela 2. Consequentemente, será encontrado o índice i_{LS} correspondente. Assim, será selecionada a matriz utilizada para aquela taxa de código R e comprimento da informação K desejada. Para a construção das matrizes é utilizado o valor do comprimento K e o menor fator de expansão Z_c que satisfaça $K_b Z_c \geq K$.

Para exemplificar, considere $K = 20$, ou seja, neste caso utiliza-se a matriz BG2. Logo, K_b será 6 e o menor fator de expansão possível será $Z_c = 4$. Esse valor, representa a ordem de expansão $i_{LS} = 0$. A equação que determina cada elemento da matriz base é dada por

$$P(i, j) = mod(V_{i,j}, Z_c), \tag{2}$$

onde os valores de $V_{i,j}$, são definidos na Tabela 5.3.2-3 da documentação de especificação da codificação para o 5G NR [3GPP-138.212 2022]. Um recorte dessa tabela para a construção da primeira linha da matriz base, pode ser observado na Figura 9.

Para a primeira linha e as cinco primeiras colunas, são executados os seguintes passos:

- $V_{0,0} = 9$, $P(i, j) = mod(9, 4) = 1$.
- $V_{0,1} = 117$, $P(i, j) = mod(117, 4) = 1$.
- $V_{0,2} = 204$, $P(i, j) = mod(204, 4) = 0$.
- $V_{0,3} = 26$, $P(i, j) = mod(26, 4) = 2$.
- $V_{0,4} =$ (não consta na tabela), $P(i, j) = -1$.

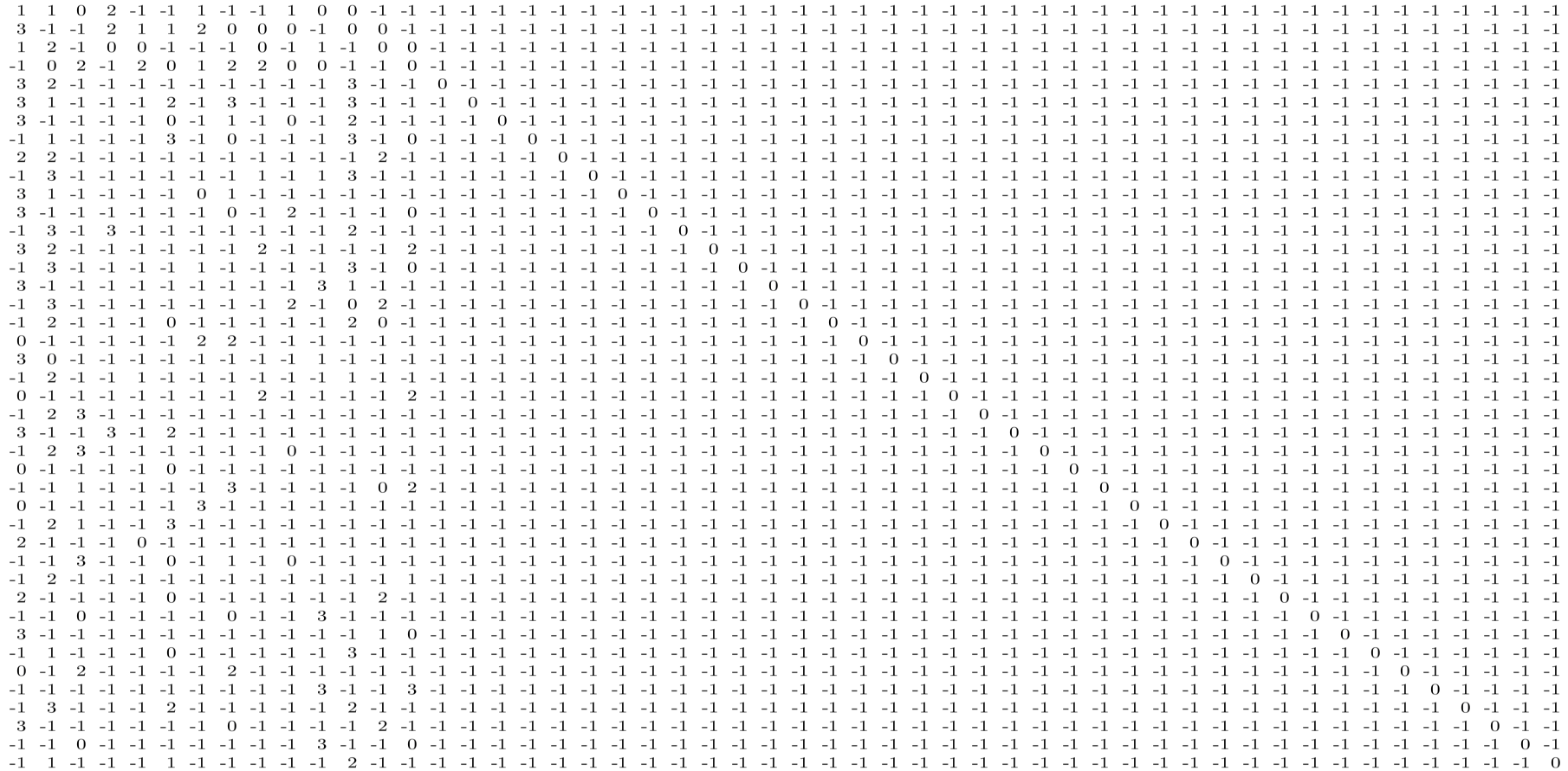
Figura 9 – Recorte da Tabela 5.3.2-3 para construção da matriz base.

$\mathbf{H}_{\mathrm{BG}}$		$V_{i,j}$							
Row index i	Column index j	Set index i_{LS}							
		0	1	2	3	4	5	6	7
0	0	9	174	0	72	3	156	143	145
	1	117	97	0	110	26	143	19	131
	2	204	166	0	23	53	14	176	71
	3	26	66	0	181	35	3	165	21
	6	189	71	0	95	115	40	196	23
	9	205	172	0	8	127	123	13	112
	10	0	0	0	1	0	0	0	1
	11	0	0	0	0	0	0	0	0

Fonte: [3GPP-138.212 2022].

A matriz BG2 tem 42 linhas e 52 colunas, ou seja, os elementos vão até $V_{41,51}$ de acordo com os passos realizados anteriormente. Para as outras colunas e linhas, deve-se consultar a tabela completa e seguir o algoritmo. Assim, encontra-se finalmente a matriz completa apresentada na Figura 10.

Figura 10 – Matriz BG2 com $K = 20$ construída no padrão 5G NR.



Fonte: Produção do próprio autor.

4.2 DECODIFICAÇÃO

Nesta seção serão descritos algoritmos de decodificação por decisão suave que podem ser aplicados aos códigos LDPC nos requisitos do 5G NR. Na Subseção 4.2.1 será introduzida a Razão Logarítmica de Verossimilhança (LLR, *Log-likelihood Ratio*) intrínseca, utilizada para realizar a decodificação baseada em estimativas de probabilidade. A Subseção 4.2.2 apresentará a LLR extrínseca com a exemplificação por meio de um código de verificação de paridade única (SPC, *Single Parity Check*). O algoritmo Soma-Produto (SPA) será apresentado na Subseção 4.2.3, o Mínima-Soma (MS) na Subseção 4.2.4, e o Mínima-Soma em Camadas (LMS) na Subseção 4.2.5. Todos os algoritmos serão apresentados e descritos passo a passo e exemplificados com pequenas matrizes.

4.2.1 Razão Logarítmica de Verossimilhança

Para realizar a decodificação de códigos LDPC existe um conceito importante para a análise dos algoritmos utilizados para decodificações por decisões suaves (*soft-decision*). Esse método é baseado em estimativas de probabilidades de cada um dos bits enviados, bem como a relação que cada um possui com todos os outros. Esse conceito recebe o nome de Razão Logarítmica de Verossimilhança (LLR, *Log-likelihood Ratio*) [Hagenauer, Offer e Papke 1996].

Dado uma mensagem binária de comprimento n, codificada através de uma estratégia qualquer, modulada em BPSK, onde $0 \rightarrow +1$ e $1 \rightarrow -1$, e transmitida por um canal gaussiano, o vetor $\mathbf{r}$ de valores recebidos no decodificador se torna

$$\mathbf{r} = [r_1, r_2, \ldots, r_n]. \tag{3}$$

A decodificação baseada em *soft-decision* recebe o vetor e transforma-o em um conjunto de valores de estimativas de probabilidades representadas pelo vetor $\mathbf{L}$ também de comprimento n,

$$\mathbf{L} = [L_1, L_2, \ldots, L_n]. \tag{4}$$

Cada um dos valores L_i é formado pela soma das estimativas intrínsecas (relacionada com o valor real recebido daquele símbolo) e extrínsecas (relacionada ao que os outros valores representam para o símbolo em particular).

