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

JÉSSICA OLIVEIRA GONÇALVES

Análise de Desempenho de Códigos LDPC em Sistemas de Comunicações Ópticas Coerentes

São João da Boa Vista
2021

JÉSSICA OLIVEIRA GONÇALVES

Análise de Desempenho de Códigos LDPC em Sistemas de Comunicações Ópticas Coerentes

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

Orientador: Profa. Dra. Cintya Wink de Oliveira Benedito
Coorientador: Profº Dr. Ivan Aritz Aldaya Garde

São João da Boa Vista
2021

G635a

Gonçalves, Jéssica Oliveira

Análise de desempenho de códigos LDPC em sistemas de comunicações ópticas coerentes / Jéssica Oliveira Gonçalves. -- São João da Boa Vista, 2021

73 f.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Telecomunicações) - Universidade Estadual Paulista (Unesp), Câmpus Experimental de São João da Boa Vista, São João da Boa Vista

Orientadora: Cintya Wink de Oliveira Benedito

Coorientador: Ivan Aritz Aldaya Garde

1. Códigos corretores de erros (Teoria da informação). 2. Teoria da codificação. 3. Comunicações ópticas. 4. Telecomunicações. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Câmpus Experimental de São João da Boa Vista. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

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

TRABALHO DE CONCLUSÃO DE CURSO

ANÁLISE DE DESEMPENHO DE CÓDIGOS LDPC EM SISTEMAS DE COMUNICAÇÕES ÓPTICAS COERENTES

Aluno: Jéssica Oliveira Gonçalves
Orientadora: Profª. Drª. Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientadora)
- Marcelo Luís Francisco Abbade (Examinador)
- Jozue Vieira Filho (Examinador)

A ata da defesa com as respectivas assinaturas dos membros encontra-se no prontuário do aluno (Expediente nº 005/2019)

São João da Boa Vista, 04 de março de 2021

Aos meus pais e irmãos, pois sem eles eu nada seria.

AGRADECIMENTOS

Primeiramente à Deus, por ter me dado forças e me sustentado até aqui.

Aos meus pais Adão e Rosalina e irmãos Elton e Daiane por todo amor e apoio incondicional.

À minha orientadora, Profª. Dra. Cintya Wink de Oliveira Benedito, que me auxiliou, incentivou e não desistiu de mim.

Ao meu coorientador, Profº. Dr. Ivan Aritz Aldaya Garde que me deu todo o suporte necessário.

Aos membros da banca examinadora pela disponibilidade, atenção e conselhos na realização deste trabalho.

Ao meu namorado Anderson Tregancini Junior, que esteve a todo momento do meu lado me dando forças e apoio para finalizar este trabalho.

À todos os professores e funcionários da Unesp SJBV pelos anos que por aqui estive, me ajudando a evoluir como pessoa e profissional.

Às minhas amigas de faculdade e da vida Larissa, Letícia e Melissa por toda a parceria e companheirismo dentro e fora da sala de aula.

Aos meus amigos Vanessa, Mateus, Pedro, Vinícius e Laura por todos os desabafos que tiveram que aguentar e por me incentivarem a não desistir.

Ao meu psiquiatra Marcelo de Toledo e a minha psicóloga Aline Bordinhon, que em conjunto me prestaram o melhor suporte ao longo de todo processo.

Gratidão!

“O fardo é proporcional às forças, como a recompensa será proporcional à resignação e à coragem.”
(ALLAN KARDEC)

RESUMO

A codificação de canal consiste na inserção de bits redundantes na informação a ser enviada, de modo a proteger a informação, possibilitando ao receptor recuperá-la sem a necessidade de retransmissão. Esta técnica foi apresentada por Shannon em 1948 e desde então vem sendo amplamente estudada e diversas estratégias de codificação de canal tem sido apresentadas. Nas últimas décadas, a técnica denominada como correção antecipada de erros (*Foward Error Correction*, FEC) baseada em códigos de verificação de baixa densidade (*Low Density Parity Check*, LDPC) vem sendo amplamente utilizada em diversas áreas, e tem se mostrado muito promissora em aplicações relacionadas a sistemas de comunicações ópticas coerentes, devido à sua eficiência em transmissões de longo alcance, bem como em transmissões com alta taxa de informação. Este trabalho tem como objetivo principal explorar a teoria e as técnicas para a obtenção de códigos LDPC eficientes, aplicando uma técnica de codificação a partir de sua matriz controle de paridade obtida por meio de um método conhecido como projeto de bloco incompleto balanceado (*Balanced Incomplete Block Design*, BIBD) e uma decodificação iterativa através de um algoritmo do tipo *soft-decision* chamado de algoritmo soma-produto (*Sum-Product Algorithm*, SPA). Serão realizadas simulações computacionais utilizando a linguagem de programação Python a fim de verificar o desempenho dos códigos LDPC em um canal de comunicação óptico, este canal foi modelado a partir de um canal gaussiano, em que utilizando certos critérios pôde ser aproximado para um canal com ruído aditivo gaussiano e branco (*Additive White Gaussian Noise*, AWGN). Os resultados mostram a possibilidade de utilizar um método de obtenção da matriz verificadora de paridade diferente do tradicional proposto por Gallager, além de demonstrar a ligação direta entre o aumento do desempenho ao número de iterações no algoritmo de decodificação SPA e possibilitando mensurar o ganho de codificação em relação a SNR, que no melhor cenário simulado foi de 4.91 dB.

PALAVRAS-CHAVE: Codificação de Canal. Códigos LDPC. Canal AWGN. Comunicações ópticas coerentes.

ABSTRACT

Channel encoding consists of the insertion of redundant bits in the information to be sent, in order to protect the information, allowing the receiver to recover it without the need for retransmission. This technique was introduced by Shannon in 1948 and since then it has been widely studied and several channel coding strategies have been presented. In the last decades, the technique known as foward error correction (FEC) based on low density verification codes (LDPC) has been widely used in several areas, but has shown to be very promising in applications related to coherent optical communications systems, due to their efficiency in long-range transmissions, as well as in transmissions with a high information rate. This work has as main objective to explore the theory and techniques for obtaining efficient LDPC codes, applying a coding technique from its parity control matrix obtained through a method known as balanced incomplete block design (BIBD) and an iterative decoding using a soft-decision algorithm called the sum-product algorithm (SPA). Computational simulations will be performed using the Python programming language in order to verify the performance of LDPC codes in an optical communication channel. This channel was modeled from a Gaussian channel, where using certain criteria it could be approximated to a channel with additive noise gaussian and white (AWGN). The results show the possibility of using a method of obtaining the parity check matrix different from the traditional one proposed by Gallager, in addition to demonstrating the direct link between the increase in performance and the number of iterations in the SPA decoding algorithm and allowing to measure the coding gain in relation to SNR, which in the best simulated scenario was 4.91 dB.

KEYWORDS: Channel encoding. LDPC codes. AWGN channel. Coherent optical communications.

LISTA DE ILUSTRAÇÕES

Figura 1 Diagrama de blocos do sistema de comunicação digital básico 16
Figura 2 Configuração de um receptor coerente 21
Figura 3 Configuração do receptor homódino de diversidade de fase 23
Figura 4 Configuração do receptor homódino de diversidade de polarização/fase 24
Figura 5 Configuração básica de um receptor coerente digital 25
Figura 6 Diagrama de blocos do processamento digital de sinais 25
Figura 7 Modulador de Mach-Zehnder 26
Figura 8 Modulador de fase e quadratura IQ 26
Figura 9 (a) BPSK (b) QPSK 27
Figura 10 Canal BSC 32
Figura 11 Canal AWGN 32
Figura 12 Palavra-código 35
Figura 13 Grafo Bipartido 43
Figura 14 Grafo Bipartido com girth 6 45
Figura 15 Distribuição de pontos em um grid retangular 48
Figura 16 Representação dos slopes 0, 1 e 2 48
Figura 17 Grid Retangular com slopes= 0, 1, 2 49
Figura 18 Diagrama de blocos das simulações 62
Figura 19 Simulação da BER para um código LPDC $C(1369, 1260)$ obtido utilizando os métodos de Gallager e BIBD 63
Figura 20 Influência do número de iterações no algoritmo de decodificação SPA para um código LDPC $C(1369, 1260)$ 64
Figura 21 Comparação da BER entre códigos LDPC com parâmetros $C(529, 462)$ e $C(4489, 4158)$ 65
Figura 22 Evolução de performance para cada iteração de um código LDPC $C(1369, 1260)$ 66
Figura 23 Ganho de codificação para cada iteração de um código LDPC $C(1369, 1260)$. 67

LISTA DE TABELAS

Tabela 1 – Codificação utilizando a matriz geradora 35
Tabela 2 – Codificação utilizando a matriz geradora na forma padrão 36
Tabela 3 – Tabela de Padrão de Erros 38
Tabela 4 – Tabela de Síndromes 39
Tabela 5 – Tabela Slopes 49
Tabela 6 – Teste 1 - Equações de Paridade 51
Tabela 7 – Teste 2 - Equações de Paridade 52
Tabela 8 – Teste 3 - Equações de Paridade 52
Tabela 9 – Parâmetros dos códigos LDPC 61

LISTA DE ABREVIATURAS E SIGLAS

FEC	*Foward Error Correction*
OOK	*On-Off-Keying*
MZM	*Mach-Zehnder Modulator*
EAM	*Electroabsorption Modulator*
ASK	*Amplitude Shift Keying*
SOP	*State of Polarization*
LO	*Local Oscillator*
PSK	*Phase Shift Keying*
FSK	*Frequency Shift Keying*
BPSK	*Binary Phase Shift Keying*
QPSK	*Quadrature Phase Shift Keying*
QAM	*Quadrature Amplitude Modulation*
EDFA	*Erbium Doped Fiber Amplifier*
ASE	*Amplified Spontaneous Emission*
PMD	*Polarization Mode Dispersion*
PM	*Phase Modulator*
WDM	*Wavelength Division Multiplexing*
DWDM	*Dense Wavelength Division Multiplexing*
DSP	*Digital Signal Processing*
IF	*Intermediate Frequency*
PBS	*Polarization Beam Splitter*
ADC	*Analog to Digital Converter*
GVD	*Group-Velocity Dispersion*
DC	*Direct Current*
OPPL	*Optical Phase-Locked Loop*

CG	*Coding Gain*
NCG	*Net Coding Gain*
BSC	*Binary Symmetric Channel*
AWGN	*Additive White Gaussian Noise*
LDPC	*Low Density Parity Check*
SPA	*Sum-Product Algorithm*
BIBD	*Balanced Incomplete Block Design*
BER	*Bit Error Rate*
SNR	*Signal-to-Noise Ratio*
FWM	*Four Wave Mixing*
XPM	*Cross Phase Modulation*

LISTA DE SÍMBOLOS

E_s	Campo elétrico do sinal
E_{LO}	Campo elétrico do oscilador local
E_1	Campo elétrico superior
E_2	Campo elétrico inferior
R	Responsividade
P_s	Potência do sinal óptico
P_{LO}	Potência do oscilador local
ω_{IF}	Frequência angular intermediária
ω_s	Frequência angular do sinal
ω_{LO}	Frequência angular do oscilador local
$\theta_{sig}(t)$	Fase do sinal em função do tempo
$\theta_{LO}(t)$	Fase do oscilador local em função do tempo
$\theta_s(t)$	Modulação de fase
$\theta_{sn}(t)$	Ruído de fase
$I(t)$	Corrente de saída do fotodiodo
$I_c(t)$	Corrente em função da amplitude complexa
I	Componente em fase
Q	Componente em quadratura
$I_I(t)$	Corrente com informação de fase
$I_Q(t)$	Corrente com informação de quadratura
$p(x_k)$	Probabilidade de um evento de interesse
$I[x_k]$	Quantidade de informação de um evento
X	Variável aleatória
H(X)	Entropia
Y	Variável aleatória

I(X;Y)	Informação mútua
C	Capacidade do canal
P	Potência de transmissão
σ^2	Variância
n	Tamanho da palavra-código
k	Tamanho da mensagem
d	Distância mínima de Hamming
F_q	Subespaço de q elementos
R_c	Taxa de codificação
d_h	Distância de Hamming
$w(u)$	Peso de Hamming
G	Matriz geradora
u	Vetor de entrada em um codificador
v	Vetor de saída em um codificador
B_C	Base de um subespaço
G'	Matriz geradora na forma sistemática
H	Matriz controle de paridade
r	Vetor recebido
e	Vetor erro
S	Síndrome
r'	Vetor recebido estimado
h_i	i-ésima coluna de H
M	Número de linhas da matriz H
N	Número de colunas da matriz H
w_r	Peso de 1's nas linhas
w_c	Peso de 1's nas colunas
$\mathcal{D}$	Densidade da matriz H

z_m	Nó de função ou nó de paridade
c_n	Nó de variável
M_n	Conjunto de índices de verificações de paridade conectados ao bit c_n codificado
N_m	Conjunto de índices dos bits codificados conectados ao nó de verificação de paridade z_m
l	Tamanho do ciclo
α	Número de repetições de um bit
β	Tamanho do subconjunto de αk
PA	Permutação aleatória
r''	Vetor **r'** atualizado
$q_{mn}(x)$	Probabilidade do n-ésimo bit codificado satisfazer a equação de paridade com exceção da m-ésima equação de paridade
$r_{mn}(x)$	Probabilidade da m-ésima equação de paridade ser satisfeita dado todos os possíveis valores de bits codificados c
$f_n^{(x)}$	Probabilidade de que o n-ésimo bit recebido seja x
$g_n^{(x)}$	Probabilidade de que o n-ésimo bit recebido seja x
$g_i^{(x)}$	Probabilidade de que o i-ésimo bit recebido seja x
f	Vetor de probabilidades dos bits recebidos
Q	Matriz com as probabilidades $g_i^{(x)}$ e posteriormente as probabilidades $q_{mn}(x)$
R	Matriz com as probabilidades $r_{mn}(x)$
β_{mn}	Constante de normalização
Q'	Vetor das probabilidades pseudo posteriori
$\hat{\mathbf{c}}$	Estimativa de uma palavra código gerada pelo decodificador

SUMÁRIO

1 INTRODUÇÃO 16

2 SISTEMAS DE COMUNICAÇÕES ÓPTICAS 19
2.1 Contextualização histórica 19
2.2 Fundamentos da Recepção Coerente 21
2.2.1 Princípios Básicos do Receptor Coerente Digital 24
2.3 Formatos de Modulação 25
2.4 FEC em Comunicações Ópticas 27

3 TEORIA DA INFORMAÇÃO E CÓDIGOS 29
3.1 Teoria da Informação 29
3.2 Códigos de Bloco Lineares 33

4 CÓDIGOS LDPC 41
4.1 Definições 41
4.2 Construção de Códigos LDPC Regulares 45
4.2.1 Construção de Gallager 45
4.2.2 Construção BIBD 47
4.3 Decodificação 50
4.3.1 *Bit-Flipping* 50
4.3.2 Algoritmo Soma-Produto 52

5 DESEMPENHO DE CÓDIGOS LDPC 61

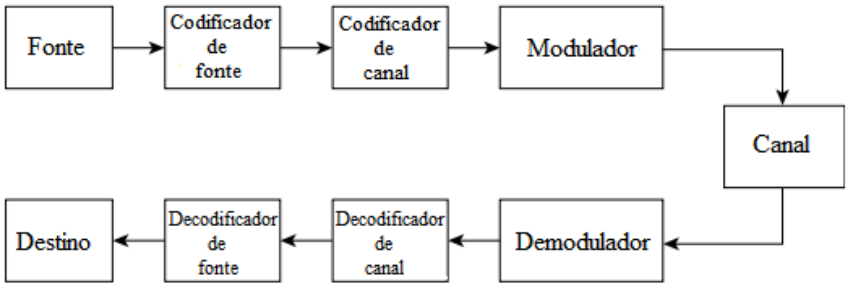
6 CONCLUSÃO 68

REFERÊNCIAS 70

1 INTRODUÇÃO

A necessidade de comunicação sempre foi inerente ao ser humano e diversos meios já foram utilizados para realizar comunicações à distância, como por exemplo o telégrafo inventado no século XIX. Entretanto, foi no século seguinte que os alicerces teóricos permitiram grandes avanços na área, alicerces esses, fundamentados em dois grandes nomes: Harry Nyquist na década de 1920 e Claude E. Shannon na década de 1940. Nyquist explorou o comportamento das transmissões elétricas e os descreveu matematicamente, publicando um artigo notável [NYQUIST 1924], que foi complementado com outro artigo [NYQUIST 1928] e resultou no Teorema da Amostragem. Foi então no ano de 1948 que Shannon publicou seu artigo que colaborou para grandes avanços tecnológicos em telecomunicações *"A Mathematical Theory of Communication"*, em que demonstrou ser possível realizar a transmissão de informação por um canal ruidoso, desde que uma técnica de codificação seja implementada e a taxa de transmissão seja inferior ou igual a capacidade do canal [SHANNON 1948]. Foi nesse mesmo estudo que Shannon definiu o modelo de um sistema de transmissão de informação, uma variação deste modelo é visto na Figura 1

Figura 1 – Diagrama de blocos do sistema de comunicação digital básico



fonte: [RYAN e LIN 2009]

- *Fonte* (Origem): é a fonte que gera a informação. Essa informação pode ser analógica como voz por exemplo, e ser digitalizada, ou pode ser de origem digital como arquivos.
- *Codificador/decodificador de fonte*: usualmente considera-se que a informação é gerada por meio de bits e o codificador de fonte transforma uma informação em uma sequência de bits a fim de inserir uma representação eficiente para fonte. A codificação de fonte é comumente conhecida como compressão de dados, e o decodificador de fonte realiza o processo inverso.
- *Codificador/decodificador de canal*: tem a função de proteger a informação que será enviada, evitando que a informação seja corrompida por intermédio de ruído, interferência e/ou distorção. A codificação de canal é feita por meio da inserção de bits de redundância à informação a fim de prover imunidade as possíveis deficiências do canal. O decodificador de canal por sua vez, tem a função de recuperar a informação que foi transmitida.
- *Modulador e demodulador*: efetua a modificação do sinal de uma portadora de acordo com o sinal modulante (que contém a informação), a escolha da modulação utilizada depende do

tipo de sistema empregado. A modulação altera parâmetros como amplitude, fase, polarização, frequência, etc e essas alterações carregam a informação que se quer transmitir. O demodulador irá fazer o processo inverso.

- *Canal*: é o meio físico pelo qual o sinal é transmitido, exemplos comuns de canais são: ar, cabo coaxial, fibra óptica, etc. Canais de comunicações podem ser responsáveis pela inserção de ruídos, interferência ou outros efeitos como atenuação que é a diminuição de potência do sinal. Canais podem ser modelados probabilisticamente, um exemplo disso é o canal com ruído aditivo gaussiano e branco (*Additive White Gaussian Noise*, AWGN) [RYAN e LIN 2009].

A partir de então, Shannon dava início a duas áreas novas de pesquisa, a teoria da informação e a teoria de códigos, foi então em 1950 que surgiram os códigos Hamming, desenvolvido por Richard Wesley Hamming são um tipo de código corretor de erro, classificado como código de bloco linear, pois atuam sobre a definição de uma palavra de tamanho fixo e a inserção de um valor também fixo de bits de paridade, o código de Hamming proposto inicialmente era do tipo $(7, 4)$, ou seja, continha 4 bits de informação e 3 bits de paridade, este código detectava até dois bits errados e corrigia um [HAMMING 1950]. Dentre os códigos de bloco lineares se destacam os códigos de verificação de baixa densidade (*Low Density Parity Check*, LDPC), pois ao serem cuidadosamente projetados sob um canal AWGN podem apresentar uma capacidade muito próximo ao limite de Shannon [SAE-YOUNG CHUNG et al. 2001]. Estes códigos são caracterizados por uma matriz controle de paridade esparsa, ou seja, com baixa densidade de elementos não-nulos em comparação com as dimensões da matriz e foram propostos pela primeira vez por Gallager [GALLAGER 1962]. Entretanto devido a baixa capacidade computacional da época em que o código LDPC foi proposto, ele só voltou a ser tema de pesquisa nos anos 90, quando Mackay e Neal revisitaram o artigo de Gallager e propuseram uma nova maneira de obter uma matriz de paridade esparsa [MACKAY e NEAL 1996]. Desde então devido ao seu alto desempenho, códigos LDPC passaram a ser amplamente estudados e aplicados em diversos sistemas, em particular em comunicações ópticas.

O estudo de código corretores de erros aplicados a comunicações ópticas iniciou-se em 1988, a partir de então diversos estudos sobre correção antecipada de erros (*Foward Error Correction*, FEC) foram propostos [GROVER 1988, PUC et al. 1999, RAMANUJAM et al. 2000]. Atualmente, os códigos FEC de terceira geração utilizam decodificação baseados em *soft-decision*, isto é, avalia a distribuição probabilística dos possíveis valores dos bits e apresentam ganhos elevados quando comparados aos códigos das gerações anteriores [TYCHOPOULOS, KOUFOPAVLOU e TOMKOS 2006]. Dentre esses códigos, o código LDPC tem se destacado em sistemas de comunicações ópticas que atuam com altas taxas de transmissão, como por exemplo em sistemas de $100Gb/s$, e além. Do mesmo modo, os códigos LDPC apresentam grande importância no que se diz respeito a transmissão de longo alcance [MIZUOCHI e MIYATA 2008, ARABACI et al. 2010, KAMIYA e SHIOIRI 2010, SONG et al. 2012, CUSHON, LARSSON-EDEFORS e ANDREKSON 2017, VASIC e DJORDJEVIC 2002].

Neste trabalho iremos avaliar por meio de simulação implementadas no software Python, o desempenho em termos da BER (*Bit Error Rate*) dos códigos LDPC regulares em um canal AWGN utilizando decodificação iterativa. Para verificar maneiras de se obter uma boa matriz controle de paridade, testamos o método proposto por Gallager que foi o primeiro a propor os códigos LDPC, e um

método que baseia-se em design combinatório denominado projeto de bloco incompleto balanceado (*Balanced Incomplete Block Design*, BIBD), que por meio de uma matriz de incidência entre pontos e blocos, facilita a obtenção da matriz verificadora de paridade [GALLAGER 1962, MOREIRA e 1 2019]. A decodificação foi implementada utilizando o algoritmo Soma-Produto (*Sum-Product Algorithm*, SPA), que é um método de decodificação iterativa baseado em *soft-decision*, onde avaliamos a influência do número de iterações utilizadas na decodificação [CARRASCO e JOHNSTON 2008]. Também foi averiguado qual o impacto em termos da BER ao utilizarmos dois códigos com tamanhos de bloco distintos. Além de apresentar qual é o ganho de código (*Coding Gain*, CG) para cada iteração aplicada ao decodificador [TYCHOPOULOS, KOUFOPAVLOU e TOMKOS 2006].

O presente trabalho encontra-se organizado conforme segue: no Capítulo 2 será apresentado os conceitos básicos de sistemas de comunicações ópticas, abrangendo os fundamentos da detecção coerente, receptores coerentes digitais, formatos de modulação bem como a utilização da FEC para sistemas ópticos. No Capítulo 3 uma breve revisão sobre teoria da informação e códigos de blocos lineares. No capítulo 4 descreveremos a teoria dos códigos LDPC, desde a construção do código por meio de diversas técnicas, assim como métodos de decodificação iterativa. No capítulo 5 serão descritos as métricas a serem analisadas, as simulações computacionais implementadas e os resultados obtidos. No capítulo 6, serão apresentadas as conclusões obtidas deste trabalho.

2 SISTEMAS DE COMUNICAÇÕES ÓPTICAS

Neste capítulo iremos realizar uma breve revisão dos conceitos básicos, relativos a sistemas de comunicações ópticas coerentes a fim de contextualizar o cenário utilizado para o desenvolvimento deste trabalho. Na Seção 2.1 é descrito rapidamente o contexto histórico da evolução das comunicações ópticas. Na Seção 2.2 é apresentado os fundamentos da detecção coerente, apresentando as definições básicas e suas caracterizações. Na Seção 2.3 foi explorado o modulador Mach-Zehnder (*Mach-Zehnder Modulator*, MZM), com uma breve explicação sobre seu funcionamento e os seu possíveis formatos de modulação. Finalmente na Seção 2.4 é descrito de maneira simplificada a implementação da FEC, bem como a evolução deste método ao longo dos anos. As definições e conceitos utilizados neste capítulo, podem ser encontrados em [GROVER 1988, KIKUCHI 2006, MORI et al. 2006, TYCHOPOULOS, KOUFOPAVLOU e TOMKOS 2006, KAMINOW, LI e WILLNER 2008, SMITH e KSCHISCHANG 2010, KIKUCHI 2016, BLAUNSTEIN et al. 2020, KUMAR e DEEN 2014]

2.1 CONTEXTUALIZAÇÃO HISTÓRICA

As comunicações ópticas ganharam um grande impulso em 1960 devido a fabricação do laser por Theodore Maiman. Os testes iniciais demonstraram a possibilidade de enviar informações utilizando estritamente a modulação de intensidade através do ar, podendo atingir muitos quilômetros de distância. Entretanto, verificou-se que a transmissão perdia confiabilidade devido ao impacto que nuvens, chuvas e nevoeiros causavam ao sinal [SENIOR 2009]. Deste modo, iniciou-se uma busca em conseguir um guia de onda óptico que possibilitasse a transmissão com o mínimo de perdas possível [DECUSATIS 2008]. Apesar do laser apresentar certas desvantagens para comunicações através do ar, foi grande motivador para a pesquisa de um guia de onda que pudesse ser utilizado como canal de comunicação. Foi então em 1966 que a primeira fibra óptica de vidro foi proposta por [WERTS 1966], todavia, as primeiras fibras ópticas apresentavam grande atenuação em torno de $1000dB/Km$. No entanto, no período de 10 anos, as perdas das fibras diminuíram para cerca de $5dB/Km$. Os modelos inicias de comunicações ópticas enviavam informações apenas por meio da modulação direta [SENIOR 2009, KIKUCHI 2016].

Definição 2.1.1 *Define-se como* ***modulação direta*** *quando a informação é modulada sobre a corrente de controle do laser, ou seja, a informação será representada de acordo com a intensidade da luz transmitida.*

Na modulação direta, apenas lasers semicondutores são aplicados na prática em decorrência de requerimentos de largura de banda e eficiência, por serem polarizados por uma corrente elétrica, toda e qualquer variação nessa corrente implicará em uma variação de potência óptica na saída. A modulação direta é a maneira mais fácil de inserir informação em uma portadora, um exemplo muito comum deste tipo de modulação é chamada de OOK (*On-Off-Keying*) que é um tipo específico da modulação ASK (*Amplitude shift keying*), onde a portadora ligada "*on*"representa o sinal lógico 1 e a portadora desligada

"*off*"representa o sinal lógico zero. Entretanto, para altas taxas de transmissão ou ainda grandes distâncias a modulação direta apresenta uma grande desvantagem, pois apresenta uma modulação de fase residual que acompanha a modulação de interesse, esta modulação de fase residual é chamada de *chirp*,que causa o efeito de alargamento do sinal óptico podendo gerar dificuldades como a aplicação da modulação direta em alguns sistemas [KAMINOW, LI e WILLNER 2008].

Os receptores para este tipo de modulação realizam o que define-se como detecção direta, pois apenas são capazes de identificar variações na intensidade luminosa do sinal. Considerando um sinal modulado OOK, delimita-se uma potência de limiar no receptor, onde qualquer sinal acima desse limiar é recebido como sendo um bit "1" e abaixo desse limiar como bit "0", em casos onde o ruído afete a potência do bit essa decisão pode ser afetada, diminuindo então o desempenho do sistema.

Definição 2.1.2 *Um **receptor coerente** é um receptor em que o sinal recebido e misturado com a luz emitida por um laser que atua como oscilador local e ao contrário dos receptores baseados em detecção direta, os receptores coerentes podem recuperar informação tanto de amplitude como de fase e polarização. E podem ser classificados como homódinos, heteródinos ou intradinos.*

A primeira proposta de comunicação óptica coerente utilizando detecção heteródina foi proposto por DeLange em 1970 [DELANGE 1970]. Já na década de 1980 os sistemas de comunicações ópticas coerentes começaram a ser amplamente estudados, apresentando características vantajosas em comparação com sistemas que utilizavam modulação de intensidade e detecção direta, como por exemplo uma melhora na sensibilidade do receptor limitado ao ruído balístico (*Shot-Noise*) que supera o ruído térmico quando aplicado uma potência suficiente ao oscilador local (*Local Oscillator*, LO), aumento da capacidade de transmissão ao utilizar diferentes formatos de modulação com uma maior eficiência espectral como por exemplo QPSK (*Quadature Phase Shift Keying*), apesar destas vantagens, após o surgimento dos amplificadores de fibra dopada com érbio (*Erbium Doped Fiber Amplifier*, EDFA) a sensibilidade do receptor limitado ao ruído balístico tornou-se menos significante, isto ocorre devido ao ruído de emissão espontânea amplificada (*Amplified Spontaneous Emission*, ASE) inserido pelo amplificador. Além do mais, considerando a maior complexidade de receptores coerentes e atrelado a utilização do EDFA como pré-amplificador em sistemas com detecção direta, fizeram com que ocorresse uma pausa de quase 20 anos nas pesquisas e desenvolvimento de sistemas ópticos coerentes. Os sistemas baseados em EDFA permitiu a implementação de enlaces que utilizam multiplexagem por divisão do comprimentos de onda (*Wavelength Division Multiplexing*, WDM) a fim de aumentar a capacidade de transmissão em uma única fibra, este tipo de sistema foi largamente utilizado devido a sua simplicidade e custo baixo. A técnica WDM gerou um aumento de 1000 vezes na capacidade de transmissão [KAMINOW, LI e WILLNER 2008].

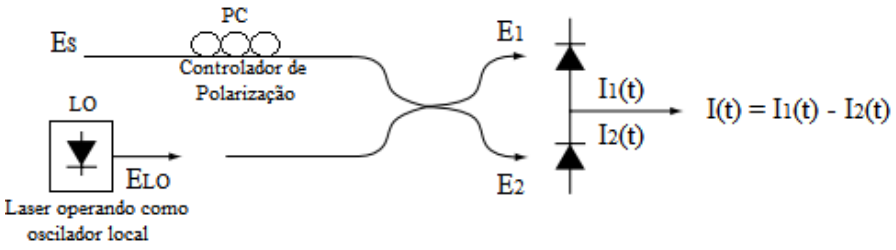
Os sistemas WDM foram progressivamente atualizados para utilizar canais com maior capacidade. Porém, quando a capacidade por canal aumentou até atingir $100Gbps$, a dispersão por modo de polarização (*Polarization Mode Dispersion*, PMD) emergiu como um impedimento limitante. Neste cenário, formatos de modulação com uma eficiência espectral mais alta foram propostos [KAHN e HO 2004]. Foi um experimento com a modulação e demodulação em QPSK que reiniciou as pesquisas sobre sistemas coerentes, ao carregar dois bits por símbolo, faz com que a taxa de bits dobre enquanto

mantém a taxa de símbolo, ou ainda manter a taxa de bits enquanto diminui a largura espectral pela metade [GRIFFIN e CARTER 2002]. A modulação em fase e quadratura foi realizado utilizando um arranjo em paralelo de moduladores do tipo MZM que permite que as componentes da portadora óptica sejam moduladas de maneira independente. Desta forma, para poder explorar tanto a diversidade de fase como polarização, a comunidade retomou a ideia dos sistemas coerentes. Porém, os mais de 20 anos transcorridos desde o estudo dos primeiros sistemas coerentes, permitiram um avanço significativo na microeletrônica, em particular, no processador digital de sinal (*Digital Signal Processing*, DSP), que possibilitou manipular o sinal elétrico em um núcleo DSP a fim de recuperar informações de fase e quadratura e permitiram a compensação do ruído de fase e das flutuações de polarização. Vale ressaltar que uma vez introduzido o DSP de alto desempenho, outros impedimentos além do ruído de fase e das flutuações de polarização também foram compensados no domínio digital. Em particular, nos sistemas ópticos coerentes digitais a dispersão cromática e a dispersão por modo de [KIKUCHI 2006].

2.2 FUNDAMENTOS DA RECEPÇÃO COERENTE

Essencialmente a recepção coerente baseia-se no resultado da interferência entre os campos elétricos do sinal modulado e do sinal do oscilador local. O esquema de recepção coerente exemplificado na Figura 2 demonstra a utilização da detecção balanceada a fim de suprimir a componente da corrente direta (*Direct Current*, DC) maximizando assim a componente de batimento entre o sinal e o LO. Assume-se que o estado de polarização (*State of Polarization*, SOP) tanto do sinal quanto do LO são alinhados, por meio de um controlador de polarização (*Polarization Controller*, PC).

Figura 2 – Configuração de um receptor coerente



fonte: Próprio Autor

Na Figura 2, E_s e E_{LO} representam os campos elétricos do sinal modulado e do LO respectivamente, E_1 e E_2 são os campos elétricos superior e inferior resultante da copolarização entre E_s e E_{LO} que irão incidir nos fotodiodos resultando nas fotocorrentes $I_1(t)$ e $I_2(t)$, a resultante entre as duas fotocorrentes na saída do receptor balanceado será

$$I(t) = I_1(t) - I_2(t) = 2R\sqrt{P_s(t)P_{LO}}\cos\{\omega_{IF}t + \theta_{sig}(t) - \theta_{LO}(t)\}, \tag{2.1}$$

onde R é a responsividade do fotodiodo, P_s e P_{LO} as potências do sinal e do oscilador local, ω_{IF} é a frequência angular da frequência intermediária que é dada por $\omega_{IF} = |\omega_s - \omega_{LO}|$ e as fases do sinal e do oscilador local, são respectivamente $\theta_{sig}(t)$ e $\theta_{LO}(t)$. Na Equação 2.1, o ruído aditivo dos fotodiodos é desconsiderado pois assume-se que é muito menor que a contribuição do ruído. Dado que

a potência do oscilador local P_{LO} é constante, o sinal LO será responsável pela inserção do ruído de fase que varia aleatoriamente com o tempo e por realizar a conversão de frequências [KIKUCHI 2016].

Conforme citado anteriormente, receptores coerentes podem ser definidos como homódinos, heteródinos ou intradinos, essa caracterização é realizada a partir do parâmetro ω_{IF}.

Determina-se a detecção heteródina quando $|\omega_{IF}| \gg 2\pi/T$, sendo $1/T$ a taxa de símbolos que define a largura de banda da modulação da portadora óptica. Assim sendo, o campo elétrico do sinal é convertido para o sinal IF (*Intermediate Frequency*) incluindo as informações de amplitude e fase. Ao analisarmos um sinal modulado em PSK (*Phase Shift Keying*), a potência do sinal P_s será constante, e a fase $\theta_{sig}(t)$ será a soma da modulação de fase $\theta_s(t)$ e do ruido de fase $\theta_{sn}(t)$. Logo, a saída do receptor será

$$I(t) = 2R\sqrt{P_s P_{LO}}\cos\{\omega_{IF}t + \theta_s(t) + \theta_n(t)\}. \tag{2.2}$$

A amplitude complexa dessa corrente pode ser escrita como sendo

$$I_c(t) = 2R\sqrt{P_s P_{LO}}\exp[j\{\theta_s(t) + \theta_n(t)\}], \tag{2.3}$$

observa-se que salvo a inserção do ruído de fase, a amplitude complexa da corrente de saída é correspondente a do sinal óptico. Apesar do ruído de fase total variar de acordo com o tempo, é possível utilizar a demodulação elétrica síncrona para decodificar o símbolo θ_s estimando o ruído de fase.