A probabilidade de um bit ser recebido ser 0, baseado no valor real visto na entrada do decodificador, é descrito como $Pr(c_i = 0|r_i)$. Para estruturação do conceito da Razão Logarítmica de Verossimilhança, utiliza-se o Teorema de Bayes dado por

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{5}$$

assim, a probabilidade do bit c_i ser igual à 0 dado o valor real r_i, será

$$Pr(c_i = 0|r_i) = \frac{f(r_i|c_i = 0)Pr(c_i = 0)}{f(r_i)}. \tag{6}$$

A probabilidade $Pr(c_i = 0)$ recebe o nome de probabilidade *a priori*. Como a mensagem codificada é binária, existe apenas dois símbolos equiprováveis (0 ou 1). Dessa forma, a probabilidade *a priori* pode ser substituída por $\frac{1}{2}$ para definição dos cálculos da LLR. Com o mesmo raciocínio, também pode-se escrever a probabilidade do bit c_i ser igual à 1

$$Pr(c_i = 1|r_i) = \frac{f(r_i|c_i = 1)Pr(c_i = 1)}{f(r_i)}. \tag{7}$$

Para encontrar a Razão de Verossimilhança, basta fazer a razão entre (6) e (7)

$$\frac{Pr(c_i = 0|r_i)}{Pr(c_i = 1|r_i)} = \frac{f(r_i|c_i = 0)}{f(r_i|c_i = 1)}. \tag{8}$$

Em uma palavra-código, modulada em BPSK, transmitida através de um canal gaussiano, o bit recebido na entrada do decodificador é no formato

$$\begin{aligned} c_i = 0 \rightarrow r_i = +1 + N(0, \sigma^2), \\ c_i = 1 \rightarrow r_i = -1 + N(0, \sigma^2). \end{aligned} \tag{9}$$

A Equação 8 pode ser reescrita substituindo as relações descritas em (9)

$$\frac{Pr(c_i = 0|r_i)}{Pr(c_i = 1|r_i)} = \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(r_i-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(r_i+1)^2}{2\sigma^2}}} = e^{\frac{2r_i}{\sigma^2}}, \tag{10}$$

ao aplicar a função logarítmica natural na Equação 10, tem-se a Razão Logarítmica de Verossimilhança Intrínseca

$$l_i = \ln \frac{Pr(c_i = 0|r_i)}{Pr(c_i = 1|r_i)} = \frac{2}{\sigma^2} r_i. \tag{11}$$

A LLR intrínseca, é acompanhada do número real recebido r_i com adição de uma constante $\frac{2}{\sigma^2}$ que depende da variância do ruído do canal gaussiano. Porém, como é uma constante positiva, usualmente é ignorado esse fator.

A LLR extrínseca será descrita com um exemplo do Código de Repetição na próxima subseção. A partir disso, pode-se generalizar para códigos com mais de um bit de paridade como são as construções dos códigos de verificação de paridade de baixa densidade (LDPC).

4.2.2 Código de Verificação de Paridade Única

Para exemplificar a LLR, nesta subseção será analisado o código de verificação de paridade única (SPC, *Single Parity Check*), com a definição do que é a estimativa extrínseca. Para melhor identificação dos conceitos, será exibido o código SPC $(3, 2)$ [Hasan 2018].

Dado um código de bloco linear (n, k) com $k = n - 1$, a mensagem enviada

$$\mathbf{m} = [m_1 \; m_2 \; \ldots \; m_{n-1}] \tag{12}$$

é codificada em uma palavra-código composta por todos os $n - 1$ bits da mensagem com a adição de

um único bit de paridade

$$\mathbf{c} = [m_1\ m_2\ \ldots\ m_{n-1}\ p]. \tag{13}$$

Esse bit é dado pela soma binária (ou exclusiva) de todos os bits da mensagem anterior, ou seja,

$$p = m_1 \oplus m_2 \oplus \ldots \oplus m_{n-1}. \tag{14}$$

A sua matriz geradora tem dimensões $n-1 \times n$. Ela possuirá uma parte com uma matriz identidade de tamanho $n-1$ e uma última coluna na posição n de 1's

$$G = \left[\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 1 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & 1 & 0 & \ldots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 & 1 \end{array}\right]. \tag{15}$$

A matriz controle de paridade a partir dessa matriz geradora tem dimensoes $1 \times n$

$$H = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \end{bmatrix}. \tag{16}$$

Ao observar o código de verificação de paridade única com parâmetros $(3, 2)$, a mensagem binária a ser enviada tem comprimento dois. Ou seja, ela possui 4 combinações possíveis. São elas

$$\mathbf{m} = [00, 01, 10, 11]. \tag{17}$$

Ao ser codificada, baseada em (14), a palavra-código se torna de comprimento três, sendo o último bit o bit de paridade

$$\mathbf{c} = [000, 011, 101, 110]. \tag{18}$$

Considera-se uma das mensagens (18) enviada através de um canal gaussiano e modulada em BPSK. Na entrada do decodificador, é recebido um vetor de números reais representados por

$$\mathbf{r} = [r_1\ r_2\ r_3], \tag{19}$$

com $r_i = c_i + N(0, \sigma^2)$, e o valor do ruído é independete para cada um dos bits. Esse vetor **r** pode ser apresentado como um vetor de estimativa de probabilidades

$$\mathbf{L} = [L_1\ L_2\ L_3], \tag{20}$$

onde $L_i = L_i^i + L_i^e$, represente a soma das estimativas de probabilidades intrínsecas e extrínsecas à cada bit. Para o caso de L_1, a informação intrínseca é proporcional à r_1 e a extrínseca apresenta a relação que r_2 e r_3 tem com c_1. Será apresentada a LLR para o bit codificado c_1.

Para simplificação dos cálculos, toma-se $Pr(c_i = 0|r_i) = p_i$ e $Pr(c_i = 1|r_i) = 1 - p_i$. Para $c_1 = 0$, têm-se as possibilidade: $c_2 = c_3 = 0$, ou $c_2 = c_3 = 1$. E para $c_1 = 1$, as possibilidades são: $c_2 = 0$ e

$c_3 = 1$, ou $c_2 = 1$ e $c_3 = 0$. Assim,

$$p_1 = p_2 p_3 + (1 - p_2)(1 - p_3), \tag{21}$$

$$1 - p_1 = p_2(1 - p_3) + (1 - p_2)p_3. \tag{22}$$

A diferença entre as probabilidades (21) e (22) traz a relação

$$p_1 - (1 - p_1) = p_2(p_3 - (1 - p_3) + (1 - p_2)((1 - p_3) - p_3), \tag{23}$$

$$p_1 - (1 - p_1) = (p_2 - (1 - p_2))(p_3 - (1 - p_3). \tag{24}$$

Ao dividir relação obtida em (24), por $1 = p_i + (1 - p_i)$, para fins de simplificação, tem-se

$$\frac{p_1 - (1 - p_1)}{p_1 + (1 - p_1)} = \frac{(p_2 - (1 - p_2))}{(p_2 + (1 - p_2))} \frac{(p_3 - (1 - p_3)}{(p_3 + (1 - p_3)}. \tag{25}$$

Novamente, ao dividir cada termo por p_i

$$\frac{1 - (\frac{1-p_1}{p_1})}{1 + (\frac{1-p_1}{p_1})} = \frac{1 - (\frac{1-p_2}{p_2})}{1 + (\frac{1-p_2}{p_2})} \frac{1 - (\frac{1-p_3}{p_3})}{1 + (\frac{1-p_3}{p_3})}. \tag{26}$$

Para encontrar a estimativa extrínseca l_1, basta utilizar a mesma razão utilizada em (11). Porém, a probabilidade de $c_i = 0$ ou $c_i = 1$ é dada com a observação dos números reais recebidos r_2 e r_3

$$l_1 = \ln \frac{Pr(c_1 = 0|r_2, r_3)}{Pr(c_1 = 1|r_2, r_3)} = \ln \frac{p_1}{1 - p_1}. \tag{27}$$

Como $e^{-\ln(x)} = \frac{1}{x}$, substitui-se (27), em (26)

$$\frac{1 - e^{-l_1^e}}{1 + e^{-l_1^e}} = \frac{1 - e^{-l_2}}{1 + e^{-l_2}} \frac{1 - e^{-l_3}}{1 + e^{-l_3}}. \tag{28}$$

A Equação 28 se relaciona com a tangente hiperbólica que pode ser escrita no formato exponencial

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}. \tag{29}$$

A relação entre as estimativas de probabilidades que c_2 e c_3 dizem sobre c_1 pode ser escrita baseada na regra da tangente hiperbólicas

$$c_1 = c_2 \oplus c_3 \rightarrow \tanh \frac{l_1^e}{2} = \tanh \frac{l_2}{2} \tanh \frac{l_3}{2}. \tag{30}$$

Pode-se isolar o valor da LLR extrínseca do bit c_1, assim encontra-se

$$l_1^e = 2 \tanh^{-1} \left(\tanh \frac{l_2}{2} \tanh \frac{l_3}{2} \right). \tag{31}$$

Para um caso geral, a Equação 31 é definida por

$$l_1^e = 2\tanh^{-1}\left(\prod_{j\neq k}\tanh\frac{l_i}{2}\right). \tag{32}$$

A tangente hiperbólica é uma função ímpar. Isto é,

$$\begin{aligned} x < 0 &\rightarrow \tanh(x) < 0, \\ x > 0 &\rightarrow \tanh(x) > 0. \end{aligned} \tag{33}$$

A partir disso, pode-se dividir a Equação 30 em duas partes. A primeira apresenta a função sinal de cada um dos termos

$$sign(l_1^e) = sign(l_2)sign(l_3), \tag{34}$$

e a segunda que apresenta a tangente hiperbólica dos valores absolutos dos termos

$$\left|\tanh\frac{|l_1^e|}{2}\right| = \left|\tanh\frac{|l_2|}{2}\right|\left|\tanh\frac{|l_3|}{2}\right|, \tag{35}$$

ao aplicar o logaritmo natural, a função se torna

$$\left|\ln\left(\tanh\frac{|l_1^e|}{2}\right)\right| = \left|\ln\left(\tanh\frac{|l_2|}{2}\right)\right| + \left|\ln\left(\tanh\frac{|l_3|}{2}\right)\right|. \tag{36}$$

E, finalmente, têm-se a LLR referente ao bit codificado c_1, dada por

$$L_1 = r_1 + sign(l_2)sign(l_3) \times 2 \times \tanh^{-1}\ln^{-1}\left(\left|\ln\left(\tanh\frac{|r_2|}{2}\right)\right| + \left|\ln\left(\tanh\frac{|r_3|}{2}\right)\right|\right). \tag{37}$$

4.2.3 Algoritmo Soma-Produto

Para a descrição do algoritmo soma-produto baseado em tangente hiperbólica, segue-se a notação utilizada na literatura por [MacKay 1999] e [Fossorier, Mihaljevic e Imai 1999]. A construção do algoritmo será baseada na LLR descrita na Subseção 4.2.2.