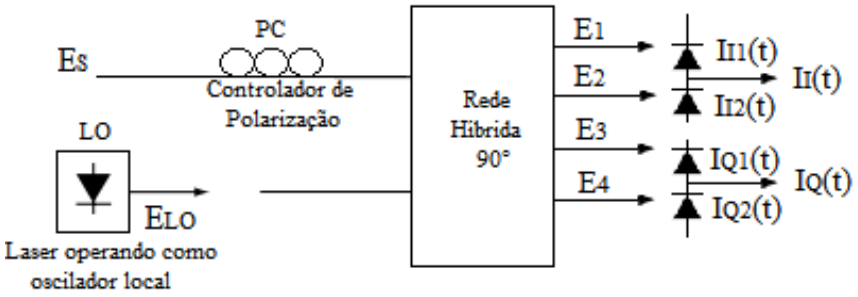
Em contrapartida, os detectores homódinos são definidos quando $\omega_s = \omega_{LO}$, ou seja, $\omega_{IF} = 0$, logo a corrente de saída do fotodiodo será

$$I(t) = 2R\sqrt{P_s P_{LO}}\cos\{\theta_{sig}(t) - \theta_{LO}(t)\}. \tag{2.4}$$

Para que o símbolo recebido seja interpretado de maneira correta, a fase do oscilador local θ_{LO} deve identificar o ruído de fase do transmissor θ_{sn} de maneira que $\theta_n = 0$, esta ação pode ser realizada por meio de um OPPL (*Optical Phase-Locked Loop*). Entretanto, a inserção de um OPPL não é de fácil execução e acaba por elevar a complexidade do sistema, além de que como a 2.4 apenas nos dá a componente cosseno que está atrelado apenas a fase do LO, a componente seno que carrega a informação da quadratura não pode ser detectada, impossibilitando a total extração de informações da amplitude complexa do sinal.

Para que um receptor coerente com detecção homódina possa recuperar toda a informação da fase e quadratura de um sinal, é possível utilizar uma rede híbrida de $90°$ conforme Figura 3, este tipo de receptor é denominado "receptor homódino de diversidade de fase",

Figura 3 – Configuração do receptor homódino de diversidade de fase



fonte: Próprio Autor

A partir deste circuito é possível obter na saída dos fotodetectores balanceados, duas correntes distintas, que serão as responsáveis por fornecer as informações das componentes em fase (I) e quadratura (Q) do sinal. Para uma modulação PSK, as correntes são dadas por

$$I_I(t) = R\sqrt{P_s P_{LO}}\cos\{\theta_s(t) + \theta_n(t)\} \tag{2.5}$$

e

$$I_Q(t) = R\sqrt{P_s P_{LO}}\sin\{\theta_s(t) + \theta_n(t)\}. \tag{2.6}$$

Sendo assim, a partir das equações 2.5 e 2.6, a amplitude complexa da corrente será

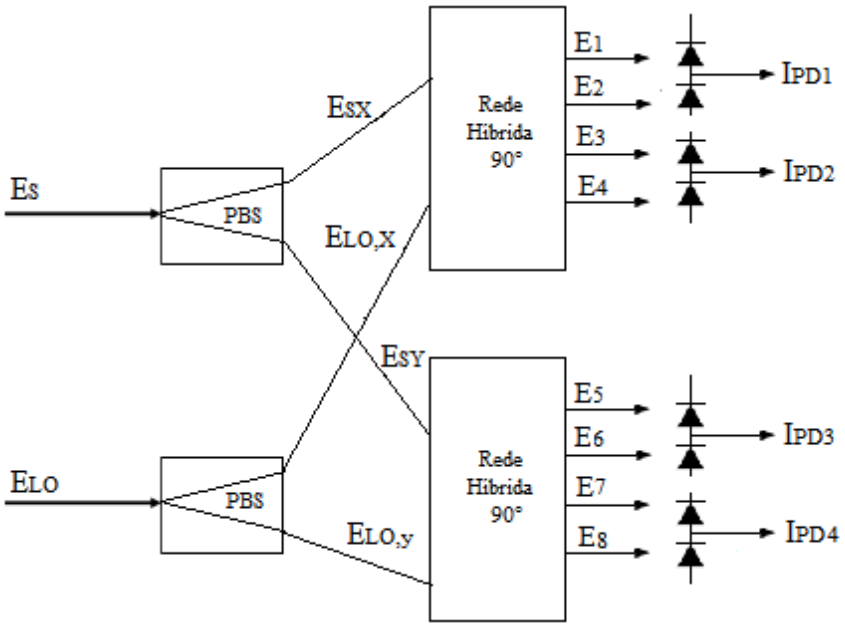
$$I_c(t) = I_I(t) + jI_Q(t) = R\sqrt{P_s P_{LO}}\exp[j\{\theta_s(t) + \theta_n(t)\}], \tag{2.7}$$

novamente o resultado está relacionado com a amplitude complexa do sinal óptico, exceto pelo incremento do ruído de fase $\theta_n(t)$. Apesar do ruído de fase ser aleatório e variante no tempo, pode ser estimado a fim de recuperar a informação de fase do sinal $\theta_s(t)$ aplicando a técnica de DSP no sinal obtido pela Equação 2.7.

Os receptores heteródinos e homódinos de diversidade de fase, podem eventualmente apresentar uma maneira análoga para recuperar de maneira completa as informações sobre a amplitude complexa do sinal óptico conforme as equações 2.3 e 2.7. Entretanto, por apresentar a capacidade de recuperar essas informações em banda base, o receptor homódino de diversidade de fase apresenta maior vantagem, pois o receptor heteródino trabalha com valores de IF bastante altos [KAMINOW, LI e WILLNER 2008].

Até então, considerou-se que a polarização do sinal óptico e do LO estavam alinhadas. Entretanto, em decorrência de birrefringências na fibra de transmissão, existem variações aleatórias na polarização do sinal óptico transmitido. A polarização é de extrema importância pois em casos em que o SOP do sinal transmitido é ortogonal ao sinal do LO, não haverá qualquer informação na saída balanceada. Deste modo, a fim de mitigar esses efeitos, aplica-se um receptor homódino com diversidade de fase e polarização conforme Figura 4

Figura 4 – Configuração do receptor homódino de diversidade de polarização/fase



fonte: Próprio Autor

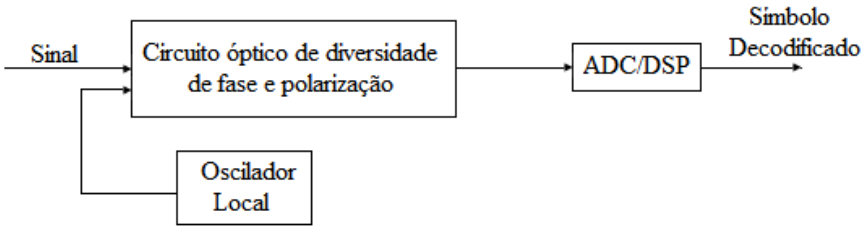
Este receptor é obtido por meio da combinação de dois receptores homódinos com diversidade de fase, de maneira a criar a diversidade de polarização, o sinal óptico transmitido chega com uma polarização arbitrária e é separado em duas componentes por um divisor de feixe de polarização (PBS - *Polarization Beam Splitter*), que serão detectados com o mesmo LO para assim recuperar a informação completa do sinal. Ainda que o sistema seja multiplexado por polarização, é possível recuperar as características da polarização de quatro saídas utilizando o DSP [KIKUCHI 2016].

2.2.1 Princípios Básicos do Receptor Coerente Digital

Um receptor coerente digital, é assim denominado pois utiliza-se de circuitos DSP na saída do receptor coerente homódino com diversidade de polarização/fase, para recuperar as informações de maneira estável, mesmo que o sinal apresente flutuações de fase da portadora, bem como do SOP.

Conforme exemplificado na Figura 5, o sinal recebido é detectado utilizando um receptor coerente homódino com diversidade de polarização/fase conforme descrito anteriormente, a saída desse receptor carrega consigo as informações completas de amplitude, fase e SOP do sinal óptico. Este sinal é convertido em dados digitais por intermédio de um conversor analógico para digital (*Analog to Digital Converter*, ADC) e processados pelo circuito DSP.

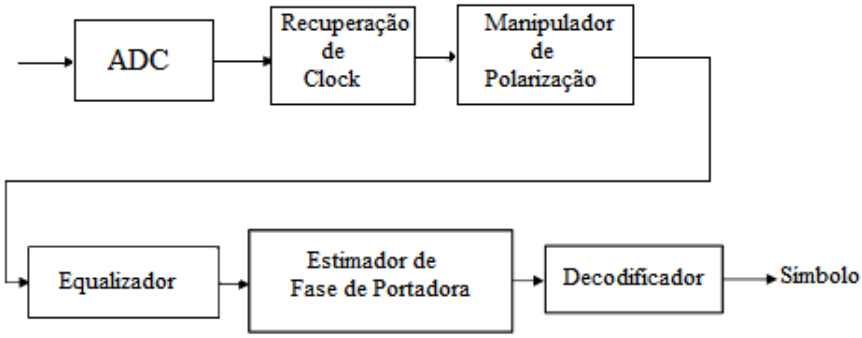
Figura 5 – Configuração básica de um receptor coerente digital



fonte: Próprio Autor

Para que haja a recuperação total das informações de um sinal modulado, o circuito DSP normalmente aplica uma sequência definida de operações ao sinal, conforme exemplificado na Figura 6.

Figura 6 – Diagrama de blocos do processamento digital de sinais



fonte: Próprio Autor

Inicialmente, conforme descrito acima ocorre a amostragem e digitalização do sinal analógico por meio do ADC, em seguida ocorre a recuperação do *clock* e a reamostragem do sinal, logo após o sinal reamostrado passa por um alinhamento de polarização, ou seja, a combinação de polarização e depois pelo equalizador onde a dispersão de velocidade de grupo (GVD - *Group-Velocity Dispersion*) é compensada, então a fase da portadora é estimada e o símbolo é decodificado. A decodificação FEC é de extrema importância neste cenário e será discutida mais à frente [KAMINOW, LI e WILLNER 2008].

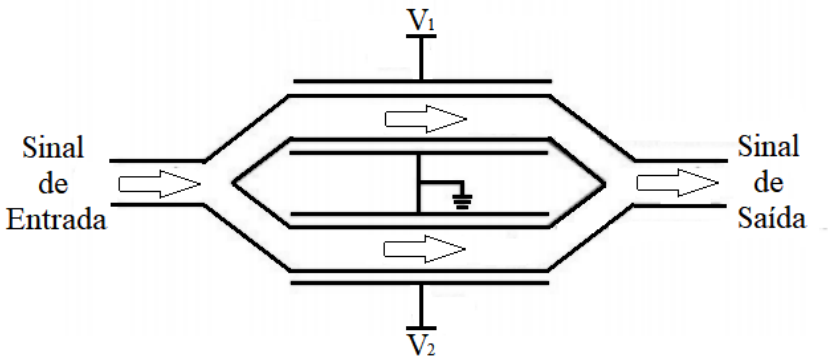
2.3 FORMATOS DE MODULAÇÃO

Em comunicações ópticas a modulação é aplicada visando o aprimoramento da eficiência espectral [KAHN e HO 2004]. Sistemas que utilizam receptores coerentes, são capazes de trabalhar com formatos de modulação multinível, sendo elas: modulação em amplitude (*Amplitude Shift Keying, ASK*, modulação em fase (*Phase Shift Keying*, PSK), modulação em amplitude e fase (*Quadrature Amplitude Modulation*, QAM) e modulação em frequência (*Frequency Shift Keying*, FSK) [MORI et al. 2006].

Um dos tipos de moduladores mais utilizados em comunicações ópticas é o MZM, que aplica o conceito de interferências parciais. Este modulador é estruturado conforme Figura 7, onde a luz

que entra no dispositivo é dividida e percorre por dois guidas de ondas, estes guias são diretamente influenciados pela aplicação de tensões externas V_1 e V_2 que causam diferença de fase entre os sinais que percorrem por cada caminho. Quando não há tensão aplicada, os campos nos dois guias de onda experimentam um deslocamento de fase idênticos, gerando na saída do acoplador uma interferência construtiva, quando a diferença de fase entre os dois campos é igual a π não haverá luz transmitida devido a interferência totalmente destrutiva entre os dois sinais [KAMINOW, LI e WILLNER 2008].

Figura 7 – Modulador de Mach-Zehnder

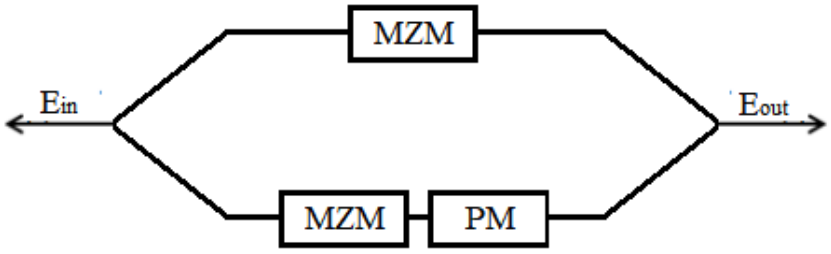


fonte: Próprio Autor

Sendo assim, a sequência de bits convertidos por meio da modulação PAM (*Pulse Amplitude Modulation*) ao ser aplicada ao modulador implicará em uma réplica óptica dessa mesma sequência. Usualmente os MZMs são fabricados a partir do niobato de lítio $(LiNbO_3)$, apresentam banda de operação óptica extensa, maior taxa de extinção (cerca de 20 dB) e menor perda de inserção quando comparado aos moduladores de eletro-absorção (*Electroabsorption modulator*, EAM).

O MZM foi amplamente aplicado pois entre suas vantagens, permite a modulação em fase e quadratira (*In-Phase and Quadrature*, IQ), que consiste na combinação entre dois MZM e um modulador de fase (*Phase Modulator*, PM) conforme Figura 8. Nessa configuração os dois MZM atuam como moduladores de amplitude, e o PM promove uma para defasar o sinal em $90°$.

Figura 8 – Modulador de fase e quadratura IQ

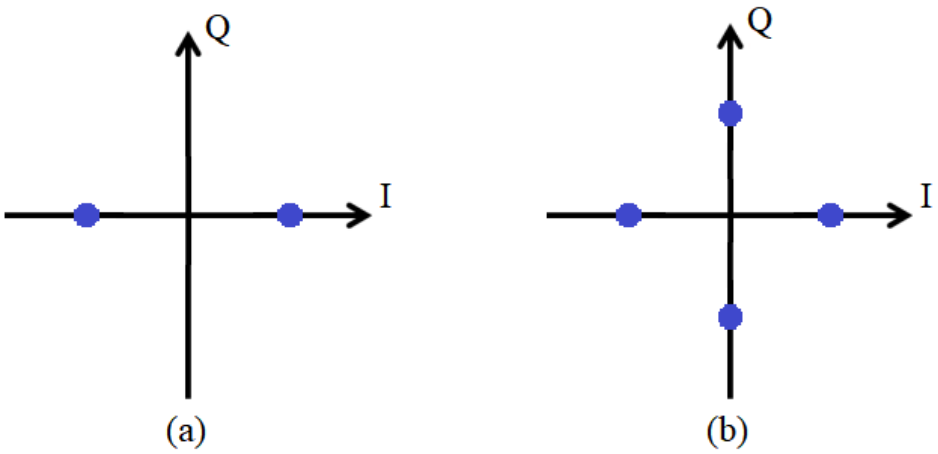


fonte: Próprio Autor

A representação de uma modulação IQ pode ser feita por meio de um diagrama de constelação, conforme mostra a Figura 9. O formato de modulação PSK mais simples é o BPSK (*Binary phase-shift keying*) onde a cada transição de um bit 0 para um bit 1 ocorre uma mudança de fase da portadora em

180° em relação ao ângulo anterior Figura 9-(a), uma outra variação deste mesmo tipo de modulação é o QPSK, onde a variação de fase se da em 90° Figura 9-(b), sendo possível assim representar dois bits por símbolo, aumentando a taxa da transmissão e mantendo a mesma probabilidade de erro do BPSK [KUMAR e DEEN 2014].

Figura 9 – (a) BPSK (b) QPSK



fonte: Próprio Autor

2.4 FEC EM COMUNICAÇÕES ÓPTICAS

A codificação de controle de erros, é um dos elementos essenciais em sistemas de comunicações ópticas modernas. A implementação do FEC consiste na inserção de bits redundantes na mensagem enviada, para que ela possa ser recuperada no receptor. Existem diversos tipos de FEC como por exemplo os códigos convolucionais e os códigos de bloco, sendo estes últimos, melhor explorado na Seção 3.2.

O desempenho FEC é caracterizado por meio da taxa de erro de bit (*Bit Error Rate*, BER) e avaliada por meio do ganho de codificação (*Coding Gain*, CG), que é a medida da diferença entre os níveis de SNR entre o sistema não codificado e o sistema codificado necessário para atingir os mesmos níveis de taxa de erro de bit (BER).

Considerando uma modulação BPSK ou QPSK e canal AWGN, a BER em função da SNR será

$$BER = \frac{1}{2}\,\mathrm{erfc}\,\sqrt{SNR}. \tag{2.8}$$

O cálculo de CG será

$$CG = 20\log_{10}\left[\frac{\mathrm{erfc}^{-1}(2BER_{out})}{\mathrm{erfc}^{-1}(2BER_{in})}\right](dB), \tag{2.9}$$

onde, BER_{out} é a BER da curva da sequência de bits à qual a FEC foi aplicada, e BER_{in} a BER da curva sem codificação, o resultado nos traz o ganho em dB associado à codificação aplicada [TYCHOPOULOS, KOUFOPAVLOU e TOMKOS 2006].

Para levar em conta a taxa de código R_C em consideração, o ganho de codificação líquida (*Net Coding Gain*, NCG) pode ser calculado

$$NCG = CG + 10\log_{10}(R_C)(dB). \tag{2.10}$$

Em comunicações ópticas a primeira geração de FEC era baseado em código único, apresentava um ganho associado de 6 dB na SNR, sendo o código *Reed-Solomon*$(255, 239)$ com *hard-decision* o mais comum dessa geração, tornando-se código bem estabelecido e indicado para transmissões de longas distâncias conforme recomendação ITU-T G.709 e G.975 [SMITH e KSCHISCHANG 2010].