Dado um código LDPC binário (N, K) com uma matriz controle de paridade esparsa de dimensões $M \times N$, e $M = N - K$. O seu grafo de Tanner é dado por nós de variável n e nós de paridade m. O conjunto dos nós de variável n conectados à um nó de paridade m é dado por $\mathcal{N}(m)$, e $\mathcal{N}(m)\backslash n$ é esse conjunto com exceção de n. O conjuntos dos nós de paridade m conectados à um nó de variável n é dado por $\mathcal{M}(n)$, e $\mathcal{M}(n)\backslash m$ é esse conjunto com exceção de m.

Cada iteração, o algoritmo apresenta o passo horizontal (ou atualização do nó de paridade) que denota a mensagem $r_{m\rightarrow n}$ que o nó de paridade m envia para os nós de variável n, se $n \in \mathcal{N}(m)\backslash n$, com a probabilidade do símbolo ser 0 ou 1. A segunda etapa, é o passo vertical (ou atualização do nó de variável) que apresenta a mensagem $q_{n\rightarrow m}$ que o nó de variável n envia para os nós de paridade m, se $m \in \mathcal{M}(n)\backslash m$, com a probabilidade do símbolo ser 0 ou 1. Os vetores das palavras-código da decodificação são representados por $\mathbf{y} = [y_1, \ldots, y_N]$ para as palavras recebidas e $\mathbf{u} = [u_1, \ldots, u_N]$ para as palavras transmitidas. A seguir, apresenta-se de forma detalhada as etapas do algoritmo de

decodificação SPA.

1. **Inicialização**

   O valor inicial das LLR são obtidos através do vetor de números reais recebido $\mathbf{y}$. Esses valores são utilizados como $q_{n\to m}$. São as primeiras informações enviadas dos nós de variável para os nós de paridade.

2. **Passo Horizontal**

   O passo horizontal processa as mensagens recebidas $q_{n\to m}$ para calcular as mensagem enviadas para os nós de variável $r_{m\to n}$. É utilizada a LLR vista na Seção 4.2.2

$$r_{m\to n} = \prod_{n'\in\mathcal{N}(m)\setminus n} sign(q_{n'\to m}) \times 2\tanh^{-1} \prod_{n'\in\mathcal{N}(m)\setminus n} \tanh\left(\frac{|q_{n'\to m}|}{2}\right), \tag{38}$$

   que também pode ser escrita no formato de logaritmo,

$$r_{m\to n} = \prod_{n'\in\mathcal{N}(m)\setminus n} sign(q_{n'\to m}) \times 2\tanh^{-1}\ln^{-1} \sum_{n'\in\mathcal{N}(m)\setminus n} \left|\ln\left(\tanh\frac{|q_{n'\to m}|}{2}\right)\right|. \tag{39}$$

3. **Passo Vertical**

   O passo vertical processa as mensagem recebidas $r_{m\to n}$ para calcular as mensagens enviadas para os nós de paridade $q_{n\to m}$

$$q_{n\to m} = \sum_{m'\in\mathcal{M}(n)\setminus m} r_{m'\to n}(u_n) + L_n. \tag{40}$$

4. **Passo Decisão**

   O valor das LLRs são calculados com base na mensagem enviada $r_{m\to n}$ e no valor do vetor recebido

$$L_n = \sum_{m\in\mathcal{M}(n)} r_{m\to n} + L_n. \tag{41}$$

   Para a modulação BPSK, caso o elemento $L_n \geq 0$ é definido o bit 0. Se não, o bit 1 será definido.

Para um melhor entendimento do algoritmo, será exemplificado o passo a passo da decodificação para uma matriz controle de paridade pequena.

Exemplo 4.2.1 *Considere um código LDPC com a seguinte matriz controle de paridade*

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

A decodificação do vetor de números reais $\boldsymbol{y} = [1,53 \;\; 1,23 \;\; 0,64 \;\; 0,95 \;\; -0,45 \;\; 1,11 \;\; 0,53]$ *recebido utilizando o algoritmo SPA será dada da seguinte forma:*

1. ***Inicialização***

   O valor inicial das LLR *são obtidos através do vetor de números reais recebido* $\boldsymbol{y}$

   $$\boldsymbol{L} = \boldsymbol{y}.$$

   Esses valores são utilizados como $q_{n\to m}$

   $$Q = \begin{bmatrix} 1,53 & 1,23 & 0,64 & 0 & -0,45 & 0 & 0 \\ 0 & 0 & 0 & 0,95 & 0 & 1,11 & 0,53 \\ 1,53 & 1,23 & 0 & 0,95 & 0 & 0 & 0,53 \\ 0 & 0 & 0,64 & 0 & -0,45 & 1,11 & 0 \end{bmatrix}.$$

2. ***Passo Horizontal 1ª iteração***

   O passo horizontal tem como objetivo atualizar a matriz R *com base na Equação 38*

   $$r_{11} = (+)(+)(-)\times 2\tanh^{-1}\left(\tanh\left(\frac{|1,23|}{2}\right)\tanh\left(\frac{|0,64|}{2}\right)\tanh\left(\frac{|-0,45|}{2}\right)\right) = -0,0750.$$

   Para todos outros valores de r_{mn}*, têm-se*

   $$R = \begin{bmatrix} -0,0750 & -0,0883 & -0,1564 & 0 & 0,2192 & 0 & 0 \\ 0 & 0 & 0 & 0,2627 & 0 & 0,2301 & 0,4536 \\ 0,1256 & 0,1478 & 0 & 0,1832 & 0 & 0 & 0,3145 \\ 0 & 0 & -0,2241 & 0 & 0,3147 & -0,1372 & 0 \end{bmatrix}.$$

   Também é possível atualizar essa matriz através da Equação 39.

3. ***Passo Vertical 1ª iteração***

   O passo vertical atualiza a matriz Q *baseado na Equação 40*

   $$Q = \begin{bmatrix} 1,6556 & 1,3778 & 0,4159 & 0 & -0,2308 & 0 & 0 \\ 0 & 0 & 0 & 1,1332 & 0 & 0,9728 & 0,8445 \\ 1,4550 & 1,1417 & 0 & 1,2127 & 0 & 0 & 0,9836 \\ 0 & 0 & 0,4836 & 0 & -0,2308 & 1,3401 & 0 \end{bmatrix}.$$

4. ***Decisão 1ª iteração***

   O vetor com as novas LLR é atualizado baseado em (41)

   $$\boldsymbol{L} = [1,5806 \;\; 1,2895 \;\; 0,2595 \;\; 1,3959 \;\; 0,0839 \;\; 1,2029 \;\; 1,2981].$$

Os bits são definidos através de

$$L_n > 0 \rightarrow c_n = 0,$$
$$L_n < 0 \rightarrow c_n = 1.$$

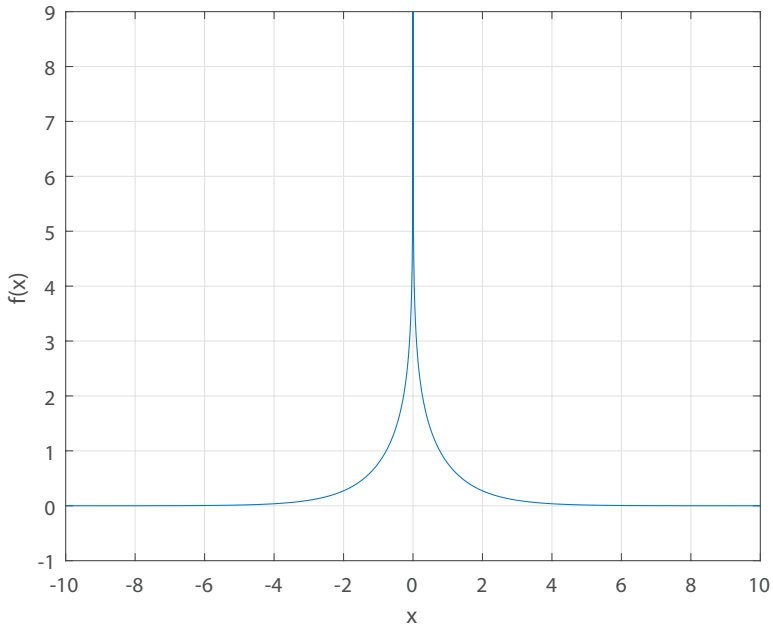
Logo, a palavra-código $c = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$ *foi encontrada na primeira iteração.*

4.2.4 Algoritmo Mínima-Soma

O custo computacional para processar grandes matrizes utilizando tangentes hiperbólicas é elevado. Com base nisso, foi estudada uma aproximação para reduzir esse custo. Porém, essa troca é realizada de forma que, possibilitar o aumento do número de iterações, acarreta em uma menor qualidade do código [Tu e Zhang 2007, Emran e Elsabrouty 2014].

A relação vista na Equação 36, através da construção do seu gráfico visto na Figura 11, é possível observar que $f(x) = \left|\ln\left(\tanh\frac{|x|}{2}\right)\right|$, é uma função íngreme. Isso significa que para valores de $|x| \rightarrow \infty$, $f(x) \rightarrow 0$, e para valores de $|x| \rightarrow 0$, $f(x) \rightarrow \infty$.

Figura 11 – Gráfico da função $f(x) = \left|\ln\left(\tanh\frac{|x|}{2}\right)\right|$.



Fonte: Produção do Próprio Autor.

Como consequência disso, os termos que apresentam os maiores valores podem ser desprezados no somatório visto em (39). Assim, a Equação 36 pode ser reescrita como

$$f(l_1^e) = f(l_2) + f(l_3) = f(min(|l_2|, |l_3|), \tag{42}$$

e pode ser simplificada para obter o valor da LLR extrínseca ao bit c_1

$$|l_1^e| = f^{-1}(f(min(|l_2|, |l_3|)). \tag{43}$$

Ou seja, para um caso geral do passo horizontal

$$|l_n^e| = min(|l_1|, |l_2|, \ldots, |l_{n-1}|). \tag{44}$$

Ao retornar para o algoritmo, a diferença entre o mínima-soma e o soma-produto é apenas no passo horizontal. A Equação 39 com tangentes hiperbólicas é substituída pela simplificação do mínimo valor

$$r_{m\to n} = \prod_{n'\in\mathcal{N}(m)\setminus n} sign(q_{n'\to m}) \times \min_{n'\in\mathcal{N}(m)\setminus n}(|q_{n'\to m}|). \tag{45}$$

Devido a semelhança entre os dois algoritmos, o passo a passo se repete com exceção do passo horizontal que será alterada a equação de atualização da matriz R. Com isso, será apresentado o algoritmo mínima-soma diretamente através do exemplo a seguir.

Exemplo 4.2.2 *Considere o código LDPC dado no Exemplo 4.2.1 e o mesmo vetor de números reais recebido. Vamos agora, aplicar o algoritmo MS:*

1. ***Inicialização***

   O valor inicial das LLR são obtidos através do vetor de números reais recebido $\boldsymbol{y}$

   $$\boldsymbol{L} = \boldsymbol{y}.$$

   Esses valores são utilizados como $q_{n\to m}$

   $$Q = \begin{bmatrix} 1,53 & 1,23 & 0,64 & 0 & -0,45 & 0 & 0 \\ 0 & 0 & 0 & 0,95 & 0 & 1,11 & 0,53 \\ 1,53 & 1,23 & 0 & 0,95 & 0 & 0 & 0,53 \\ 0 & 0 & 0,64 & 0 & -0,45 & 1,11 & 0 \end{bmatrix}.$$

2. ***Passo Horizontal 1ª iteração***

   O passo horizontal tem como objetivo atualizar a matriz R com base na Equação 45

   $$r_{11} = (+)(+)(-) \times |-0,45| = -0,45.$$

   Observa-se que a atualização da matriz R ocorre de maneira mais simples que o algoritmo SPA. Para todos outros valores de r_{mn}, têm-se

   $$R = \begin{bmatrix} -0,45 & -0,45 & -0,45 & 0 & 0,64 & 0 & 0 \\ 0 & 0 & 0 & 0,53 & 0 & 0,53 & 0,95 \\ 0,53 & 0,53 & 0 & 0,53 & 0 & 0 & 0,95 \\ 0 & 0 & -0,45 & 0 & 0,64 & -0,45 & 0 \end{bmatrix}.$$

3. ***Passo Vertical 1ª iteração***

Atualização da matriz Q baseado na Equação 40

$$Q = \begin{bmatrix} 2,06 & 1,76 & 0,19 & 0 & 0,19 & 0 & 0 \\ 0 & 0 & 0 & 1,48 & 0 & 0,66 & 1,48 \\ 1,08 & 0,78 & 0 & 1,48 & 0 & 0 & 1,48 \\ 0 & 0 & 0,19 & 0 & 0,19 & 1,64 & 0 \end{bmatrix}.$$

4. ***Decisão 1ª iteração***

   O vetor com as novas LLR é atualizado baseado em (41)

$$\boldsymbol{L} = [1,61 \;\; 1,31 \;\; -0,26 \;\; 2,01 \;\; 0,83 \;\; 1,19 \;\; 2,43].$$

   Os bits são definidos através de

$$L_n > 0 \rightarrow c_n = 0,$$
$$L_n < 0 \rightarrow c_n = 1.$$

   Logo, o vetor encontrado na primeira etapa $[0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$ *não é uma palavra-código válida. Como não foi detectada, o algoritmo passa por mais um processo de iteração com inicio no passo horizontal, utilizando a última matriz Q obtida da iteração 3.*

5. ***Passo Horizontal 2ª iteração***

   Atualização da matriz R

$$r_{11} = (+)(+)(+) \times |0,19| = 0,19.$$

   Novamente, nota-se a simplificação da atualização da matriz R. Para todos outros valores de r_{mn}, têm-se

$$R = \begin{bmatrix} 0,19 & 0,19 & 0,19 & 0 & 0,19 & 0 & 0 \\ 0 & 0 & 0 & 0,66 & 0 & 1,48 & 0,66 \\ 0,78 & 1,08 & 0 & 0,78 & 0 & 0 & 0,78 \\ 0 & 0 & 0,19 & 0 & 0,19 & 0,19 & 0 \end{bmatrix}.$$

6. ***Passo Vertical 2ª iteração***

   Atualização da matriz Q

$$Q = \begin{bmatrix} 2,39 & 2,39 & -0,07 & 0 & 1,02 & 0 & 0 \\ 0 & 0 & 0 & 2,79 & 0 & 1,38 & 3,21 \\ 1,80 & 1,50 & 0 & 2,67 & 0 & 0 & 3,09 \\ 0 & 0 & -0,07 & 0 & 1,02 & 1,38 & 0 \end{bmatrix}.$$

7. ***Decisão 2ª iteração***

O vetor com as novas LLR é atualizado baseado em (41)

$$\boldsymbol{L} = [2,58 \;\; 2,58 \;\; 0,12 \;\; 3,45 \;\; 1,21 \;\; 2,86 \;\; 3,87].$$

Os bits são definidos através de

$$L_n > 0 \rightarrow c_n = 0,$$
$$L_n < 0 \rightarrow c_n = 1.$$

Logo, a palavra-código $c = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$ *foi encontrada na segunda iteração.*

4.2.5 Decodificação em Camadas

Os algoritmos SPA e MS possuem uma outra abordagem que se baseia em uma decodificação realizada por camadas. Neste caso, a matriz controle de paridade é dividida em camadas horizontais, onde o processo iterativo de decodificação é aplicado sequencialmente. Cada uma dessas iterações nas camadas, são chamadas de subiterações e, ao finalizar o processo das subiterações, ocorre a superiteração. Esta estratégia de decodificação por camadas apresenta a mesma performance dos algoritmos, porém com uma convergência mais rápida [Hocevar 2004].

Cada uma das camadas, deve possuir o peso $\omega_c = 1$, ou seja, somente um 1 em cada coluna. Isso significa que em cada camada só ocorrerá um processo de subiteração. Essa característica da abordagem de decodificação em camadas é compatível com a decodificação de código para o 5G NR. A construção da matriz base, é composta por n linhas, e cada linha apresenta m elementos que são expandidos em matrizes identidades (ou nulas) com permutações circulares, o que resulta em m camadas com peso unitário nas colunas, como visto no Capítulo 4, Seção 4.1.

Os passos para a execução do algoritmo realizado em camadas que será apresentado a seguir, se baseiam nos passos do algoritmo MS. A diferença neste caso é que a matriz controle de paridade é dividida em várias submatrizes. Tal algoritmo é chamdao de algortimo Mímima-Soma em Camadas (LSM). Neste caso, cada submatriz pode ser vista como uma matriz H^z, e serão realizado os passos vistos para o algoritmo MS, sequencialmente. Os termos r e q utilizados possuem a representação com um superíndice que indica qual camada se refere o elemento.

1. **Inicialização**

   O valor inicial das LLR são obtidos através do vetor de números reais recebido **y**

   $$\mathbf{L} = \mathbf{y}. \tag{46}$$

   Esses valores são utilizados como $q^1_{n\rightarrow m}$. São as primeiras informações enviadas dos nós de variável para os nós de paridade, na primeira camada.

2. **Loop das** $j = 1 : z$ **camadas, Passo Horizontal**

   Esse *loop* ocorre para todas as z camadas da matriz completa.

2.1. **Atualização da matriz** R

$$r_{m\to n}^{j} = \prod_{n'\in\mathcal{N}^{j}(m)\backslash n} sign(q_{n'\to m}^{j}) \times \min_{n'\in\mathcal{N}^{j}(m)\backslash n}(|q_{n'\to m}^{j}|). \tag{47}$$

2.2. **Atualização da LLR**

$$L_n = \sum_{m\in\mathcal{M}^{j}(n)} r_{m\to n}^{j} + L_n. \tag{48}$$

2.3. **Atualização da matriz** Q **até** $j \leq z-1$

O vetor **L** é enviado para $q_{n\to m}^{j+1}$.

3. **Loop das** $j = 1 : z$ **camadas, Passo Vertical**

3.1. **Atualização da LLR**

$$L_n = L_n - \sum_{m\in\mathcal{M}^{j}(n)} r_{m\to n}^{j}. \tag{49}$$

3.2. **Atualização da matriz Q**

O vetor **L** é enviado para $q_{n\to m}^{j}$.