A segunda geração foi impulsionada pelo estabelecimento dos sitemas que utilizam multiplexagem densa em comprimento de onda (*Dense Wavelength Division Multiplexing*, DWDM) e utilizou a combinação de códigos concatenados com técnicas de intercalação, decodificação iterativa e *hard-decision*, o padrão ITU-T G.975.1 estabeleceu oito algoritmos de FEC, a maioria desses algoritmos apresentavam um ganho de cerca de 8 dB e suportam transmissões de longas distâncias de $40Gb/s$ a $100Gb/s$.

A terceira geração foi fortemente influenciada pela utilização de recepção coerente juntamente com a evolução dos circuitos integrados, possibilitando assim empregar códigos baseados em *soft-decidion*, estes código forneceram cerca de 11 dB de ganho, sendo muito aplicados em sistemas com longas distâncias e altas taxas além de $100Gb/s$. FEC baseado em *soft-decision* normalmente utilizam códigos turbo ou ainda códigos de verificação de baixa densidade (LDPC) [TYCHOPOULOS, KOUFOPAVLOU e TOMKOS 2006]. Sendo os códigos LDPC o objeto de estudo deste trabalho, sua teoria e conceitos são investigados no Capítulo 4.

3 TEORIA DA INFORMAÇÃO E CÓDIGOS

Teoria matemática da informação ou teoria da informação é uma área de estudo oriunda da matemática estatística e da teoria da probabilidade, desenvolvida para trabalhar com sistemas de comunicações, transmissão de dados, codificação, entre outros. O surgimento da teoria da informação teve um grande marco em 1948 com a publicação do artigo *"A Mathematical Theory of Communication"* escrito por Claude E. Shannon, este artigo foi de tamanha importância que Shannon passou a ser conhecido como o "pai da teoria da informação". Diversos conceitos foram apresentados no artigo em questão, sendo alguns deles: entropia, informação mútua, capacidade do canal, codificação do canal entre outros. A partir dos limites fundamentais descritos por Shannon, inúmeros estudos foram desenvolvidos ao longo dos anos a fim de atingir ou de se aproximar o máximo possível destes limites [SHANNON 1948, SAE-YOUNG CHUNG et al. 2001]. O conceito fundamental dos códigos corretores de erro, baseia-se na codificação de uma informação inicial adicionando bits de redundância, de tal forma que, após a transmissão e mesmo que haja algum tipo de distúrbio sobre o sinal, seja possível recuperar a mensagem original. Existem basicamente dois tipos de códigos, os códigos convolucionais e os códigos de bloco, sendo este último objeto do nosso estudo.. Neste capítulo iremos realizar uma breve revisão de conceitos que serão necessários para a elaboração deste trabalho. Na Seção 3.1 são apresentados conceitos como quantidade de informação, entropia, informação mútua e capacidade do canal, exemplificando dois canais comumente utilizados. Na Seção 3.2 serão discutidos os códigos de bloco lineares, apresentando conceitos básicos, exemplificando a sua codificação e decodificação. Os conceitos expostos nesse capítulo podem ser encontrados em [SHANNON 1948, SAE-YOUNG CHUNG et al. 2001, COVER e THOMAS 2006, RYAN e LIN 2009, HAYKIN 2001].

3.1 TEORIA DA INFORMAÇÃO

Inicialmente Shannon propôs uma definição matemática para informação, associando a probabilidade da ocorrência de um evento com a quantidade de informação que essa ocorrência carrega, por exemplo, se temos uma fonte de informação que envia apenas o símbolo A, nossa incerteza sobre o próximo símbolo que a ser recebido é nula, pois a probabilidade de ocorrer o evento é máxima. Porém se recebermos o símbolo B, automaticamente essa ocorrência nos traria grande informação devido à raridade do evento.

Definição 3.1.1 *Sendo $p(x_k)$ a probabilidade de uma variável aleatória $x = x_k$, a* ***quantidade de informação*** *de um evento é definido como*

$$I(x_k) = log_2\left(\frac{1}{p(x_k)}\right) = -log_2[p(x_k)]. \tag{3.1}$$

O cálculo da informação pode ser feito utilizando qualquer base no logaritmo, desde que seja utilizado a mesma base em todo o cálculo. Entretanto, a base do logaritmo define a *unidade* da

informação, ou seja, quando usada a base 2 a unidade é o *bit*, se for usada a base natural e a unidade será *nat* e se o cálculo for realizado na base 10 a informação estará sendo medida em *hatler*.

A partir da informação é possível calcular uma grandeza fundamental da Teoria da Informação, a entropia, que representa a imprevisibilidade de algo.

Definição 3.1.2 ***Entropia*** *pode ser definida como a média da quantidade de informação, ou seja, sendo uma fonte discreta X, a* ***entropia*** *pode ser descrita como*

$$\mathcal{H}(X) = -\sum p(x) log_2[p(x)], \tag{3.2}$$

onde $p(x)$ é a função massa de probabilidade.

Exemplo 3.1.1 *Seja um dado de 6 faces e X a variável aleatória que representa a face do dado em um lançamento. X pode assumir os seguintes valores*

$$X = \{1, 2, 3, 4, 5, 6\},$$

e a probabilidade de cada ocorrência é

$$p(x) = \left\{\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right\}.$$

Logo, a entropia associada ao lançamento do dado a partir da Equação 3.2 é:

$$\mathcal{H}(X) = -\sum_{0}^{6} p(x) log_2[p(x)] = -6\left(\frac{1}{6}\right) log_2\left(\frac{1}{6}\right) = 2.585 \textit{ bits}.$$

Exemplo 3.1.2 *Agora considere X sendo o lançamento de uma moeda honesta. Assim,*

$$X = \{0, 1\},$$

e

$$p(x) = \left\{\frac{1}{2}, \frac{1}{2}\right\}.$$

Então, a entropia relacionada a uma moeda honesta será:

$$\mathcal{H}(X) = -\sum_{0}^{2} p(x) log_2[p(x)] = -2\left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right) = 1 \textit{ bit}.$$

É possível observar por meio dos exemplos que a entropia máxima está associada a quantidade de símbolos, dado que os símbolos são equiprováveis.

Diretamente correlacionada com a entropia, temos também a informação mútua, que representa a diminuição da incerteza de uma variável devido a observação de outra.

Definição 3.1.3 *Dado duas variáveis aleatórias X e Y, sendo $p(x,y)$ a probabilidade conjunta e, $p(x)$ e $p(y)$ as probabilidades marginais, a **informação mútua** pode ser expressa conforme segue*

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y) log_2 \left[\frac{p(x,y)}{p(x)p(y)}\right]. \tag{3.3}$$

A partir da Equação 3.3, temos algumas propriedades da informação mútua.

1. $I(X;Y) = 0$ se, e somente se, X e Y forem variáveis aleatórias independentes;
2. $I(X;Y) \geq 0$;
3. $I(X;Y) = I(Y;X)$.

A informação mútua pode ser simplificada e ser escrita em função da entropia da seguinte forma

$$I(X;Y) = \mathcal{H}(X) - \mathcal{H}(X|Y) = \mathcal{H}(Y) - \mathcal{H}(Y|X). \tag{3.4}$$

A informação mútua é um poderoso parâmetro para verificar a eficiência do canal, pois reflete a confiabilidade da transmissão, ou seja, quanto maior for a dependência das variáveis, maior será a informação mútua caracterizando assim, uma transmissão confiável. Essas definições serão importantes para definir a capacidade de um canal, pois foi demonstrado por Shannon que todo canal de comunicação tem um limite de taxa medido em bits por segundo em que a informação pode ser transmitida de forma confiável, este limite ficou conhecido como *Limite de Shannon* [COVER e THOMAS 2006].

Definição 3.1.4 *Considerando as variáveis aleatórias X como a mensagem transmitida e Y como a mensagem recebida, a **capacidade do canal** é o máximo da informação mútua considerando todas as possíveis distribuições de probabilidade $p(x)$, ou seja*

$$C = \max_{p(x)} I(X,Y). \tag{3.5}$$

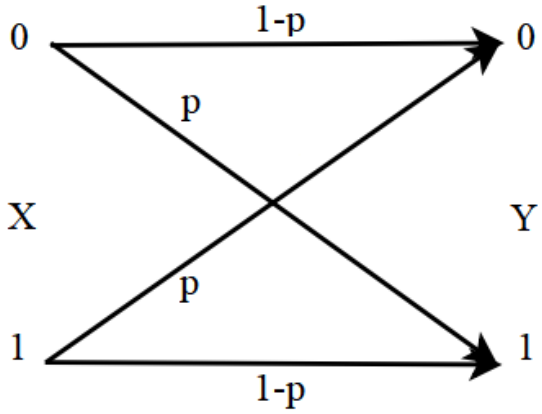
Em seguida iremos utilizar dois tipos de canais muito comuns, no caso discreto e no caso contínuo respectivamente, para exemplificar o cálculo da capacidade do canal.

Exemplo 3.1.3 *Na Figura 10 está representado o anal binário simétrico BSC (Binary Symmetric Channel). Neste tipo de canal a fonte enviará apenas um símbolo por vez $(0$ ou $1)$, e a probabilidade de erro será igual independente do bit transmitido, isto significa que, $P[Y = 1|X = 0] = P[Y = 0|X = 1] = p$*

Para calcular a capacidade deste canal é necessário encontrar a informação mútua máxima, para isso a entropia também deve ser máxima. Visto que $P[Y = 1|X = 1] = P[Y = 0|X = 0] = 1 - p$, aplicando a Equação 3.3 temos a capacidade do canal como

$$C = 1 + plog_2(p) + (1-p)log_2(1-p). \tag{3.6}$$

Figura 10 – Canal BSC



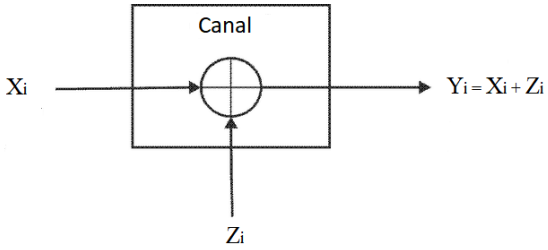
fonte: Próprio Autor

Em termos da entropia dada na Equação 3.2, a capacidade do canal pode ser simplificada para:

$$C = 1 - \mathcal{H}(p).$$

Exemplo 3.1.4 *Na Figura 11 está representado um canal muito importante no caso contínuo, o canal gaussiano. Neste caso, um sinal X_i no processo de transmissão sofre uma distorção por meio de um ruído Z_i, este ruído é aditivo gaussiano e branco (Additive White Gaussian Noise - AWGN) de média 0 e variância σ^2*

Figura 11 – Canal AWGN



fonte: Próprio Autor

Para que a capacidade deste canal não ser infinita, uma restrição em uma transmissão acontece em relação a potência P, desta forma o cálculo da capacidade do canal será

$$C = \max_{p(x)]\leqslant P} \frac{1}{2} \log_2 \left(1 + \frac{P}{\sigma^2}\right), \tag{3.7}$$

onde a fração $\frac{P}{\sigma^2}$ representa a relação entre a potência do sinal e a potência do ruído (Signal-to-Noise Ratio - SNR) e C é dado em unidade de informação por unidade de tempo.

Além da limitação em potência, o canal ainda pode ser limitado em banda, neste caso a capacidade do canal será

$$C = Blog_2 \left(1 + \frac{P}{\sigma^2}\right) = Blog_2(1 + SNR), \tag{3.8}$$

onde a banda B do canal é dada em Hz, a SNR é representado em watts resultando na capacidade C do canal em $bits/s$.

Shannon demonstrou em seu estudo que é possível acrescentar uma redundância ao sinal enviado a fim de diminuir a taxa de erros na recepção do sinal, provando que desde que a capacidade do canal não seja ultrapassada, é possível diminuir a taxa de erros tanto quando se queira [SHANNON 1948].

Teorema 3.1.1 *Seja um canal de capacidade C e uma fonte com taxa R, logo, se $R \leq C$ existe uma técnica de codificação tal que a saída da fonte pode ser transmitida através do canal com probabilidade de erro tão pequena quanto se queira.*

3.2 CÓDIGOS DE BLOCO LINEARES

Códigos de bloco são assim denominados, pois possuem dimensões constantes, são elas: tamanho da palavra-código n, comprimento da mensagem k. Os bits de redundância $(n - k)$ não possuem informações, eles tem a função exclusiva de permitir que possíveis erros sejam identificados e corrigidos. Um código de bloco é classificado como sendo linear se a combinação entre duas palavras-código resulte em outra palavra-código válida e a palavra código nula (constituída apenas por bits 0) seja considerada uma palavra-código válida. É possível definir os códigos de bloco lineares a partir de corpos finitos, considerando a teoria descrita por Galois.

Definição 3.2.1 *Sendo F_q um corpo finito de q elementos, um código C de dimensão k e comprimento n sobre F_q é dito um **código de bloco linear** se ele for um subespaço vetorial de F_q^n.*

Os códigos binários são representados a partir de $q = 2$, quando $q = 3$ os códigos são chamados ternários e para q qualquer temos um código $q$ário. O número de palavras-código de tamanho n é dado por q^k, e para códigos binários será 2^k. Nestas condições, um código de bloco C pode ser determinando pelos parâmetros n e k sendo denotado por $C(n, k)$.

Definição 3.2.2 *A razão entre o número de bits de informação k e o tamanho total do bloco n é denominado **taxa de codificação** R_C e é dado por*

$$R_C = \frac{k}{n}. \tag{3.9}$$

Definição 3.2.3 *Define-se **peso de Hamming** w como a quantidade de valores não-nulos de um determinado vetor ou palavra-código.*

Definição 3.2.4 *Sejam quaisquer dois vetores de comprimento n, a **distância de Hamming** d_h será dada pelo números de bits que diferem entre os dois vetores analisados.*

Exemplo 3.2.1 *Considere os vetores $\mathbf{v_1} = (011001)$, $\mathbf{v_2} = (110101)$ e $\mathbf{v_3} = (000101)$. O peso de Hamming será:*

$$w(011001) = 3,$$

$$w(110101) = 4,$$

$$w(000101) = 2.$$

E a distância de Hamming:

$$d(011001, 110101) = 3,$$

$$d(011001, 000101) = 3,$$

$$d(110101, 000101) = 2.$$

Definição 3.2.5 *A **distância mínima** d de um código C é determinada pela menor distância de Hamming entre todos os pares de palavra-código.*

Considerando o exemplo anterior, a distância mínima será $d = 2$. Este parâmetro é de suma importância para determinar a capacidade de detecção e de correção de erros, quanto maior a distância mínima de um código, maior será essa capacidade. Um código de bloco linear com distância mínima d é representado por $C(n, k, d)$.

Teorema 3.2.1 *Dado uma distância mínima d, a capacidade de detecção de erros será de $(d-1)$ e de correção $\lfloor \frac{d-1}{2} \rfloor$.*

Definição 3.2.6 *Seja $B = \{G_0, G_1, ..., G_{k-1}\}$ uma base de C e considere a matriz G cujas linhas são os vetores $G_i = \{G_{i,0}, G_{i,1}, ..., G_{i,n-1}\}$ onde $i = 0, 1, ..., k$. A matriz G é chamada **matriz geradora** de C, tem dimensões $k \times n$ e terá forma,*

$$G = \begin{pmatrix} G_0 \\ G_1 \\ \vdots \\ G_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{0,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1}. \end{pmatrix}.$$

Considerando $\mathbf{u} = (u_0, u_1, ..., u_{k-1})$ como a informação a ser codificada, as palavras-código serão geradas a partir da combinação linear dessa informação com a matriz geradora G conforme segue

$$\mathbf{v} = \mathbf{u}.G = u_0.G_0 + u_1.G_1 + \cdots + u_{k-1}.G_{k-1}. \tag{3.10}$$

Exemplo 3.2.2 *Seja o código de bloco linear $C(6, 3)$. Inicialmente é necessário definir uma base de C para construir a matriz geradora a partir dos $2^3 = 8$ vetores possíveis. Escolhendo arbitrariamente 3 vetores linearmente independentes, temos como uma possível base de C,*

$$B_C = \{110101, 101100, 011110\},$$

deste modo a matriz G será dada por:

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

Realizando as operações de acordo com a Equação 3.10, temos a codificação conforme segue:

Tabela 1 – Codificação utilizando a matriz geradora

u	**v** = **u**.G	**v**
000	v_0 = 0 (110101) + 0 (101100) + 0 (011110)	000000
001	v_1 = 0 (110101) + 0 (101100) + 1 (011110)	011110
010	v_2 = 0 (110101) + 1 (101100) + 0 (011110)	101100
101	v_3 = 1 (110101) + 0 (101100) + 1 (011110)	101011
100	v_4 = 1 (110101) + 0 (101100) + 0 (011110)	110101
101	v_5 = 1 (110101) + 0 (101100) + 1 (011110)	101011
110	v_6 = 1 (110101) + 1 (101100) + 0 (011110)	011001
111	v_7 = 1 (110101) + 1 (101100) + 1 (011110)	000111

Ao observar a mensagem codificada **v** na terceira coluna da Tabela 1, não é possível identificar a mensagem que foi enviada, ou seja, a mensagem só voltará a ser conhecida após passar pelo processo de decodificação. Esse tipo de codificação é denominado **não sistemática** [LIN e JR. 2004].

A **codificação sistemática** é aquela que permite que a palavra-código seja composta por dois segmentos: um segmento de informação composto por k bits e outro composto pelos $n-k$ bits de redundância. A ordem em que os segmentos são dispostos é apenas uma questão de convenção e irá depender do formato da matriz G, de forma geral definimos.

Definição 3.2.7 *Uma matriz geradora G de um código C está na **forma padrão ou sistemática** se*

$$G = [P|I_k], \tag{3.11}$$

onde P é uma matriz de dimensões $k \times (n-k)$ e I é uma matriz identidade de dimensões $k \times k$).

A matriz identidade na Definição 3.2.7 é responsável pela parcela do vetor código que irá mostrar a informação original, ou seja, conforme a Equação 3.11 o resultado do vetor código será:

Figura 12 – Palavra-código



fonte: Próprio Autor

Visto que a matriz geradora é composta de vetores linearmente independentes, é sempre possível por meio de permutações entre as linhas obter uma matriz na forma sistemática.

Exemplo 3.2.3 *Seja a matriz geradora do código C(6, 3) apresentada no Exemplo 3.2.2, é possível obter uma nova matriz G' na forma sistemática.*

$$G = \begin{pmatrix} G_0 \\ G_1 \\ G_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

A nova matriz G' será obtida por intermédio das seguintes operações

$$G'_0 = G_1 = 101100,$$

$$G'_1 = G_1 + G_2 = 110010,$$

$$G'_2 = G_0 + G_1 = 011001.$$

Logo,

$$G' = \begin{pmatrix} G'_0 \\ G'_1 \\ G'_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Consequentemente, realizando a codificação a partir da Equação 3.10 para a nova matriz G', teremos as seguintes palavras-código,

Tabela 2 – Codificação utilizando a matriz geradora na forma padrão

u	**v** = **u**.G	**v**
000	v_0 = 0 (101100) + 0 (110010) + 0 (011001)	000000
001	v_1 = 0 (101100) + 0 (110010) + 1 (011001)	011001
010	v_2 = 0 (101100) + 1 (110010) + 0 (011001)	110010
101	v_3 = 1 (101100) + 0 (110010) + 1 (011001)	110101
100	v_4 = 1 (101100) + 0 (110010) + 0 (011001)	101100
101	v_5 = 1 (101100) + 0 (110010) + 1 (011001)	110101
110	v_6 = 1 (101100) + 1 (110010) + 0 (011001)	011110
111	v_7 = 1 (101100) + 1 (110010) + 1 (011001)	000111

É possível observar que a matriz G' gera um código equivalente ao da matriz G, porém, os vetores código obtidos são diferentes.

Definição 3.2.8 *Dada a matriz geradora na forma sistemática, conforme a Equação 3.11, é possível obter uma* ***matriz controle de paridade****, conforme segue*

$$H = [I_{(n-k)\times(n-k)}|P^T]. \tag{3.12}$$

Enquanto a matriz geradora G é utilizada para codificar a informação, a matriz controle de paridade H é utilizada para verificar se uma palavra-código pertence ou não ao código. Devido a dualidade do subespaço gerado por H com relação a G, isso significa que os vetores de G são ortogonais aos de H, deste modo

$$G.H^T = \mathbf{0}. \tag{3.13}$$

Ou seja, se **c** é uma palavra-código de um código de bloco linear $C(n,k)$ então, **c** = **u**G, onde **u** é a mensagem a ser codificada. Assim,

$$\mathbf{c}H^T = \mathbf{u}GH^T = \mathbf{0}.$$

Logo, um vetor recebido $\mathbf{v}$ será considerado uma palavra-código se $\mathbf{v}GH^T = \mathbf{0}$. Caso contrário, detectamos que um erro ocorreu.

Exemplo 3.2.4 *Seja a matriz geradora na forma sistemática para o código C(6,3) obtido no Exemplo 3.2.3,*

$$G = [P_{k\times(n-k)}|I_{k\times k}] = \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right),$$

a obtenção de H a partir de 3.12 será,

$$H = [I_{(n-k)\times(n-k)}|P^T] = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right).$$

Considerando dois vetores $\mathbf{v_1} = (110101)$ *e* $\mathbf{v_2} = (110011)$*, para verificar se estes vetores pertencem ao código, realiza-se o produto interno entre o vetor recebido e a matriz controle de paridade transposta* H^T *conforme condição mostrada em (3.13), deste modo,*

$$\mathbf{v_1}.H^T = (110101)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = 1(100) + 1(010) + 0(001) + 1(101) + 0(110) + 1(011) = 000,$$

$$\mathbf{v_2}.H^T = (110011)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = 1(100) + 1(010) + 0(001) + 0(101) + 1(110) + 1(011) = 011.$$

Logo, apenas o vetor $\mathbf{v_1}$ *pertence ao código.*

Ao transmitir uma informação através de um canal, a informação pode ser corrompida de acordo com as características do canal, uma das maneiras de realizar a detecção e a correção do erro é a partir dos vetores síndrome e erro que definiremos a seguir.

Definição 3.2.9 *Seja* $\mathbf{v}$ *o vetor recebido após a transmissão da palavra-código* $\mathbf{c}$ *Se*

$$\mathbf{v} = \mathbf{c} + \mathbf{e}, \tag{3.14}$$

então definimos $\mathbf{e} = (e_0, e_1..., e_{n-1})$ *como o* ***vetor erro****, e a* ***síndrome*** *é definida a partir da relação com a matriz controle de paridade conforme segue*

$$\mathbf{S} = \mathbf{v}.H^t. \tag{3.15}$$

Correlacionando as Equações 3.14 e 3.15, temos que

$$\mathbf{S} = (\mathbf{c} + \mathbf{e}).H^t = \mathbf{c}.H^t + \mathbf{e}H^t. \tag{3.16}$$

Entretanto, como

$$\mathbf{c}.H^t = 0, \tag{3.17}$$

então a síndrome da palavra recebida é a mesma do padrão de erro

$$\mathbf{S} = \mathbf{e}.H^t. \tag{3.18}$$

Cada padrão de erro terá uma única síndrome associada, o que auxiliará no processo de correção de eventuais erros. Entretanto existem padrões de erros que podem resultar em uma síndrome nula, estes são conhecidos como padrão de erros não-detectáveis e ocorre quando o padrão de erro é exatamente igual a uma palavra-código.

Ao avaliar a síndrome é possível identificar se ocorreram erros e posteriormente corrigí-los de acordo com a capacidade do código. O processo de decodificação pela síndrome inicia-se encontrando as síndromes conforme a Equação 3.15, se o resultado for igual a **0** isso indica que a palavra é uma palavra-código válida, caso contrário indica que um erro foi detectado. Ao identificar o padrão de erro é possível identificar a posição em que o erro ocorreu e corrigí-lo a partir da chamada tabela de síndrome, a qual será exemplificada a seguir.

Exemplo 3.2.5 *Seja o código de bloco linear* $(6, 3, 3)$*. Como a distância mínima é igual a 3, podemos partir da equação* $t = \lfloor \frac{d-1}{2} \rfloor$ *e calcular a capacidade de correção deste código que será de* 1 *erro, a partir dessa informação é possível determinar os padrões de erros.*

Tabela 3 – Tabela de Padrão de Erros

$\mathbf{e}$					
1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

Dada a matriz H *como sendo:*

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

Podemos calcular a partir da Equação 3.18 a síndrome associada a cada padrão de erro, por exemplo,

$$\mathbf{S_0} = \mathbf{e_0}.H^T = (100000).H^T = 100,$$

$$\mathbf{S_1} = \mathbf{e_1}.H^T = (010000)H^T = 010,$$

e assim por diante. Após calcular a sindrome para todos os padrões de erros, podemos então montar a tabela da síndrome, que nos auxiliará na decodificação do código.

Tabela 4 – Tabela de Síndromes

$\mathbf{e}$	$\mathbf{S}$
100000	100
010000	010
001000	001
000100	101
000010	110
000001	011

Após a obtenção das informações acima é possível identificar e corrigir qualquer palavra código que apresente até 1 *erro. Seja a palavra-código transmitida* $\mathbf{c} = (101011)$*, e após ter um padrão de erros inseridos pelo canal, a palavra recebida foi* $\mathbf{v} = (101010)$*. Para corrigir o erro inserido é necessário identificar o padrão de erro, utilizando a Equação 3.15:*

$$\mathbf{S} = \mathbf{v}.H^t = [101010] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = 1(100) + 0(010) + 1(001) + 0(101) + 1(110) + 0(011) = 011.$$

Identificada a síndrome associada ao padrão de erro como sendo $\mathbf{S} = 011$*, podemos consultar a tabela de síndromes (Tabela 4) e identificar o bit que sofreu alteração, e em seguida corrigí-lo. Assim, o erro associado a síndrome encontrada é* $\mathbf{e} = 000001$*, ou seja, o erro ocorreu no último bit da palavra código transmitida, para corrigí-lo soma-se o padrão de erro a palavra código recebida,*

$$\mathbf{c} = \mathbf{v} + \mathbf{e} = 101010 + 000001 = 101011.$$

Logo a palavra-código $\mathbf{c}$ *transmitida foi recuperada.*

A decodificação por síndrome apresenta limitações devido ao custo computacional, quando utilizada em códigos de dimensões elevadas pode requerer um grande números de cálculos e memória para

armazenar todos os possíveis padrões de erros. Ainda a partir da síndrome, podemos decodificar e corrigir um padrão de erro na palavra recebida apenas analisando as colunas de H. Seja h_i a *i-ésima* coluna de H. Inicialmente calcula-se a síndrome da palavra recebida $\mathbf{v}$, caso a síndrome seja $\mathbf{S} = 0$ consideramos que não houve erro na transmissão e assumimos $\mathbf{v}$ como a palavra que foi transmitida, caso contrário verificamos se a transposta da síndrome $\mathbf{S}^t$ corresponde a alguma das h_i colunas da matriz controle de paridade H. Em caso positivo, a correção será aplicada ao *i-ésimo* bit transmitido.

Exemplo 3.2.6 *Utilizando o Exemplo 3.2.5, dado a transmissão da palavra-código* $\mathbf{c} = (101011)$*, temos que o vetor recebido foi* $\mathbf{v} = (101010)$*. Sabemos a partir dos cálculos já realizados que a síndrome será* $\mathbf{S} = (011)$*. Neste método iremos comparar a transposta da síndrome calculada com as colunas de* H*. Sendo*

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

e

$$\mathbf{S}^t = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix},$$

podemos verificar que $\mathbf{S}^t$ *corresponde à sexta coluna de* H*, ou seja, isto significa que o erro ocorreu na sexta posição da palavra recebida. Sendo assim, mudamos o sexto bit da palavra recebida que deixa de ser* $\mathbf{v} = (101010)$ *e passa a ser* $\mathbf{c} = (101011)$.

4 CÓDIGOS LDPC

Dentre as classes de códigos de blocos lineares, os códigos de verificação de paridade de baixa densidade (*Low Density Parity Check Code*, LDPC) se destacaram por seu desempenho, que quando projetados cuidadosamente como em [SAE-YOUNG CHUNG et al. 2001], pode se aproximar do limite teórico para canais gaussianos estabelecido por Shannon. Este tipo de código foi proposto pela primeira vez por Gallager no início dos anos 60, entretanto, devido a falta de recursos computacionais da época, não foi viável implementá-lo e o estudo acabou esquecido por anos [GALLAGER 1962]. O tema voltou a ser relevante apenas na década de 90 por meio do trabalho de Mackay e Neal, que acabou despertando grande interesse da comunidade científica, gerando assim diversos estudos na área [MACKAY e NEAL 1996, MIZUOCHI e MIYATA 2008, ARABACI et al. 2010].

Neste capítulo serão apresentados os conceitos referentes aos códigos LDPC. Na Seção 4.1 estão descritas as definições pertinentes para este trabalho, iniciando com os parâmetros essenciais e que caracterizam os códigos como regulares ou irregulares, além de descrever e exemplificar o conceito de grafos bipartidos aplicados à códigos LDPC e a importância do *girth*. Na Seção 4.2, são apresentados 2 métodos para a obtenção da matriz controle de paridade de um código LDPC regular, são eles: na Subseção 4.2.1 a construção de Gallager e na Subseção 4.2.2 o método conhecido como BIBD. Finalmente na Seção 4.3 são demonstrados e exemplificados dois métodos de decodificação: na Subseção 4.3.1 o método *bit-flipping* que utiliza o critério de decisão *hard-decision* e na Subseção 4.3.2 o algoritmo soma-produto (SPA) que ao contrário do primeiro utiliza o método de decisão *soft-decision*. Os conteúdos abordados neste capítulo podem ser encontrados em [GALLAGER 1962, CARRASCO e JOHNSTON 2008, RICHARDSON, SHOKROLLAHI e URBANKE 2001, LIN e JR. 2004, RYAN e LIN 2009]

4.1 DEFINIÇÕES

Definição 4.1.1 *Um código de bloco linear $C(n, k)$ será chamado de um **código LDPC** se, e somente se, a matriz controle de paridade H for esparsa, ou seja, apresentar uma baixa densidade de elementos não-nulos em comparação com a dimensão da matriz.*

Neste trabalho iremos considerar somente códigos sobre F_2, ou seja, códigos binários. Assim, iremos buscar construir matrizes com baixa densidade de 1's. Considera-se como densidade de uma fonte de bits aleatórios, a fração esperada de bits 1. Assim, uma fonte é considerada esparsa se a sua densidade $\mathcal{D} < 0.5$.

Os códigos LDPC podem ser classificados como regulares ou irregulares, dependendo da quantidade de 1's nas linhas e colunas serem fixos ou não. Se todas as linhas da matriz controle de paridade H tiverem o mesmo peso de Hamming assim como as colunas, dizemos que o código LDPC é regular.

Definição 4.1.2 *Um **código LDPC regular** C com parâmetros (n, w_c, w_r) é um código de comprimento n com uma matriz H em que cada coluna contém w_c 1's e cada linha contém w_r 1's, com $w_c \geq 3$ e $w_r \geq w_c$.*

Se H é a matriz controle de paridade de um código LDPC regular C com parâmetros (n, w_c, w_r) e dimensão $m \times n$, onde $m = n - k$, então

$$mw_r = nw_c. \tag{4.1}$$

Neste caso, podemos definir a **densidade** de H, como sendo

$$\mathcal{D} = \frac{mw_r}{mn} = \frac{w_r}{n} = \frac{nw_c}{mn} = \frac{w_c}{m}. \tag{4.2}$$

Como vimos, a matriz será considerada esparsa se, e somente se, $\mathcal{D} < 0.5$, para isso é desejável que $w_c \ll n - k$ e $w_r \ll n$. A baixa densidade de 1's nos códigos LDPC binários pode não garantir uma matriz controle de paridade que apresente independência linear entre todas as linhas, porém, por intermédio da eliminação de Gauss é possível obter uma matriz linearmente independente que provavelmente não terá peso constante nas colunas e linhas. Habitualmente essas dependências são desconsideradas afim de manter a regularidade do código.

Definição 4.1.3 *Seja C um código LDPC regular com parâmetros (n, w_c, w_r). Então a **taxa de codificação** será*

$$R_C = \frac{k}{n} \geq 1 - \frac{w_c}{w_r}. \tag{4.3}$$

Exemplo 4.1.1 *Seja um código de bloco linear C com parâmetros $(n, k) = (10, 5)$ com matriz controle de paridade dada por*

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

O código C pode ser definido como um código LDPC regular, pois, o peso de todas as colunas é $w_c = 2$ e o peso de todas as linhas é $w_r = 4$. Além disso, a densidade a partir da Equação 4.2 será:

$$\mathcal{D} = \frac{w_r}{n} = \frac{4}{10} = 0.4.$$

Do mesmo modo que a taxa de codificação a partir da Equação 4.3 será:

$$R_C = 1 - \frac{w_c}{w_r} = 1 - \frac{2}{4} = 0.5.$$

Definição 4.1.4 *Um código LDPC C que apresenta uma matriz H em que a quantidade de 1's das linhas e colunas não são constantes é denominado como **código LDPC irregular**.*

Exemplo 4.1.2 *Um código LDPC irregular C pode ser construído a partir da seguinte matriz controle de paridade com dimensão 5×10*

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

onde a densidade de H pode ser calculada como a proporção de 1's em H, então $\mathcal{D} = 0.48 < 0.5$

Richardsom e Urbanke apresentam em seu trabalho, o melhor desempenho dos códigos irregulares para blocos longos quando comparado aos códigos regulares, ainda assim, códigos regulares são largamente aplicados devido à sua simplicidade de implementação [RICHARDSON, SHOKROLLAHI e URBANKE 2001]. Neste trabalho iremos considerar apenas construções de códigos LDPC regulares, não aprofundando portanto, nos códigos LDPC irregulares.

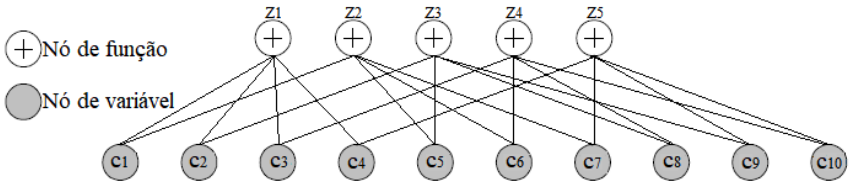
Uma característica importante dos códigos de bloco lineares muito utilizada para representar códigos LDPC é que para qualquer código de bloco linear $C(n, k)$ com matriz controle de paridade H, esta matriz pode ser representada por meio de um **grafo bipartido** segundo Michael Tanner, conhecido também como **grafo de Tanner**, onde cada elemento 1 da matriz H será representado como uma ligação entre um **nó de função** ou **nó de paridade** z_m, correspondente a $m = n - k$ equações de paridade, e um **nó de variável** c_n, correspondente a n bits na palavra-código.

Exemplo 4.1.3 *Seja C um código LDPC regular com parâmetros $(10, 2, 4)$ e com matriz controle de paridade dada por*

$$H = \begin{array}{c} \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

Cada elemento 1 da matriz irá representar uma ligação entre o nó de função z_i e o nó de variável c_j, com $i = 1, \ldots, 5$ e $j = 1, \ldots, 10$. Logo, a representação de H em um grafo bipartido será:

Figura 13 – Grafo Bipartido



fonte: Próprio Autor

Se um código LDPC regular C com parâmetros (n, w_c, w_r) e com matriz controle de paridade H é representada por um grafo bipartido, então em um nó de função z_m, a soma das informações, ou seja,

a soma dos nós de variavéis conectados a esse nó de função deverá ser 0. Esta operação define uma equação de paridade.