3.3. **Atualização da matriz R**

$$r_{m\to n}^{j} = \prod_{n'\in\mathcal{N}^{j}(m)\backslash n} sign(q_{n'\to m}^{j}) \times \min_{n'\in\mathcal{N}^{j}(m)\backslash n}(|q_{n'\to m}^{j}|). \tag{50}$$

3.4. **Atualização da LLR**

$$L_n = \sum_{m\in\mathcal{M}^{j}(n)} r_{m\to n}^{j} + L_n. \tag{51}$$

4. **Decisão**

A decisão utiliza o mesmo critério do SPA. O vetor LLR utilizado, será o último obtido após finalizar uma superiteração.

Para compreensão do algoritmo, será exemplificado o passo a passo da decodificação com uma matriz controle de paridade pequena, com a apresentação das camadas e identificação de cada uma das etapas.

Exemplo 4.2.3 *Considere o código LDPC dado no Exemplo 4.2.1 e o mesmo vetor de números reais recebido. A matriz H, pode ser divida em duas camadas, da seguinte maneira:*

$$H = \left[\frac{H_1}{H_2}\right] = \left[\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}\right].$$

Agora, inicia-se o processo das iterações do algoritmo LMS:

1. ***Inicialização***

   O vetor $\boldsymbol{L}$ *recebe os valores do vetor recebido* $\boldsymbol{y}$

$$\boldsymbol{L} = [1,53 \;\; 1,23 \;\; 0,64 \;\; 0,95 \;\; -0,45 \;\; 1,11 \;\; 0,53],$$

   e a primeira camada da matriz Q *é atualizada*

$$Q = \left[\begin{array}{ccccccc} 1,53 & 1,23 & 0,64 & 0 & -0,45 & 0 & 0 \\ 0 & 0 & 0 & 0,95 & 0 & 1,11 & 0,53 \\ \hline - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \end{array}\right].$$

2. ***Passo Horizontal***

   Para a primeira iteração, atualiza-se a matriz R^1

$$r_{11}^1 = (+1)(+1)(-1) \times |-0,45| = -0,45.$$

   Para todos outros valores de r_{mn}^1*, têm-se*

$$R = \left[\begin{array}{ccccccc} -0,45 & -0,45 & -0,45 & 0 & 0,64 & 0 & 0 \\ 0 & 0 & 0 & 0,53 & 0 & 0,53 & 0,95 \\ \hline - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \end{array}\right].$$

   O novo vetor $\boldsymbol{L}$*, será*

$$\boldsymbol{L} = [1,08 \;\; 0,78 \;\; 0,19 \;\; 1,48 \;\; 0,19 \;\; 1,64 \;\; 1,48].$$

   Este vetor é enviado para submatriz 2 *da matriz* Q*, de forma que agora ela será*

$$Q = \left[\begin{array}{ccccccc} 1,51 & 1,23 & 0,64 & 0 & -0,45 & 0 & 0 \\ 0 & 0 & 0 & 0,95 & 0 & 1,11 & 0,53 \\ \hline 1,08 & 0,78 & 0 & 1,48 & 0 & 0 & 1,48 \\ 0 & 0 & 0,19 & 0 & 0,19 & 1,64 & 0 \end{array}\right].$$

   Retorna-se ao passo de atualização da matriz R^2

$$r_{11}^2 = (+1)(+1)(+1) \times |0,78| = 0,78.$$

Para todos outros valores de r^2_{mn}, têm-se

$$R = \left[\begin{array}{ccccccc} -0,45 & -0,45 & -0,45 & 0 & 0,64 & 0 & 0 \\ 0 & 0 & 0 & 0,53 & 0 & 0,53 & 0,95 \\ \hline 0,78 & 1,08 & 0 & 0,78 & 0 & 0 & 0,78 \\ 0 & 0 & 0,19 & 0 & 0,19 & 0,19 & 0 \end{array}\right].$$

O novo vetor $\boldsymbol{L}$*, será*

$$\boldsymbol{L} = [1,86 \;\; 1,86 \;\; 0,38 \;\; 2,26 \;\; 0,38 \;\; 1,83 \;\; 2,26].$$

3. ***Passo Vertical***

Para a primeira iteração, atualiza-se o vetor $\boldsymbol{L}$

$$\boldsymbol{L} = [2,31 \;\; 2,31 \;\; 0,83 \;\; 1,73 \;\; -0,26 \;\; 1,30 \;\; 1,31].$$

Em seguida, atualiza-se a primeira camada da matriz Q

$$Q = \left[\begin{array}{ccccccc} 2,31 & 2,31 & 0,83 & 0 & -0,26 & 0 & 0 \\ 0 & 0 & 0 & 1,73 & 0 & 1,30 & 1,31 \\ \hline 1,08 & 0,78 & 0 & 1,48 & 0 & 0 & 1,48 \\ 0 & 0 & 0,19 & 0 & 0,19 & 1,64 & 0 \end{array}\right].$$

Após definir a matriz Q*, atualiza-se a matriz* R^1

$$r^1_{11} = (+1)(+1)(-1) \times |-0,26| = -0,26.$$

Para todos outros valores de r^1_{mn}, têm-se

$$R = \left[\begin{array}{ccccccc} -0,26 & -0,26 & -0,26 & 0 & 0,83 & 0 & 0 \\ 0 & 0 & 0 & 1,30 & 0 & 1,31 & 1,30 \\ \hline 0,78 & 1,08 & 0 & 0,78 & 0 & 0 & 0,78 \\ 0 & 0 & 0,19 & 0 & 0,19 & 0,19 & 0 \end{array}\right].$$

O novo vetor $\boldsymbol{L}$ *será*

$$\boldsymbol{L} = [2,05 \;\; 2,05 \;\; 0,57 \;\; 3,03 \;\; 0,57 \;\; 2,61 \;\; 2,61].$$

Para a segunda camada, atualiza-se o vetor $\boldsymbol{L}$

$$\boldsymbol{L} = [1,27 \;\; 0,97 \;\; 0,38 \;\; 2,25 \;\; 0,38 \;\; 2,42 \;\; 1,83].$$

Em seguida, atualiza-se a segunda camada da matriz Q

$$Q = \left[\begin{array}{ccccccc} 2,31 & 2,31 & -0,45 & 0 & -1,92 & 0 & 0 \\ 0 & 0 & 0 & 1,73 & 0 & 0,92 & 1,31 \\ \hline 1,27 & 0,97 & 0 & 2,25 & 0 & 0 & 1,83 \\ 0 & 0 & 0,38 & 0 & 0,38 & 2,42 & 0 \end{array}\right].$$

Após definir a matriz Q, atualiza-se a matriz R^2

$$r_{11}^2 = (+1)(+1)(+1) \times |0,97| = 0,97.$$

Para todos outros valores de r_{mn}^2, têm-se

$$R = \left[\begin{array}{ccccccc} 0,45 & 0,45 & 1,92 & 0 & 0,45 & 0 & 0 \\ 0 & 0 & 0 & 0,92 & 0 & 1,31 & 0,92 \\ \hline 0,97 & 1,27 & 0 & 0,97 & 0 & 0 & 0,97 \\ 0 & 0 & 0,38 & 0 & 0,38 & 0,38 & 0 \end{array}\right].$$

E encontra-se a LLR final

$$\boldsymbol{L} = [2,24 \;\; 2,24 \;\; 0,76 \;\; 3,22 \;\; 0,76 \;\; 2,80 \;\; 2,80],$$

através de

$$L_n > 0 \rightarrow c_n = 0,$$
$$L_n < 0 \rightarrow c_n = 1.$$

Logo, a palavra-código $c = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$ foi encontrada na primeira superiteração.

5 RESULTADOS E DISCUSSÕES

Duas medidas importantes de performance de um código corretor de erros são a razão-sinal-ruído (SNR) e a taxa de erro de bit (BER). Essas medidas trazem a razão entre a potência do sinal e a potência do ruído do sistema de comunicação, e o número de bits errados que o receptor identifica em relação aos bits de mensagem enviado, respectivamente.

A SNR é uma métrica indicada para analisar a proporção do efeito do ruído dada uma potência de sinal. Tipicamente, ela é definida na unidade logarítmica decibel. Para uma SNR alta, significa que a potência do sinal é muito maior que a potência do ruído, consequentemente a taxa de erro será reduzida. Seja P_s a potência do sinal, e P_n a potência do ruído, a SNR em decibel será

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_s}{P_n} \right). \tag{1}$$

A BER apresenta a probabilidade do bit recebido $\hat{w}_i$ ser diferente do bit enviado w_i. Normalmente, essa métrica é utilizada para contabilizar erros de mensagens com grandes comprimentos. Em grande parte dos sistemas de comunicação, o valor aceitável dessa taxa é de 1 bit errado para cada 1 milhão de bits enviados, ou seja, 10^{-6}. Seja n_e o número de bits com erro, e N_b o número de bits enviados, a BER será

$$BER = \frac{n_e}{N_b}. \tag{2}$$

Em um sistema de comunicação com ruídos do tipo AWGN e modulação BPSK, a partir da SNR, é possível desenvolver uma função para encontrar a taxa de erro de bit. Como limiar de decisão, utiliza-se o valor de sinal de 0. Assim, caso o valor seja menor que 0, o sistema entenderá que foi recebido o bit 1 e, ao contrário, o sistema entenderá que o bit recebido será 0. Para a situação em que o bit enviado seja $0 \rightarrow s = +1$ e o valor do ruído seja $n < -1$, acontecerá um erro. Para o caso do bit enviado $1 \rightarrow s = -1$ e o valor do ruído seja $n > +1$, também ocorrerá um erro

$$BER = Prob\left\{s = +1\right\} Prob\left\{n < -1\right\} + Prob\left\{s = -1\right\} Prob\left\{n > +1\right\}. \tag{3}$$