O grau de um nó, de função ou de variável, é determinado de acordo com o números de linhas que chegam a determinado nó, em outras palavras, para um código LPDC regular (n, w_c, w_r) representado por meio do grafo de Tanner haverá m nós de paridade de grau w_r e n nós de variável com grau w_c. Dessa forma, podemos introduzir dois conjuntos:

- M_n como o conjunto de índices das verificações de paridade conectadas ao bit c_n codificado;
- N_m sendo o conjunto de índices dos bits codificados que estão conectados à verificação de paridade z_m.

O grafo de Tanner e obtenção das equações de verificação de paridade, serão fundamentais para facilitar o entendimento do processo de decodificação iterativa em códigos LDPC. No exemplo a seguir, exemplificamos as equações de paridade bem como os conjuntos de índices.

Exemplo 4.1.4 *Considerando o grafo bipartido do Exemplo 4.1.3 representado na Figura 13, as equações de verificação de paridade serão:*

$$z_1 = c_1 + c_2 + c_3 + c_4 = 0,$$

$$z_2 = c_1 + c_5 + c_6 + c_7 = 0,$$

$$z_3 = c_2 + c_5 + c_8 + c_9 = 0,$$

$$z_4 = c_3 + c_6 + c_8 + c_{10} = 0,$$

$$z_5 = c_4 + c_7 + c_9 + c_{10} = 0.$$

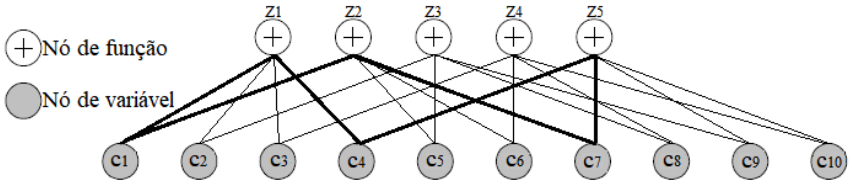
Para este caso, temos por exemplo que $M_1 = 1, 2$, visto que no nó z_1 e z_2 estão conectados ao nó c_1. E, $N_1 = 1, 2, 3, 4$, pois os nós c_1, c_2, c_3 e c_4 estão conectados ao nó de função z_1.

Definição 4.1.5 *Em um grafo bipartido, as ligações das arestas que formam um caminho fechado é chamado de ciclo de tamanho l. O menor ciclo de um grafo é denominado **girth**.*

O conceito de *girth* está intimamente relacionado ao desempenho de alguns tipos de decodificações iterativas, como por exemplo o algoritmo soma-produto (SPA) que será apresentado neste trabalho. Ao projetar um código LDPC é necessário buscar obter o maior *girth* possível ao construir a matriz H, todo ciclo será de tamanho par e seu valor mínimo é 4, sendo esse o pior caso em relação ao desempenho do código. A fim de evitar um ciclo de comprimento 4, é necessário garantir que quaisquer duas colunas da matriz de paridade tenham no mínimo um 1 em comum na mesma posição [CARRASCO e JOHNSTON 2008, RYAN e LIN 2009].

Exemplo 4.1.5 *Utilizando o grafo da Figura 13 do Exemplo 4.1.3, é possível identificar ciclos de vários tamanhos, sendo que o menor ciclo é o destacado em negrito na Figura 14. Portanto, o girth é igual a 6.*

Figura 14 – Grafo Bipartido com girth 6



fonte: Próprio Autor

É possível construir uma matriz controle de paridade H satisfatória de inúmeras formas. Entretanto o maior desafio para um código LDPC é sua codificação. Como os códigos LDPC são um tipo de código de bloco linear, essencialmente é possível codificá-lo a partir da obtenção da matriz geradora, conforme citado na Seção 3.2 e indicado em (3.11). Como os códigos LDPC focam na obtenção da matriz H, na maioria das vezes essa matriz não tem a forma sistemática, sendo possível obter uma matriz de paridade na forma sistemática utilizando a eliminação de Gauss. Porém, ao realizar permutações nas linhas e colunas, pode-se obter então uma matriz com densidades de $1's$ totalmente diferente da original, mas com desempenho semelhante. Embora a obtenção da matriz geradora pareça direta, possui um custo computacional alto devido ao número de operações necessárias para a obtenção de G que dificilmente será esparsa.

4.2 CONSTRUÇÃO DE CÓDIGOS LDPC REGULARES

Códigos LDPC são pautados principalmente em obter uma matriz controle de paridade H esparsa, e que cumpra alguns requisitos definidos, como por exemplo a Definição 4.1.2 de regularidade ou ainda garantir um determinado valor de *girth*. Para quaisquer restrições iniciais, existirá um conjunto enorme de códigos que as satisfaçam. A obtenção de uma controle de paridade é feita de maneira quase aleatória, porém ao seguir algumas regras é possível maximizar as probabilidades de se obter um código com eficiência elevada. Para tal, ao longo dos anos diversos métodos foram estudados para a obtenção da matriz H. Nesta seção, iremos apresentar três construções de códigos LDPC baseadas na forma de se obter a matriz controle de paridade H. Na Subseção 4.2.1 apresentamos a construção proposta por Gallager e na Subseção 4.2.2 o método por meio de projetos de blocos incompletos balanceados, conhecido como BIBD. No Capítulo 5 iremos utilizar as construções de Gallager e por BIBD para analises de desempenho.

4.2.1 Construção de Gallager

Gallager inicialmente propôs um método que se obtêm uma matriz de paridade com alta esparsidade, porém, a matriz não é representada de forma sistemática. Este método é para construção de códigos LDPC regulares, [GALLAGER 1962].

Seja C um código LDPC regular com parâmetros (n, w_c, w_r) e matriz controle de paridade H de ordem $m \times n$, onde $m = n - k$. A partir da relação da Equação 4.1, podemos determinar a quantidade

de linhas m da matriz H por

$$m = \frac{nw_c}{w_r}. \tag{4.4}$$

A construção de Gallager para construir a matriz controle de paridade H, é dada pelos seguintes passos:

1. Determinar o número m de linhas da matriz por meio da Equação 4.4.
2. Dividir a quantidade de linhas m em w_c submatrizes de ordem $(m/w_c) \times n$, onde cada submatriz terá apenas um único 1 por coluna.
3. Na primeira submatriz, cada linha tera uma sequência de w_r $1's$, sempre deslocando de uma linha para outra mantendo um único 1 por coluna.
4. As demais $(w_c - 1)$ submatrizes de H são completadas com permutações aleatórias das colunas da primeira submatriz gerada, respeitando as quantidades w_c e w_r.

Como as três primeiras etapas do processo são fixadas, a fim de obter códigos LDPC mais eficientes a partir da construção de Gallager, na Etapa 4. busca-se realizar as permutações aleatórias de modo a obter o maior *girth* possível. No exemplo a seguir, ilustramos a construção de uma matriz controle de paridade H obtida pelo método de Gallager.

Exemplo 4.2.1 *Seja C um código LDPC regular com $(n, w_c, w_r) = (20, 3, 4)$. Primeiramente a partir da Equação 4.4 obtemos que $m = 15$ e teremos $w_c = 3$ submatrizes, onde a primeira linha da primeira submatriz terá $w_r = 4$ $1's$ de forma sequencial e nas demais linhas a sequência de 1's será deslocada de modo a obter um único 1 por coluna. Em seguida as demais submatrizes são obtidas por meio de permutações aleatórias das colunas da primeira submatriz. Uma das possibilidades de matriz controle de paridade para esse exemplo, será*

$$H = \left[\begin{array}{cccccccccccccccccccc}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1
\end{array}\right]. \tag{4.5}$$

4.2.2 Construção BIBD

A construção por design de blocos incompletos balanceados BIBD (*Balanced Incomplete Block Design*) é baseada em uma técnica, conhecida como design combinatório, que é uma maneira de organizar elementos de um conjunto finito em subconjuntos para que certas propriedades sejam satisfeitas. Esta técnica pode ser aplicada em teoria da codificação, neste caso, podendo ser utilizada para obter uma matriz controle de paridade de um código LDPC regular [VASIC e DJORDJEVIC 2002, AMMAR et al. 2004].

Definição 4.2.1 *Considere um design combinatório com parâmetros $(\mathcal{V}, B, \mathcal{R}, \mathcal{K}, \lambda)$, onde $\mathcal{V}$ é um conjunto de pontos, B é o número de subconjuntos de $\mathcal{V}$ chamados de blocos, $\mathcal{R}$ é o número de repetição dos blocos, $\mathcal{K}$ é o número de elementos contidos em cada bloco e λ é o número de blocos que o mesmo elemento se repete e considerando que $\mathcal{V} \geq \mathcal{K}$ (para que nenhum bloco contenha todos os elementos do conjunto). Definimos esse design como $(\mathcal{V}, B, \mathcal{R}, \mathcal{K}, \lambda)$- BIBD, e é chamado de* ***design de bloco incompleto balanceado****.*

Utilizando o design combinatório apresentado na Definição 4.2.1 e aplicando parâmetros específicos é possível construir uma matriz controle de paridade H para códigos LDPC regulares, como uma matriz de incidência entre pontos e blocos. Entre os parâmetros, o mais importante para caracterizar um código LDPC, é o número de blocos contendo o mesmo par de pontos λ, que para códigos LDPC sempre será considerado 1, desta forma podemos representar o design combinatório neste caso como $(\mathcal{V}, \mathcal{K}, \lambda)$-BIBD [VASIC e DJORDJEVIC 2002, MOREIRA e 1 2019].

Ao aplicar a construção BIBD à um código LDPC, os parâmetros se relacionam da seguinte maneira:

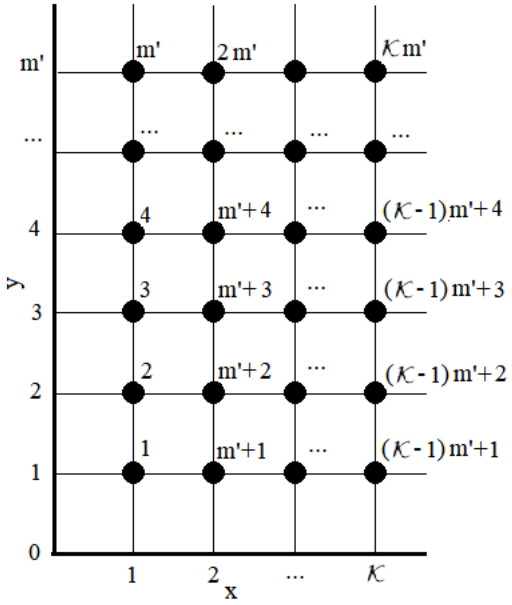
- $\mathcal{V}$ é o número de bits redundantes $n - k$ e
- $\mathcal{K}$ é o peso de 1 nas colunas.

Por meio do método BIBD é possível determinar a matriz verificadora de paridade H como uma matriz de incidência, onde cada elemento de um bloco B determina um elemento não-nulo da matriz. Utilizando um *grid* retangular, é possível distribuir os pontos de interesse de maneira que o grid haverá $\mathcal{K}$ linhas verticais, e o número de linhas horizontais terá um valor $m' = \frac{\mathcal{V}}{\mathcal{K}}$ de modo a obter $\mathcal{V}$ pontos de cruzamento entre as linhas verticais e horizontais, conforme exemplificado na Figura 15.

Utilizando o *grid* retangular, os blocos serão determinados a partir dos *slopes* (inclinação), essa inclinação será uma linha reta que agrupará $\mathcal{K}$ elementos em um bloco, ou seja, um *slope* 0, significa que os elementos de cada linha estarão agrupados juntos, visto que seria uma linha reta sem inclinação nenhuma. Conforme exemplificado na Figura 16, a linha vermelha representa o *slope* 0, a linha azul representa o *slope* 1 e a linha vermelha o *slope* 2.

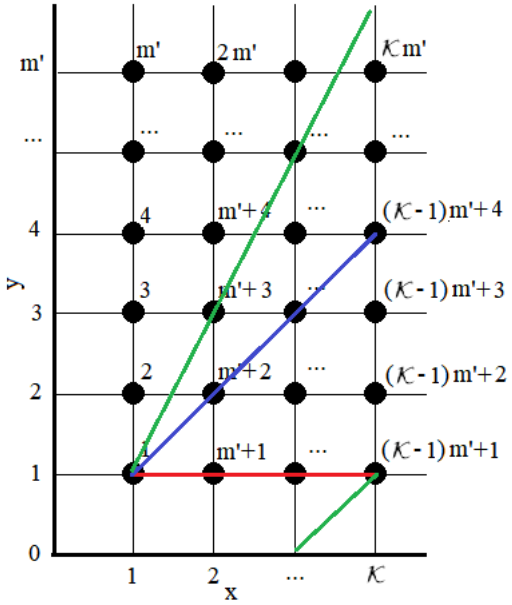
A quantidade de *slopes* a serem analisados será $s = 0, 1, 2, ..., m' - 1$, dado que a análise inicia-se no *slope* 0. Sendo assim haverão $B = (m')^2$ blocos de $\mathcal{K}$ elementos. A partir dos blocos, a matriz controle de paridade H pode ser obtida de maneira que cada bloco B representa uma coluna da matriz H, e cada elemento do bloco, indica a linha da matriz que haverá um elemento não-nulo.

Figura 15 – Distribuição de pontos em um grid retangular



fonte: Próprio autor

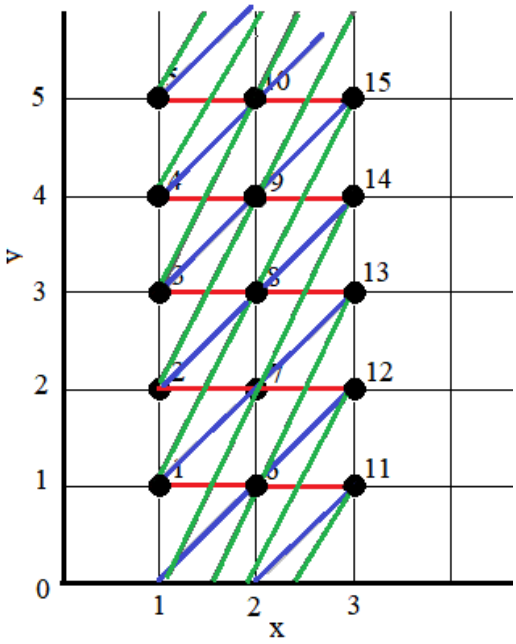
Figura 16 – Representação dos slopes 0, 1 e 2



fonte: Próprio autor

Exemplo 4.2.2 *Sejam C um código LDPC regular com parâmetros $(25, 3, 5)$ e o design combinatório BIBD da forma $2 - (15, 3, 1)$, logo $m' = 5$. Podemos então determinar o grid retangular como:*

Figura 17 – Grid Retangular com slopes= 0, 1, 2



fonte: Próprio Autor

Para este caso teremos $s = 0, 1, 2, 3, 4$ *e a partir desses valores é possível dispor todos os pontos conforme a Tabela 5.*

Tabela 5 – Tabela Slopes

s=0	s=1	s=2	s=3	s=4
1 6 11	1 7 13	1 8 15	1 9 12	1 10 14
2 7 12	2 8 14	2 9 11	2 10 13	2 6 15
3 8 13	3 9 15	3 10 12	2 6 14	3 7 11
4 9 14	4 10 11	4 6 13	4 7 15	4 8 12
5 10 15	5 6 12	5 7 14	5 8 11	5 9 13

Após construir a tabela, cada bloco B *de* 3 *elementos, define a posição dos elementos* $1's$ *em cada coluna da matriz* H*, que terá dimensão* $\mathcal{V} \times B$ *onde* $B = (m')^2$*. Neste caso,* 15×25*, ou seja, a primeira coluna da matriz* H *terá elementos 1's nas linhas* $1, 6$ *e* 11*. Seguindo essa lógica a matriz*

correspondente será

$$H = \begin{bmatrix}
1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0 \\
0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0 \\
0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0 \\
0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0 \\
0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1 \\
1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \\
0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0 \\
0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0 \\
0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1 \\
0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0 \\
1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\
0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\
0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\
0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\
0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0
\end{bmatrix}. \tag{4.6}$$

4.3 DECODIFICAÇÃO

Os decodificadores para códigos LDPC, na maioria dos casos, utiliza algoritmos de passagem de mensagens utilizando o conceito de grafos. Entre os tipos de algoritmo de decodificação estão o *bit-flipping* e algoritmo soma-produto (*Sum-Product Algorithm* - SPA), também conhecido como *belief propagation*, ambas designações são encontradas em diversos trabalhos e descrevem o mesmo algoritmo. A depender do método de decisão, decodificadores usualmente são divididos em duas classes: *hard-decision* e *soft-decision*. Dentre os algoritmos citados, o *bit-flipping*, que será apresentado na Subseção 4.3.1, é um algoritmo do tipo *hard-decision* e o algoritmo soma-produto, que será apresentado na Subseção 4.3.2, é um algoritmo do tipo *soft-decision*.

Algoritmos baseados em *hard-decision*, na sua grande maioria são menos complexos requerendo assim, um menor processamento computacional quando comparado com o *soft-decision*. Entretanto, algoritmos baseados em *soft-decision* apresentam um melhor desempenho em termos da BER, o que os tornam mais utilizados em diversos estudos. No Capítulo 5, para análise dos resultados, aplicaremos a decodificação utilizando o algoritmo soma-produto, que conforme citado baseia-se em *soft-decision*.

4.3.1 *Bit-Flipping*

O algoritmo *bit-flipping* foi proposto por Gallager e utiliza o conceito de *hard-decidion* para realizar a decodificação.

Definição 4.3.1 *Seja um código C(n, k), em que o conjunto de símbolos na entrada do decodificador é finito* $(0, 1)$*, a decisão se cada bit recebido foi* 0 *ou* 1 *é tomada antes do decodificador, em seguida*

escolhe-se a palavra-código mais próxima do vetor recebido. Este método é chamado de ***hard-decidion****.*

O método *bit-flipping* é considerado simples e consiste basicamente em utilizar os grafos e equações de paridade para verificar quais equações de paridade falham, e analisando as equações estimar o bit que mais falha e em seguida alterá-lo e testar novamente, até obter o menor erro possível.

Seja C um código LDPC com matriz controle de paridade H e seja $\mathbf{v}$ o vetor recebido. O algoritmo segue os seguintes passos:

1. Para o vetor recebido $\mathbf{v}$, escreva todas as equações de paridade.
2. Verificar para cada bit quantas equações de paridade falharam.
3. Após identificar o(s) bit(s) que mais apresentaram equações de paridade com falha, realizar a troca desse bit $(0 \to 1, 1 \to 0)$, gerando um vetor $\mathbf{v}'$.
4. Repetir os passos de $1.$ a $3.$ com o novo vetor $\mathbf{v}'$ até satisfazer todas as equações de paridade.

Exemplo 4.3.1 *Seja C o código LDPC regular com parâmetros $(10, 2, 4)$ como no Exemplo 4.1.3. A matriz controle de paridade H foi dada por*

$$H = \begin{array}{c} \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

Considerando que a palavra recebida foi $\mathbf{v} = (1101001001)$, testa-se todas as equações de paridade como na Tabela 6

Tabela 6 – Teste 1 - Equações de Paridade

z	Equações de Paridade	Teste
1	$(c_1 + c_2 + c_3 + c_4)$	1
2	$(c_1 + c_5 + c_6 + c_7)$	0
3	$(c_2 + c_5 + c_8 + c_9)$	1
4	$(c_3 + c_6 + c_8 + c_9)$	1
5	$(c_4 + c_7 + c_9 + c_{10})$	1

Verifica-se todas as equações que falharam e estima-se os bits que mais apareceram, neste caso o bit c_9 apareceu em 3 das 4 equações que falharam. Assim, alteramos o bit e teremos a nova palavra $\mathbf{v}' = (1101001011)$. Para $\mathbf{v}'$, realiza-se os testes novamente com o novo vetor como na Tabela 7

Tabela 7 – Teste 2 - Equações de Paridade

z	Equações de Paridade	Teste
1	$(c_1 + c_2 + c_3 + c_4)$	1
2	$(c_1 + c_5 + c_6 + c_7)$	0
3	$(c_2 + c_5 + c_8 + c_9)$	0
4	$(c_3 + c_6 + c_8 + c_9)$	1
5	$(c_4 + c_7 + c_9 + c_{10})$	0

Neste segundo passo o único bit em comum nas duas equações que apresentaram falha foi o bit c_3*. Realizando a atualização do vetor teremos* $\mathbf{v}" = (1111001011)$*, e em seguida o teste é novamente realizado, como na Tabela 8*

Tabela 8 – Teste 3 - Equações de Paridade

z	Equações de Paridade	Teste
1	$(c_1 + c_2 + c_3 + c_4)$	0
2	$(c_1 + c_5 + c_6 + c_7)$	0
3	$(c_2 + c_5 + c_8 + c_9)$	0
4	$(c_3 + c_6 + c_8 + c_9)$	0
5	$(c_4 + c_7 + c_9 + c_{10})$	0

Finalmente, obtemos todas as equações de paridade satisfeita, indicando que a palavra transmitida foi $\mathbf{c} = \mathbf{v}" = (1111001011)$*. Para comprovar basta verificar a condição* $\mathbf{c}.H^T = 0$*.*

Existem algumas variações da aplicação do algoritmo *bit-flipping*, como por exemplo o *weighted bit-flipping* que é uma versão *soft-decidion* do *bit-flipping* [SANTOS 2014].

4.3.2 Algoritmo Soma-Produto

O algoritmo soma-produto - SPA (*Sum-Product Algorithm*) por sua vez, utiliza-se do método de decisão *soft-decision*.

Definição 4.3.2 *Considerando um código C(n, k), em que o decodificador ao receber a palavra transmitida avalia a distribuição probabilística dos possíveis valores dos bits. Define-se este método de decodificação como* ***soft-decision****, onde a decisão é tomada baseada na palavra inteira.*

Seja C um código LDPC com matriz controle de paridade H representada por $m = n - k$ equações de paridade associadas aos nós de paridade z_m, e os nós de variável c_n, correspondente a n bits na palavra-código. O SPA consiste basicamente em encontrar a palavra código $\mathbf{c}$ de comprimento n onde cada bit codificado c_n maximize a probabilidade de que as equações de paridade z_m que envolvam esse bit c_n sejam satisfeitas [CARRASCO e JOHNSTON 2008]. Ou seja, queremos maximizar a seguinte probabilidade

$$P(c_n | z_m = 0), \quad \text{com} \quad m \in M_n, \tag{4.7}$$

onde M_n como visto na Seção 4.1 são os índices de z_m conectados a c_n.

O algoritmo soma-produto é baseado no cálculo de duas probabilidades. A primeira probabilidade q_{mn}, que representa a probabilidade do n-ésimo bit codificado satisfazer a equação de paridade a que está correlacionado, com exceção da m-ésima equação de paridade, ou seja

$$q_{mn}(x) = P(c_n = x | z_m = 0), \quad \text{com} \quad m' \in M_n/m, \tag{4.8}$$

onde M_n/m representa todos os índices dos nós de paridade conectados a c_n exceto o nó de paridade z_m.

A segunda probabilidade r_{mn}, representa a probabilidade da m-ésima equação de paridade ser satisfeita condicionada a todos os possíveis valores dos bits codificados da palavra-código $\mathbf{c}$, ou seja

$$r_{mn}(x) = P(z_m = 0 | \mathbf{c}). \tag{4.9}$$

O algoritmo soma-produto utiliza o método iterativo, onde cada iteração atualiza os valores das probabilidades, aumentando a eficiência do código. A inicialização da probabilidade de $q_{mn}(x)$ em (4.8), é dada pela probabilidade de que o n-ésimo bit recebido seja x, ou seja, $P(c_n = x)$. Tal probabilidade irá variar de acordo com o canal. Iremos considerar neste trabalho uma distribuição de probabilidade gaussiana de média zero e variância σ^2, ou seja, um canal AWGN dado por

$$g_n^{(x)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}. \tag{4.10}$$

A probabilidade $g_n^{(x)}$ de um bit recebido ser 0 ou 1 para uma modulação BPSK, será respectivamente

$$g_n^{(0)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-(v_n-1)^2/2\sigma^2} \tag{4.11}$$

e

$$g_n^{(1)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-(v_n+1)^2/2\sigma^2}, \tag{4.12}$$

onde v_n são os bits do vetor recebido.

Considerando uma modulação QPSK, a probabilidade de um símbolo recebido ser 00, 01, 10 ou 11, serão

$$g_n^{(00)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(v_n^{(I)} - \frac{1}{\sqrt{2}}\right)^2 + \left(v_n^{(Q)} - \frac{1}{\sqrt{2}}\right)^2\right] / 2\sigma^2}, \tag{4.13}$$

$$g_n^{(01)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(v_n^{(I)} - \frac{1}{\sqrt{2}}\right)^2 + \left(v_n^{(Q)} + \frac{1}{\sqrt{2}}\right)^2\right] / 2\sigma^2}, \tag{4.14}$$

$$g_n^{(10)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(v_n^{(I)} + \frac{1}{\sqrt{2}}\right)^2 + \left(v_n^{(Q)} - \frac{1}{\sqrt{2}}\right)^2\right] / 2\sigma^2}, \tag{4.15}$$

$$g_n^{(11)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(v_n^{(I)} + \frac{1}{\sqrt{2}}\right)^2 + \left(v_n^{(Q)} + \frac{1}{\sqrt{2}}\right)^2\right] / 2\sigma^2}, \tag{4.16}$$

onde $v_n^{(I)}$ e $v_n^{(Q)}$ são os componentes em fase e quadratura do símbolo recebido.

As probabilidades de cada símbolo recebido é então inserido em um vetor $\boldsymbol{f}$, o vetor a seguir representa a probabilidade de uma modulação BPSK

$$\boldsymbol{f} = \left[\begin{array}{c:c:c:c:c} g_1^{(0)} & g_2^{(0)} & \cdots & g_{(n-1)}^{(0)} & g_n^{(0)} \\ g_1^{(1)} & g_2^{(1)} & \cdots & g_{(n-1)}^{(1)} & g_n^{(1)} \end{array}\right]. \tag{4.17}$$

Este vetor é utilizado para inicializar a matriz Q, em que cada elemento dessa matriz representa a probabilidade do i-ésimo bit recebido seja 0, $g_i^{(0)}$ ou de que seja 1, $g_i^{(0)}$, para $i = 1, \ldots, n$. Deste modo

$$Q = \left[\begin{array}{c:c:c:c:c} g_1^{(0)} & g_2^{(0)} & \cdots & g_{(n-1)}^{(0)} & g_n^{(0)} \\ g_1^{(1)} & g_2^{(1)} & \cdots & g_{(n-1)}^{(1)} & g_n^{(1)} \\ \hdashline \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hdashline g_1^{(0)} & g_2^{(0)} & \cdots & g_{(n-1)}^{(0)} & g_n^{(0)} \\ g_1^{(1)} & g_2^{(1)} & \cdots & g_{(n-1)}^{(1)} & g_n^{(1)} \end{array}\right]. \tag{4.18}$$

Após inicializar o algoritmo de decodificação com as probabilidades iniciais, iremos executar os demais passos em duas etapas, as quais chamaremos de passo horizontal e o passo vertical.

Iniciando o passo horizontal, as probabilidades são definidas conforme segue

$$r_{mn}(x) = \sum_{\boldsymbol{c}:c_n=x} P(z_m = 0|\boldsymbol{c}) P(\boldsymbol{c}|c_n = x), \tag{4.19}$$

onde dado que o n-ésimo bit $c_n = x$ é um valor fixo então o vetor **c** será composto pelo bit c_n determinado e pelos demais bits restantes que poderão assumir o valor 0 ou 1 em qualquer combinação possível. Entretanto, nem todas as combinações possíveis irão satisfazer a equação de paridade, por exemplo, quando $x = 0$ para que a função de paridade $z_m = 0$ seja válida, iremos considerar apenas as combinações que apresente um número par de bits 1, sendo assim a probabilidade de uma combinação com número ímpar de 1 satisfazer a equação de paridade é zero. Logo, $r_{mn}(x)$ será a soma de todas as probabilidades das sequências binárias $w_r - 1$ que somadas satisfaça seja igual a x, onde $P(z_m = 0|c)$ será zero ou um,

$$r_{mn}(x) = \sum_{\boldsymbol{c}:c_n=x} P(z_m = 0|\boldsymbol{c}) \prod_{n' \in N_m \backslash n} q_{mn'}(x), \tag{4.20}$$

as probabilidades obtidas de $r_{mn}(x)$ podem ser dispostas em uma matriz R, da seguinte maneira

$$R = \left[\begin{array}{c:c:c:c:c} r_{11}(0) & r_{12}(0) & \ldots & r_{1(n-1)}(0) & r_{1n}(0) \\ r_{11}(1) & r_{12}(1) & \ldots & r_{1(n-1)}(1) & r_{1n}(1) \\ \hdashline \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hdashline r_{m1}(0) & r_{m2}(0) & \ldots & r_{m(n-1)}(0) & r_{mn}(0) \\ r_{m1}(1) & r_{m2}(1) & \ldots & r_{m(n-1)}(1) & r_{mn}(1) \end{array}\right], \tag{4.21}$$

onde cada elemento da matriz contém as probabilidades de $r_{mn}(0)$ e $r_{mn}(1)$. Após definir as probabilidades $r_{mn}(x)$, inicia-se o passo vertical, onde as probabilidades de $q_{mn}(x)$ são atualizadas de maneira mais simples que no passo horizontal, por intermédio da Regra de Bayes.

Aplicando a Regra de Bayes na Equação 4.8, podemos escrever a probabilidade $q_{mn}(x)$ conforme segue

$$q_{mn}(x) = P(c_n = x|\{z_m = 0, \quad m' \in M_n/m\}) \tag{4.22}$$

$$= \frac{P(\{z_m = 0, \quad m' \in M_n/m\}|c_n = x)P(c_n = x)}{P(\{z_m = 0, \quad m' \in M_n/m\})}. \tag{4.23}$$

A partir da Equação 4.9 e utilizando a substituição de $P(c_n = x) = f_n^x$, podemos escrever que

$$q_{mn}(x) = \beta_{mn} f_n^x \prod_{m' \in M_n/m} r_{m'n}(x), \tag{4.24}$$

onde β_{mn} é uma constante de normalização tal que $\sum q_{mn}(x) = 1$, sendo assim

$$\beta_{mn} = \frac{1}{\sum_x f_n^x \prod_{m' \in M_n \setminus m} r_{m'n}(x)}. \tag{4.25}$$

Desta forma, a nova matriz Q será,

$$Q = \left[\begin{array}{c:c:c:c:c} q_{11}(0) & q_{12}(0) & \cdots & q_{1(n-1)}(0) & q_{1n}(0) \\ q_{11}(1) & q_{12}(1) & \cdots & q_{1(n-1)}(1) & q_{1n}(1) \\ \hdashline \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hdashline q_{m1}(0) & q_{m2}(0) & \cdots & q_{m(n-1)}(0) & q_{mn}(0) \\ q_{m1}(1) & q_{m2}(1) & \cdots & q_{m(n-1)}(1) & q_{mn}(1) \end{array}\right], \tag{4.26}$$

onde cada elemento $q_{mn}(0)$ e $q_{mn}(1)$ representam as probabilidades de cada possibilidade. Em seguida, é possível calcular a probabilidade pseudo posteriori de $q_n(x)$ como segue

$$q_n(x) = \beta_n f_n^x \prod_{m \in M_n} r_{mn}(x), \tag{4.27}$$

novamente realizamos a normalização de $\sum q_n(x) = 1$ como β_n. Cada probabilidade da pseudo posteriori é disposto em um vetor Q'

$$Q' = \left[\begin{array}{c:c:c:c:c} q_1(0) & q_2(0) & \cdots & q_{(n-1)}(0) & q_n(0) \\ q_1(1) & q_2(1) & \cdots & q_{(n-1)}(1) & q_n(1) \end{array}\right]. \tag{4.28}$$

Finalmente, a partir das probabilidades da pseudo posteriori, conseguimos estimar a palavra código

transmitida como

$$\hat{\mathbf{c}}_{\mathbf{n}} = \arg\max_{x} \beta_n f_n^x \prod_{m \in M_n} r_{mn}(x). \tag{4.29}$$

A complexidade da decodificação soma-produto está majoritariamente ligado ao passo horizontal, devido a quantidade de combinações binárias possíveis que cresce de acordo com o tamanho do código. Existem algumas técnicas que reduzem a complexidade do algoritmo soma-produto, como por exemplo a utilização da Transformada Rápida de Fourrier [CARRASCO e JOHNSTON 2008].

A seguir apresentamos um exemplo do funcionamento do SPA de modo a deixar mais claro o funcionamento do algoritmo.

Exemplo 4.3.2 *Seja C um código LDPC regular com parâmetros $(n, w_c, w_r) = (10, 3, 6)$. Uma possível matriz controle de paridade para este código é*

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

Considere que a palavra transmitida foi $\mathbf{c} = (0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1)$ *e a palavra recebida foi* $\mathbf{v} = (0\,0\,0\,1\,1\,1\,0\,1\,0\,1)$*, ou seja, a palavra recebida apresenta apenas um erro na quinta posição do vetor recebido. Sendo as probabilidades iniciais $\boldsymbol{f}$ dada por*

$$\boldsymbol{f} = \begin{bmatrix} 0.78 & 0.84 & 0.81 & 0.52 & 0.45 & 0.13 & 0.82 & 0.21 & 0.75 & 0.24 \\ 0.22 & 0.16 & 0.19 & 0.48 & 0.55 & 0.87 & 0.18 & 0.79 & 0.25 & 0.76 \end{bmatrix}. \tag{4.30}$$

A matriz Q é inicializada conforme a seguir

$$Q = \left[\begin{array}{cccccccccc} 0.78 & 0.84 & 0.81 & 0 & 0 & 0.13 & 0.82 & 0 & 0 & 0.24 \\ 0.22 & 0.24 & 0.19 & 0 & 0 & 0.87 & 0.18 & 0 & 0 & 0.76 \\ \hdashline 0.78 & 0 & 0.81 & 0 & 0.45 & 0.13 & 0 & 0.21 & 0.75 & 0 \\ 0.22 & 0 & 0.19 & 0 & 0.55 & 0.87 & 0 & 0.79 & 0.25 & 0 \\ \hdashline 0 & 0 & 0.81 & 0.52 & 0.45 & 0 & 0.82 & 0 & 0.75 & 0.24 \\ 0 & 0 & 0.19 & 0.48 & 0.55 & 0 & 0.18 & 0 & 0.25 & 0.76 \\ \hdashline 0 & 0.84 & 0 & 0.52 & 0.45 & 0.13 & 0 & 0.21 & 0 & 0.24 \\ 0 & 0.24 & 0 & 0.48 & 0.55 & 0.87 & 0 & 0.79 & 0 & 0.76 \\ \hdashline 0.78 & 0.84 & 0 & 0.52 & 0 & 0 & 0.82 & 0.21 & 0.75 & 0 \\ 0.22 & 0.24 & 0 & 0.48 & 0 & 0 & 0.18 & 0.79 & 0.25 & 0 \end{array}\right] \tag{4.31}$$