Os símbolos $+1$ e -1 são equiprováveis, ou seja, as probabilidades de cada um ser enviado, é de $1/2$. Para o caso das probabilidades dos ruídos, é preciso retornar à função densidade de probabilidade de um ruído gaussiano para $\mu = 0$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}}, \tag{4}$$

e assim as duas probabilidades $Prob\left\{n < -1\right\}$ e $Prob\left\{n > +1\right\}$ podem ser representadas através da função $Q(1/\sigma)$, que é a função distribuição cumulativa complementar. A equação da BER em função da SNR será

$$BER = Q(1/\sigma) = Q(\sqrt{SNR}). \tag{5}$$

Para um sistema sem codificação, cada bit enviado representa uma informação. De acordo com a Definição 2.2.4, para todos os casos onde as palavras enviadas são codificadas, aumentar o número de

bits da palavra-código que não contém informação, significa reduzir a taxa. Consequentemente, para enviar uma mesma mensagem, reduzir a taxa necessita uma maior potência. Por essa razão, a métrica recomendada para calcular a taxa de erro de bit é a energia por bit de informação

$$E_b = \frac{E_s}{R}. \tag{6}$$

A SNR pode então ser definida como a relação entre a energia por bit de informação e a potência do ruído definida por $N_0/2 = \sigma^2$

$$SNR = \frac{E_s}{\sigma^2} = 2R\left(\frac{E_b}{N_0}\right), \tag{7}$$

e a BER pode ser escrita por meio da substituição de (7) em (5)

$$BER = Q\left(\sqrt{2R\left(\frac{E_b}{N_0}\right)}\right). \tag{8}$$

Como requisito referente a codificação de canal, o cenário de comunicação 5G NR é especificado para taxas de erro de bit menores que 10^{-6}, de modo a apresentar uma Comunicação Ultraconfiável (URC, *Ultra-Reliable Communication*) em um canal fixo sem a análise de possíveis desvanecimentos [Popovski 2014, Lema et al. 2017, 3GPP-138.212 2022]. Neste sentido, para todas as simulações que serão apresentadas neste trabalho, é procurado uma relação que atinja esse requisito, tanto para variação de taxa do código, de comprimento do bloco ou do número de iterações.

A mensagem enviada será uma mensagem aleatória de tamanho k codificada para uma palavra-código de tamanho n. Essa mensagem é modulada em BPSK, onde os bits $0 \rightarrow +1$ e os bits $1 \rightarrow -1$, e transmitida através de um canal gaussiano com $N(0, \sigma^2)$. Esse canal insere um ruído AWGN independente para cada bit, de acordo com a relação sinal-ruído estipulada para cada simulação. O vetor recebido é um vetor de números reais que será decodificado através do algoritmo LMS.

O algoritmo de codificação utilizado nas simulações se baseia nas características das matrizes bases utilizadas. Cada linha da matriz base (antes de ser expandida por Z_c) é considerada uma camada para realização dos passos. É possível notar que, como cada elemento da linha é representado por uma matriz identidade (ou nula), o peso das colunas sempre será igual à 1 ($w_c = 1$). O algoritmo é inicializado com uma mensagem codificada adicionada de ruído, o que gera um vetor com estimativas de probabilidade. É então executado a subiteração com o *loop* das M camadas, que realiza as etapas do algoritmo Mínima-Soma sequencialmente e atualiza as matrizes Q e R. Após completar esse passo, é considerada uma superiteração. Ao utilizar o termo iterações, para esse algoritmo, considera-se que é realizada uma superiteração.

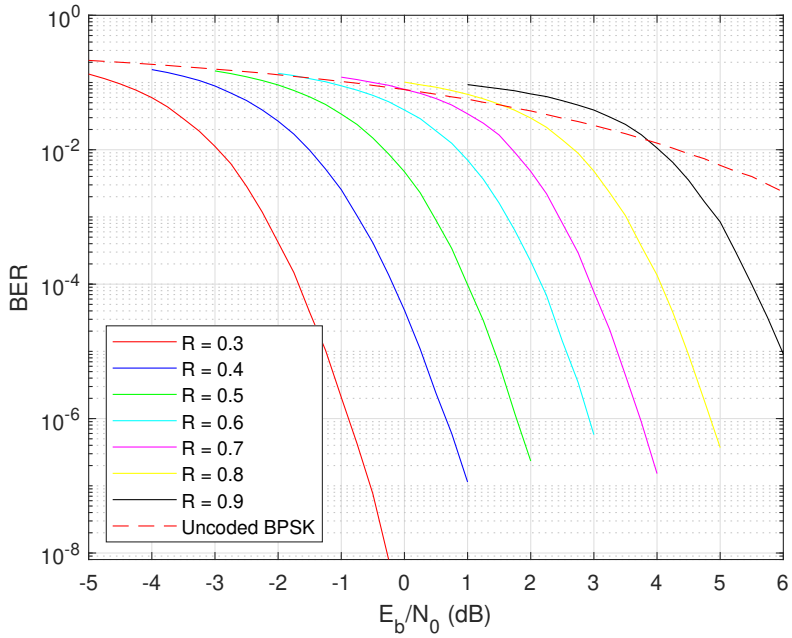
Essa estratégia de decodificação por camadas, reduz o tempo de processamento e melhora o algoritmo MS para uma performance próxima ao algoritmo SPA. A utilização do algoritmo Soma-Produto e análises desse algoritmo em códigos LDPC aplicado à sistemas de Comunicações Ópticas Coerentes foram apresentadas em [Gonçalves 2021].

Como critérios de simulação, o código utiliza um limite de erros de blocos ou de blocos enviados. Para os primeiros valores da razão sinal-ruido, o algoritmo tende a utilizar o critério de parada de 200

blocos errados visto que para razões muito pequenas o código encontra maior número de erros. Para os maiores valores, o código utiliza como critério de parada, 100000 blocos enviados. A cada bloco enviado são contabilizados os erros a partir da comparação da palavra-código decodificada com o conjunto das possíveis palavras-código do sistema.

A primeira das simulações, consiste na análise da matriz BG1 para diferentes taxas com o comprimento do bloco de informações fixo em $K = 5000$. De acordo com as normas do 3GPP e a Figura 8, para um valor de $K > 3824$, a utilização da matriz base BG1 depende da taxa ser $R > 0,25$. Portanto, para primeira simulação variou-se a taxa de $R = 0,3$ a $R = 0,9$, com o passo de $0,1$. A Figura 12 apresenta os resultados obtidos da simulação.

Figura 12 – Curvas de BER para matriz BG1 com $K = 5000$ para diferentes taxas.



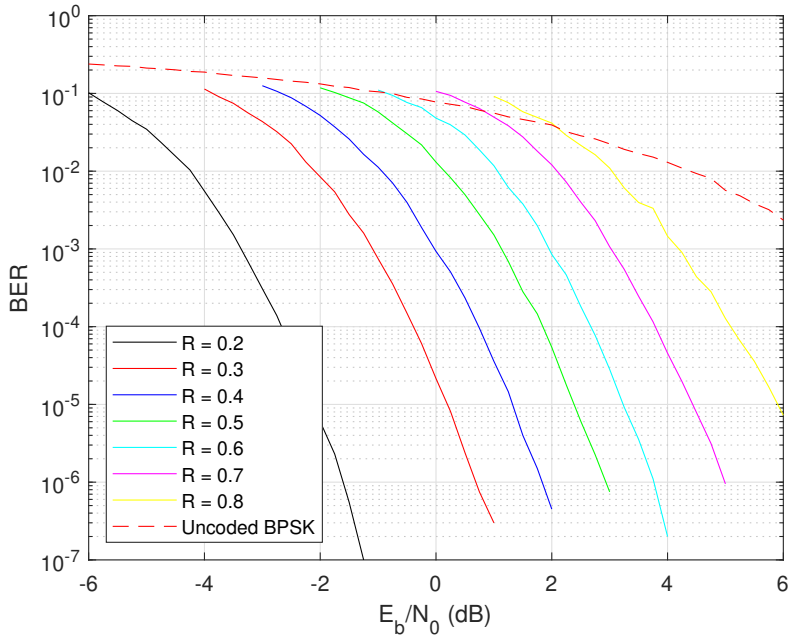
Fonte: Produção do Próprio Autor.

Todos os valores de taxa, com exceção de $R = 0,9$, atingem o requisito mínimo para a BER. Através desse gráfico é possível identificar que para taxas cada vez menores, o código possui um desempenho melhor. Isso ocorre porque, ao reduzir a taxa, significa que menos bits de informação estão sendo enviados em relação ao número de bits de paridade. Aumentar a quantidade de bits de paridade, traz uma maior confiabilidade e capacidade do código para realizar a decodificação das palavras-código ruidosas recebidas.

A próxima simulação estende o mesmo conceito abordado para a matriz BG1, agora para a matriz BG2. Com um valor de $K = 200$ fixo, variou-se as taxas de $R = 0,2$ a $R = 0,8$, com o passo de $0,1$ conforme as possibilidades descritas na documentação do 3GPP. De acordo com a Figura 8, para um comprimento $K \leq 292$, qualquer taxa definida, utiliza a matriz BG2. A Figura 13 apresenta os resultados obtidos da simulação.

Observa-se novamente que para um mesmo comprimento de mensagem enviada, ao reduzir a taxa, o código apresenta melhor desempenho. Com exceção de $R = 0,8$, todas as outras taxas foram capazes de atingir o valor mínimo da BER requisitado. Com isso, confirma-se a análise feita para a

Figura 13 – Curvas de BER para matriz BG2 com $K = 200$ para diferentes taxas.



Fonte: Produção do Próprio Autor.

matriz BG1. Ou seja, reduzir a taxa, diminui a proporção de bits de informação para bits de paridade, o que permite enviar uma mesma mensagem com menor razão sinal-ruído.

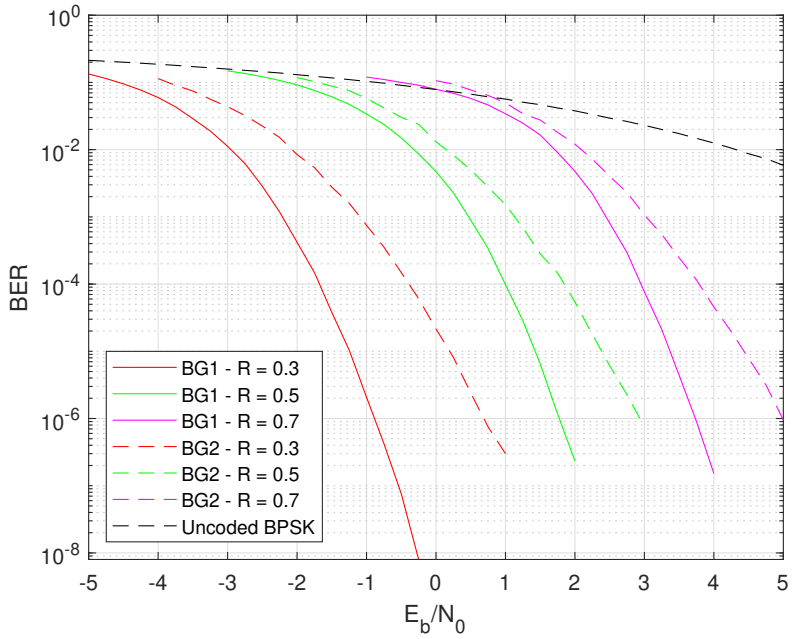
Para analisar as diferenças entre as matrizes BG1 e BG2, os resultados para taxas fixas e diferentes comprimentos foram comparados em um mesmo gráfico. Primeiro, utilizou-se os comprimentos de códigos das duas simulações descritas anteriormente, ou seja, $K = 5000$ para BG1, e $K = 200$ para BG2, com as taxas $R = 0,3$, $R = 0,5$, e $R = 0,7$. É importante destacar que ao utilizar as matrizes descritas na documentação do 3GPP, é impossível ter o uso das duas matrizes para uma mesma taxa e comprimento do bloco de informação.

A Figura 14 apresenta a diferença entre os códigos. Para uma mesma taxa, em todos os casos, a matriz BG1 apresenta um melhor desempenho. Com menor relação sinal-ruído ela converge para a BER de 10^{-6} mais rápido que a matriz BG2. Isso evidencia que, o aumento das dimensões da matriz controle de paridade, afeta o desempenho proporcionalmente. Além de possuir um comprimento do bloco maior, a matriz base 1 possui dimensões maiores que a matriz base 2. Essa característica é referente ao tipo de código utilizado. Um código LDPC apresenta melhor desempenho para matrizes controle de paridade maiores e com menor densidade.

Em sequência, para realizar uma comparação mais justa, utilizou-se comprimentos próximos de mensagem, $K = 4000$ para BG1 e $K = 3500$ para BG2. De acordo com a Figura 8, para estes comprimentos de bloco de informação, não é possível utilizar as taxas utilizadas na simulação anterior. Para isso, definiu-se o uso das taxas $R = 0,4$ e $R = 0,5$, o que permite a utilização das respectivas matrizes.

A Figura 15 apresenta uma diferença menor entre os desempenhos das matrizes base. Observa-se que ainda assim, as matrizes BG1 têm um desempenho superior às matrizes BG2. Reduzir o comprimento do bloco de informação que aquela transmite, não é suficiente para que esta ultrapasse o

Figura 14 – Curvas de BER para matrizes BG1 com $K = 5000$ e BG2 com $K = 200$ para diferentes taxas.



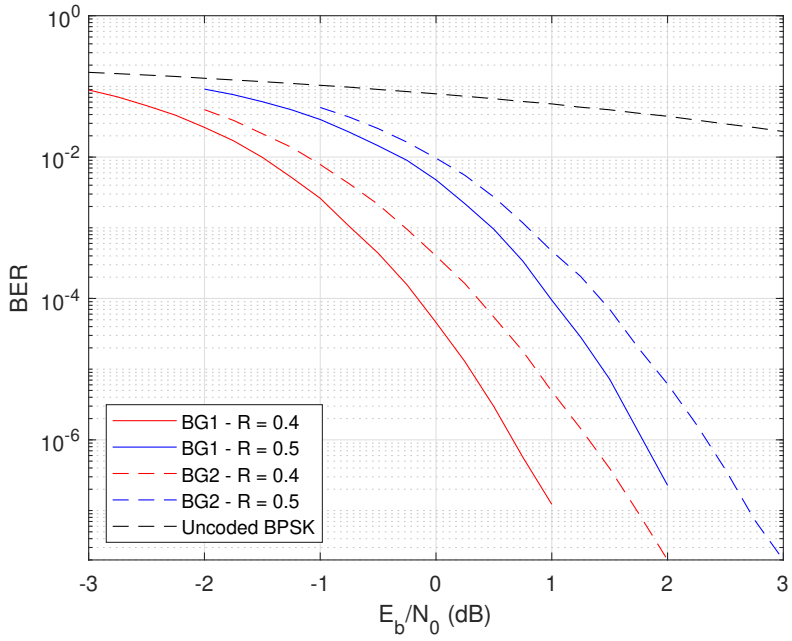
Fonte: Produção do Próprio Autor.

seu desempenho.

Por fim, a última análise realizada é sobre o efeito do número de iterações realizadas no algoritmo para a decodificação. Essa simulação tem como objetivo identificar como repetir os passos melhoram o desempenho do código. Como referência, foi utilizado uma taxa fixa $R = 0,5$ e dois comprimentos que permitem a verificação para as duas matrizes bases, $K = 4000$ para a matriz BG1 e $K = 3500$ para a matriz BG2. O número de iterações variou entre 4, 7 e 10 repetições.

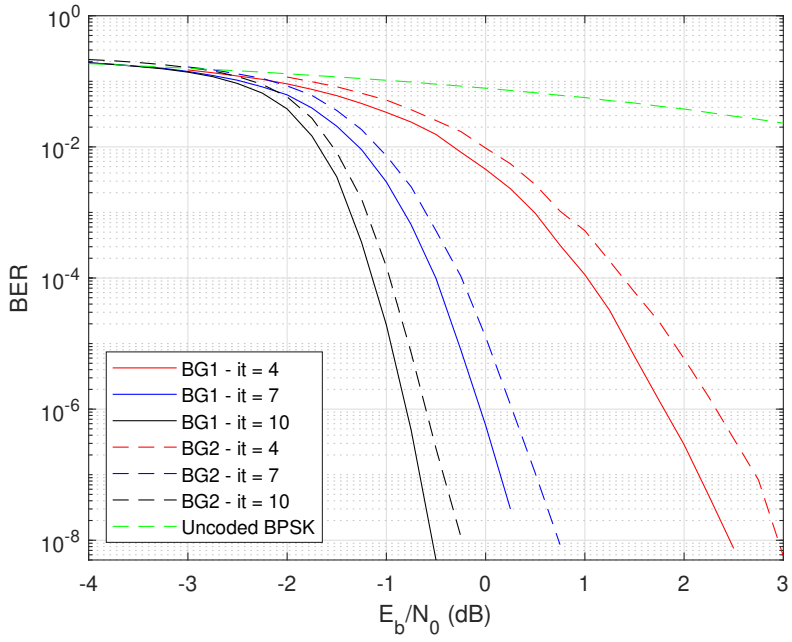
Através da Figura 16, é possível notar que aumentar o número de iterações aumenta o desempenho do código. Porém, aumentar o número de iterações significa aumentar o tempo de processamento da decodificação. Com isso, é encontrado um *trade-off* entre qualidade da decodificação e tempo de processamento. Novamente, observa-se que para comprimentos maiores, o algoritmo se comporta melhor. E, para menores relações sinal-ruído, ele atinge o valor mínimo requisitado para a taxa de erro de bits. Isso ocorre pelo fato de que ao rodar mais iterações, o código tem maior chance de encontrar os erros associados aos bits que sofreram alteração devido ao ruído do canal gaussiano.

Figura 15 – Curvas de BER para matrizes BG1 com $K = 4000$ e BG2 com $K = 3500$ para diferentes taxas.



Fonte: Produção do Próprio Autor.

Figura 16 – Curvas de BER para matrizes BG1 com $K = 5000$ e BG2 com $K = 200$ e taxas $R = 0.5$ para diferentes iterações.



Fonte: Produção do Próprio Autor.

6 CONCLUSÃO

Este trabalho apresentou um estudo sobre códigos LDPC construídos a partir de matrizes base e com algoritmos de decodificação com decisão suave para analisar a eficiência destes códigos em cenários 5G NR, descrito na introdução com as suas regulamentações para canal, modulação e técnicas de codificação para esse padrão. Para isso, o foi apresentado alguns conceitos básicos importantes para a codificação de canal. Iniciou com definições da Teoria da Informação para chegar à capacidade do canal de comunicação. A partir disso, descreveu-se o que é um código de bloco linear e foi exemplificado o código de Hamming $(7, 4)$, apresentando a taxa de um código e os conceitos de decisão abrupta e decisão suave. Após apresentar os conceitos iniciais foi definido o código LDPC, apresentando os grafos bipartidos, os códigos LDPC regulares e a estratégia proposta por Gallager para a construção da matriz controle de paridade para estes códigos. Na decodificação, foi descrito o algoritmo *bit-flipping* que é uma das técnicas baseadas em decisão abrupta.

Nos códigos LDPC aplicados ao 5G NR, descreveu-se como as matrizes controle de paridade são definidas a partir do conceito de matriz base e como estas são selecionadas. Também foi apresentado a forma de construção das matrizes base baseado na documentação do 3GPP sobre as normas de codificação de canal. Os algoritmos de decodificação apresentados, utilizaram o conceito da razão logarítmica de verossimilhança aplicado a códigos SPC, para então descrever tais algoritmos: SPA, MS, e LMS baseados em tangentes hiperbólicas. Cada um desses algoritmos foi exemplificado com uma matriz de tamanho muito pequeno, pelo fato de que as matrizes do 5G NR são de grandes dimensões e impossíveis de serem descritas em pequenos exemplos.

As simulações apresentadas foram realizadas utilizando as matrizes bases BG1 e BG2 para a codificação e o algoritmo LMS para a decodificação. Foi considerado nas simulações a modulação BPSK e as mensagens foram transmitidas por um canal gaussiano com ruído AWGN. Os parâmetros avaliados foram as taxas de erro de bits em função da razão sinal-ruído. As simulações consideraram a variação de comprimento e taxa dos códigos, além do número de iterações no algoritmo de decodificação. Em todos os resultados apresentados, a taxa de erro de bit obtida alcançou 10^{-6} como é o desejado no cenário 5G NR. Através das simulações realizadas encontra-se algumas conclusões sobre o algoritmo MS na decodificação de palavras-código nos padrões do 5GNR. A matriz BG1 apresenta um melhor comportamento do que a matriz BG2. Uma das razões para essa melhora se deve ao fato das dimensões e comprimentos de mensagem utilizadas pela BG1 serem maiores que da BG2. Isso comprova que códigos LDPC possuem melhor eficácia em matrizes grandes. Em relação ao número de iterações, aumentá-la traz melhor qualidade da decodificação. Associada a isso, o código aumenta o tempo de execução causando um atraso para encontrar a mensagem enviada. Através desse *trade-off*, é possível encontrar um resultado ótimo que seja capaz de minimizar o uso de potência para enviar a mensagem e que não tenha um atraso muito grande para a decodificação. Importante observar que como o 5G NR trabalha com diferentes faixas de frequência, haverá a necessidade de que para todas as taxas e comprimentos, o código atinja a BER necessária, o que ocorreu em todas as simulações, com exceções das taxas $R = 0.9$ para a matriz BG1 e $R = 0.8$ para a matriz BG2.

Para continuação do trabalho, é possível fazer a análise da decodificação que se estende para evoluções do algoritmo LMS, como por exemplo a normalização da mínima-soma com a adição de um *offset* para aproximar o desempenho ao algoritmo SPA [Emran e Elsabrouty 2014], bem como utilizar técnicas de modulação atuais como o QPSK e o M-QAM para modulação da portadora do OFDM. A utilização de canais móveis com desvanecimentos, também pode ser estudada para caracterizar o desempenho dos códigos LDPC. Além disso, é possível analisar mais alguns métodos e normas explicitados na documentação do 3GPP, como por exemplo o *Rate Matching* [Hamidi-Sepehr, Nimbalker e Ermolaev 2018].

REFERÊNCIAS

3GPP-138.104. 5g; nr; base station (bs) radio transmission and reception. *ETSI TS 138 104*, 2022.

3GPP-138.211. 5g; nr; physical channels and modulation. *ETSI TS 138 211*, 2022.

3GPP-138.212. 5g; nr; multiplexing and channel coding. *ETSI TS 138 212*, 2022.

BALDI, M. **QC-LDPC Code-Based Cryptosystems**. Ancona, Marcas - Italy: Springer International Publishing, 2014.

CHELIKANI, N. **5G-NR DL-SCH LDPC Channel Coding Base Graph selection and Coding Procedure**. 2019. Disponível em: <https://www.linkedin.com/pulse/5g-nr-dl-sch-ldpc-channel-coding-base-graph-selection-chelikani/>.

COVER, T. M.; THOMAS, J. A. **Elements of Information Theory 2nd Edition**. New York, New York - United States: Wiley-Interscience, 2006.

EBU-UER. Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (dvb-s2). **ETSI EN 302 307**, 2009.

EMRAN, A. A.; ELSABROUTY, M. Simplified variable-scaled min sum ldpc decoder for irregular ldpc codes. In: **2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)**. [S.l.: s.n.], 2014. p. 518–523.

FOSSORIER, M.; MIHALJEVIC, M.; IMAI, H. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. **IEEE Transactions on Communications**, v. 47, n. 5, p. 673–680, 1999.

GALLAGER, R. Low-density parity-check codes. **IRE Transactions on Information Theory**, v. 8, n. 1, p. 21–28, 1962.

GOLDSMITH, A. **Wireless Communications**. Palo Alto, California - United States: Cambridge University Press, 2005.

GONÇALVES, J. O. **Análise de Desempenho de Códigos LDPC em Sistemas de Comunicações Ópticas Coerentes**. 2021. Disponível em: <http://hdl.handle.net/11449/214055>.

HAGENAUER, J.; OFFER, E.; PAPKE, L. Iterative decoding of binary block and convolutional codes. **IEEE Transactions on Information Theory**, v. 42, n. 2, p. 429–445, 1996.

HAMIDI-SEPEHR, F.; NIMBALKER, A.; ERMOLAEV, G. Analysis of 5g ldpc codes rate-matching design. In: **2018 IEEE 87th Vehicular Technology Conference (VTC Spring)**. [S.l.: s.n.], 2018. p. 1–5.

HASAN, A. A. **Efficient Log Likelihood Ratio Estimation for Polar Codes**. Tese (Doutorado) — Carleton University, Ottawa, Ontario - Canada, 2018. COLOCARTEXTO.

HOCEVAR, D. A reduced complexity decoder architecture via layered decoding of ldpc codes. In: **IEEE Workshop onSignal Processing Systems, 2004. SIPS 2004.** [S.l.: s.n.], 2004. p. 107–112.

HU, X.-Y. et al. Efficient implementations of the sum-product algorithm for decoding ldpc codes. In: **GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)**. [S.l.: s.n.], 2001. v. 2, p. 1036–1036E vol.2.

JIANG, M.; FAN, D. A low-latency bf decoding of ldpc codes with dynamic thresholds. **IEEE Communications Letters**, v. 25, n. 9, p. 2781–2785, 2021.

LEMA, M. A. et al. Business case and technology analysis for 5g low latency applications. **IEEE Access**, v. 5, p. 5917–5935, 2017.

LIN, S.; COSTELLO, D. J. **Error Control Coding**. [S.l.]: Prentice hall New York, 2001. v. 2.

MACKAY, D. Good error-correcting codes based on very sparse matrices. **IEEE Transactions on Information Theory**, v. 45, n. 2, p. 399–431, 1999.

MACKAY, D. J.; NEAL, R. M. Near shannon limit performance of low density parity check codes. **Electronics letters**, v. 33, n. 6, p. 457–458, 1997.

NGUYEN, T. T. B.; TAN, T. N.; LEE, H. Efficient qc-ldpc encoder for 5g new radio. **Electronics**, MDPI, v. 8, n. 6, p. 668, 2019.

OSSEIRAN, A.; MONSERRAT, J. F.; MARSCH, P. **5G mobile and wireless communications technology**. [S.l.]: Cambridge University Press, 2016.

PERAHIA, E.; STACEY, R. **Next generation wireless LANs: 802.11 n and 802.11 ac**. [S.l.]: Cambridge university press, 2013.

POPOVSKI, P. Ultra-reliable communication in 5g wireless systems. In: **1st International Conference on 5G for Ubiquitous Connectivity**. [S.l.: s.n.], 2014. p. 146–151.

POPOVSKI, P. Ultra-reliable communication in 5g wireless systems. In: **1st International Conference on 5G for Ubiquitous Connectivity**. [S.l.: s.n.], 2014. p. 146–151.

RICHARDSON, T.; URBANKE, R. The renaissance of gallager's low-density parity-check codes. **IEEE Communications Magazine**, v. 41, n. 8, p. 126–131, 2003.

SHANNON, C. E. A mathematical theory of communication. **The Bell system technical journal**, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948.

TANNER, R. A recursive approach to low complexity codes. **IEEE Transactions on Information Theory**, v. 27, n. 5, p. 533–547, 1981.

TU, Z.; ZHANG, S. Overview of ldpc codes. In: **7th IEEE International Conference on Computer and Information Technology (CIT 2007)**. [S.l.: s.n.], 2007. p. 469–474.

WANG, Y.; YEDIDIA, J.; DRAPER, S. Construction of high-girth qc-ldpc codes. In: **2008 5th International Symposium on Turbo Codes and Related Topics**. [S.l.: s.n.], 2008. p. 180–185.

ZARKESHVARI, F.; BANIHASHEMI, A. On implementation of min-sum algorithm for decoding low-density parity-check (ldpc) codes. In: **Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE**. [S.l.: s.n.], 2002. v. 2, p. 1349–1353 vol.2.