A partir da matriz obtida, podemos iniciar o passo horizontal para determinar as probabilidades $r_{mn}(x)$ e então a matriz R. Para calcular a probabilidade $r_{11}(1)$ é necessário utilizar a equação de paridade a seguir

$$c_1 + c_2 + c_3 + c_6 + c_7 + c_{10} = 0. \tag{4.32}$$

Logo, utilizaremos as informações extrínsecas ao nó c_1, de modo que é necessário calcular as probabilidades de todas as combinações possíveis para que a equação de paridade seja satisfeita quando $c_1 = 1$, ou seja, será necessário determinar c_2, c_3, c_6, c_7 e c_{10} afim de satisfazer

$$c_2 + c_3 + c_6 + c_7 + c_{10} = c_1 = 1. \tag{4.33}$$

Conforme citado anteriormente, ao utilizar o alfabeto binário, apenas combinações ímpares irão satisfazer a equação acima, logo teremos 16 combinações possíveis e probabilidade de $r_{11}(1)$ será

$$\begin{aligned}
r_{11}(1) = {} & (q_{12}(1)q_{13}(0)q_{16}(0)q_{17}(0)q_{1,10}(0)) + (q_{12}(0)q_{13}(1)q_{16}(0)q_{17}(0)q_{1,10}(0)) + \\
& + (q_{12}(0)q_{13}(0)q_{16}(1)q_{17}(0)q_{1,10}(0)) + (q_{12}(0)q_{13}(0)q_{16}(0)q_{17}(1)q_{1,10}(0)) + \\
& + (q_{12}(0)q_{13}(0)q_{16}(0)q_{17}(0)q_{1,10}(1)) + (q_{12}(1)q_{13}(1)q_{16}(1)q_{17}(0)q_{1,10}(0)) + \\
& + (q_{12}(0)q_{13}(1)q_{16}(1)q_{17}(1)q_{1,10}(0)) + (q_{12}(0)q_{13}(0)q_{16}(1)q_{17}(1)q_{1,10}(1)) + \\
& + (q_{12}(1)q_{13}(1)q_{16}(0)q_{17}(0)q_{1,10}(1)) + (q_{12}(1)q_{13}(1)q_{16}(0)q_{17}(1)q_{1,10}(0)) + \\
& + (q_{12}(0)q_{13}(1)q_{16}(1)q_{17}(0)q_{1,10}(1)) + (q_{12}(1)q_{13}(0)q_{16}(1)q_{17}(0)q_{1,10}(1)) + \\
& + (q_{12}(1)q_{13}(0)q_{16}(0)q_{17}(1)q_{1,10}(1)) + (q_{12}(0)q_{13}(1)q_{16}(0)q_{17}(1)q_{1,10}(1)) + \\
& + (q_{12}(1)q_{13}(0)q_{16}(1)q_{17}(1)q_{1,10}(0)) + + (q_{12}(1)q_{13}(1)q_{16}(1)q_{17}(1)q_{1,10}(1)) \\
& = 0.00316 + 0.004038 + 0.116495 + 0.003821 + 0.055123 + 0.005205 + \\
& + 0.005998 + 0.080978 + 0.002463 + 0.000171 + 0.086533 + 0.070267 + \\
& + 0.002305 + 0.002838 + 0.004871 + 0.0036618 = 0.448086.
\end{aligned}$$

Após obter a probabilidade de $r_{11}(1) = 0.448086$ e a probabilidade de $r_{11}(0) = 1 - r_{11}(1) = 0.551914$, este procedimento é repetido para todo r_{mn}, e os valores serão dispostos na matriz R

$R =$

$$\left[\begin{array}{cccccccccc}
0.551914 & 0.542753 & 0.546890 & 0 & 0 & 0.460714 & 0.545425 & 0 & 0 & 0.444092 \\
0.448086 & 0.457247 & 0.453110 & 0 & 0 & 0.539286 & 0.454575 & 0 & 0 & 0.555908 \\
\hdashline
0.493347 & 0 & 0.493991 & 0 & 0.537255 & 0.505034 & 0 & 0.506423 & 0.493347 & 0 \\
0.506653 & 0 & 0.506009 & 0 & 0.462745 & 0.494966 & 0 & 0.493577 & 0.507451 & 0 \\
\hdashline
0 & 0 & 0.500333 & 0.505158 & 0.497937 & 0 & 0.500322 & 0 & 0.500413 & 0.499603 \\
0 & 0 & 0.499667 & 0.494842 & 0.502063 & 0 & 0.499678 & 0 & 0.499587 & 0.500397 \\
\hdashline
0 & 0.500446 & 0 & 0.507588 & 0.496965 & 0.499590 & 0 & 0.499477 & 0 & 0.499416 \\
0 & 0.499554 & 0 & 0.492412 & 0.503035 & 0.500410 & 0 & 0.500523 & 0 & 0.500584 \\
\hdashline
0.497476 & 0.497921 & 0 & 0.464662 & 0 & 0 & 0.497791 & 0.502437 & 0.497173 & 0 \\
0.502524 & 0.502079 & 0 & 0.535338 & 0 & 0 & 0.502209 & 0.497563 & 0.502827 & 0
\end{array}\right].$$

Em seguida o passo vertical irá atualizar a matriz inicial Q conforme a Equação 4.27

$$q_{11}(0) = \beta_{11} f_1^0 \prod_{m' \in M_1 \setminus 1} r_{m'1}(0)$$
$$= \beta_{11} \times 0.78 \times (0.493347 \times 0.497476) = 0.191434\beta_{11}.$$

$$q_{11}(1) = \beta_{11} f_1^1 \prod_{m' \in M_1 \setminus 1} r_{m'1}(1)$$
$$= \beta_{11} \times 0.22 \times (0.506653 \times 0.502524) = 0.056013\beta_{11}.$$

Visto que $0.191434\beta_{11} + 0.056013\beta_{11} = 1$*,*

$$\beta_{11} = \frac{1}{0.191334 + 0.056013} = 4.042903.$$

Logo,

$$q_{11}(0) = 4.042903 \times 0.78 \times (0.493347 \times 0.497476) = 0.773636.$$
$$q_{11}(1) = 4.042903 \times 0.22 \times (0.506653 \times 0.502524) = 0.226364.$$

A partir dessa operação é possível obter os valores atualizados de todos os elementos da matriz, e desta forma atualizá-la matriz. A matriz Q atualizada será,

$Q =$

$$\left[\begin{array}{cccccccccc}
0.773636 & 0.839121 & 0.806481 & 0 & 0 & 0.132106 & 0.818884 & 0 & 0 & 0.239285 \\
0.226364 & 0.160879 & 0.193519 & 0 & 0 & 0.867894 & 0.181116 & 0 & 0 & 0.760715 \\
\hdashline
0.812140 & 0 & 0.8374611 & 0 & 0.444958 & 0.113039 & 0 & 0.211273 & 0.748185 & 0 \\
0.187860 & 0 & 0.162539 & 0 & 0.555042 & 0.886961 & 0 & 0.788727 & 0.251815 & 0 \\
\hdashline
0 & 0 & 0.833978 & 0.492203 & 0.484126 & 0 & 0.844187 & 0 & 0.742212 & 0.201076 \\
0 & 0 & 0.166022 & 0.507797 & 0.515874 & 0 & 0.155813 & 0 & 0.257788 & 0.798924 \\
\hdashline
0 & 0.860727 & 0 & 0.489773 & 0.485097 & 0.115241 & 0 & 0.215940 & 0 & 0.201196 \\
0 & 0.139273 & 0 & 0.510227 & 0.514903 & 0.884759 & 0 & 0.784060 & 0 & 0.798804 \\
\hdashline
0.809608 & 0.861934 & 0 & 0.532711 & 0 & 0 & 0.845514 & 0.213942 & 0.744684 & 0 \\
0.190392 & 0.138066 & 0 & 0.467289 & 0 & 0 & 0.154486 & 0.786058 & 0.255316 & 0
\end{array}\right].$$

Enfim, podemos calcular a probabilidade posteriori para então estimar a palavra transmitida a partir da Equação 4.28

$$q_1(0) = \beta_1 f_1^0 \prod_{m \in M_1} r_{m1}(0) \quad = \beta_1 \times 0.78 \times (0.551914 \times 0.493347 \times 0.497476) = 0.105655\beta_1.$$
$$q_1(1) = \beta_1 f_1^1 \prod_{m \in M_1} r_{m1}(1) \quad = \beta_1 \times 0.22 \times (0.448086 \times 0.506653 \times 0.502524) = 0.025099\beta_1.$$

Teremos

$$\beta_{11} = \frac{1}{0.105655 + 0.025099} = 7.64797,$$

então

$$q_1(0) = 7.64797 \times 0.78 \times (0.551914 \times 0.493347 \times 0.497476) = 0.808046,$$
$$q_1(1) = 7.64797 \times 0.22 \times (0.448086 \times 0.506653 \times 0.502524) = 0.191956.$$

Empregando a Equação 4.29, é possível identificar que o o primeiro bit será $\hat{c}_1 = 0$*, pois é o que apresenta a maior probabilidade a posteriori. Os demais símbolos apresentam as seguintes probabilidades*

$$Q' = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c} 0.808046 & 0.860941 & 0.834162 & 0.497361 & 0.482065 & 0.115074 & 0.844356 & 0.215586 & 0.742528 & 0.200821 \\ 0.191954 & 0.139059 & 0.165838 & 0.502639 & 0.517935 & 0.884926 & 0.155644 & 0.784414 & 0.257472 & 0.799179 \end{array}\right].$$

Sendo assim, a palavra-código será $\hat{\mathbf{c}} = (0\,0\,0\,1\,1\,1\,0\,1\,0\,1)$*, é possível verificar que a palavra ainda não foi corrigida, o erro continua no quinto bit, sendo assim, será necessário mais uma iteração para tentar corrigi-lo. A segunda iteração é iniciada com a última matriz* Q *obtida, ou seja, a mais atualizada, e os demais passos são seguidos. Para este exemplo, a palavra só será corrigida após a terceira iteração, suprimindo os passos intermediários, as matrizes finais* R, Q *e* Q' *serão,*

$R =$

$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c} 0.549960 & 0.540086 & 0.544369 & 0 & 0 & 0.463092 & 0.542650 & 0 & 0 & 0.447890 \\ 0.450040 & 0.459914 & 0.455631 & 0 & 0 & 0.536908 & 0.457350 & 0 & 0 & 0.552110 \\ \hline 0.493114 & 0 & 0.493650 & 0 & 0.545393 & 0.505532 & 0 & 0.507453 & 0.491301 & 0 \\ 0.506886 & 0 & 0.506350 & 0 & 0.454607 & 0.494468 & 0 & 0.492547 & 0.508699 & 0 \\ \hline 0 & 0 & 0.499989 & 0.500176 & 0.502649 & 0 & 0.499989 & 0 & 0.499985 & 0.500012 \\ 0 & 0 & 0.500011 & 0.499824 & 0.497351 & 0 & 0.500011 & 0 & 0.500015 & 0.499988 \\ \hline 0 & 0.499975 & 0 & 0.500415 & 0.503915 & 0.500023 & 0 & 0.500032 & 0 & 0.500030 \\ 0 & 0.500025 & 0 & 0.499585 & 0.496085 & 0.499977 & 0 & 0.499968 & 0 & 0.499970 \\ \hline 0.496595 & 0.497094 & 0 & 0.457904 & 0 & 0 & 0.496955 & 0.503693 & 0.495668 & 0 \\ 0.503405 & 0.5502906 & 0 & 0.542096 & 0 & 0 & 0.503045 & 0.496307 & 0.504332 & 0 \end{array}\right],$$

$Q =$

$$\left[\begin{array}{cccccccccc}
0.772854 & 0.838418 & 0.806053 & 0 & 0 & 0.132534 & 0.818189 & 0 & 0 & 0.240031 \\
0.227146 & 0.161582 & 0.193947 & 0 & 0 & 0.867466 & 0.181811 & 0 & 0 & 0.759969 \\
\hdashline
0.810391 & 0 & 0.835883 & 0 & 0.456507 & 0.114177 & 0 & 0.212482 & 0.746725 & 0 \\
0.189609 & 0 & 0.164117 & 0 & 0.543493 & 0.885823 & 0 & 0.787518 & 0.253275 & 0 \\
\hdashline
0 & 0 & 0.832375 & 0.478244 & 0.499266 & 0 & 0.842265 & 0 & 0.740099 & 0.203955 \\
0 & 0 & 0.167625 & 0.521756 & 0.500734 & 0 & 0.157735 & 0 & 0.259901 & 0.796045 \\
\hdashline
0 & 0.859034 & 0 & 0.478005 & 0.498000 & 0.116425 & 0 & 0.217493 & 0 & 0.203943 \\
0 & 0.140966 & 0 & 0.521995 & 0.502000 & 0.883575 & 0 & 0.782507 & 0 & 0.796057 \\
\hdashline
0.808242 & 0.860424 & 0 & 0.520590 & 0 & 0 & 0.843871 & 0.215010 & 0.743407 & 0 \\
0.191758 & 0.139576 & 0 & 0.479410 & 0 & 0 & 0.156129 & 0.784990 & 0.256593 & 0
\end{array}\right]$$

e

$$Q' = \left[\begin{array}{cccccccccc}
0.806122 & 0.859023 & 0.832369 & 0.478419 & 0.501915 & 0.116434 & 0.842260 & 0.217514 & 0.740088 & 0.203963 \\
0.193878 & 0.140977 & 0.167631 & 0.521581 & 0.498085 & 0.883566 & 0.157740 & 0.782486 & 0.259913 & 0.796037
\end{array}\right].$$

Portanto, após três iterações conseguimos recuperar a palavra-código

$$\hat{\mathbf{c}} = (0\,0\,0\,1\,0\,1\,0\,1\,0\,1).$$

5 DESEMPENHO DE CÓDIGOS LDPC

Neste capítulo serão apresentados os resultados das simulações envolvendo os códigos LDPC com o objetivo de verificar a resposta do desempenho desta estratégica de codificação de canal, FEC, em função da BER a diversos parâmetros, como o método de obtenção da matriz controle de paridade, tamanho do código e parâmetros da decodificação iterativa.

Antes de iniciar as simulações, foi necessário buscar um modelo de canal representasse nosso canal óptico, deste modo considerou-se um sistema WDM de múltiplos canais, as simulações representam a análise de um desses canais, onde a iteração dos demais canais geram a mistura de quatro ondas (*Four-Wave Mixing*, FWM) e a modulação de fase cruzada (*cross-phase modulation*, XPM), causando uma interferência não-linear (*non-linear interference*, NLI). O modelo que considera o ruído gaussiano (*gaussian noise*, GN) provou ser relativamente simples e, ao mesmo tempo,ferramenta suficientemente confiável para previsão de desempenho ao longo de uma ampla gama de cenários. Deste modo, considerando algumas suposições do canal sem memória onde a saída atual não depende da saída anterior dado que as entradas são estatisticamente independentes, e um ruído branco local (*Locally-White Noise*, LWN), o canal considerado coincide com o conhecido canal AWGN quando considera-se que a NLI pode ser representada como um ruído gaussiano aditivo, e seu desempenho em termos da BER pode ser encontrado a partir das fórmulas analíticas [POGGIOLINI et al. 2014].

Consideremos $C(n, k)$ um código LPDC regular com pesos w_c e w_r de 1's nas colunas e linhas, respectivamente. Os testes apresentados neste trabalho foram realizados utilizando 3 parâmetros distintos de (n, k) de códigos LDPC regulares, onde cada código possui pesos w_c e w_r característicos. Esses parâmetros, assim como as suas respectivas taxas, estão dispostos na tabela a seguir [ELEFTHERIOU e ÖLÇER 2001].

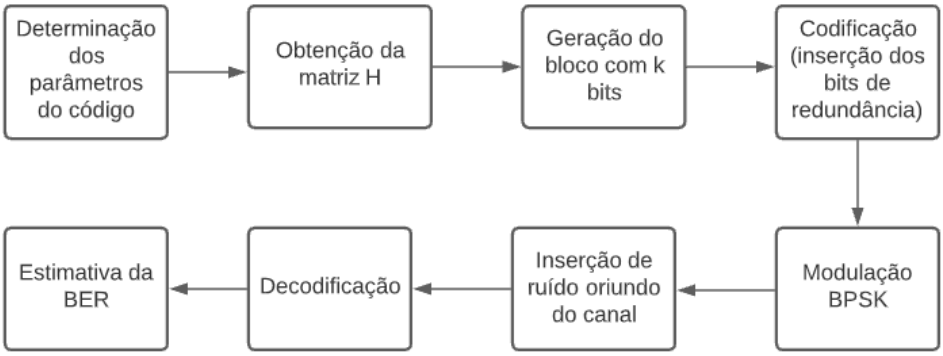
Tabela 9 – Parâmetros dos códigos LDPC

C(n, k)	w_c	w_r	Taxa k/n
(529,462)	3	23	0.8733
(1369,1260)	3	37	0.9204
(4489,4158)	5	67	0.9263

As simulações foram executadas conforme exemplificado no diagrama de blocos da figura Figura 18

Para os códigos LDPC com parâmetros apresentados na Tabela 9, temos como objetivo determinar a matriz controle de paridade H a fim de maximizar seu desempenho em termos da BER. Como os códigos LDPC são códigos do tipo bloco, a sua codificação pode ser realizada por intermédio de uma matriz geradora G como apresentado em (3.10). Conforme a Definição 3.2.8, a partir da matriz verificação de paridade escrita na forma sistemática é possível obter uma matriz geradora também na forma padrão, que será responsável pela codificação. Entretanto, para códigos LDPC, dado que a matriz H apresenta poucos elementos não-nulos, o número de iterações necessárias para colocá-la em sua forma padrão de modo que a matriz G seja obtida pode ser muito elevada, gerando um alto custo

Figura 18 – Diagrama de blocos das simulacões



fonte: Próprio Autor

computacional. Além da possibilidade de aumentar a densidade de elementos não-nulos da matriz H, fazendo com que perca sua característica principal de ser esparsa. Por esses motivos visando diminuir a complexidade da simulação e poder analisar a eficiência dos códigos estudados em relação a matriz controle de paridade H construída, para todas as simulações o vetor de informação $\mathbf{u}$ a ser codificado com dimensão k será o vetor nulo [SANTOS 2014, LULU 2012].

Após a inserção dos bits redundantes, o vetor $\mathbf{v}$ com dimensão n passa por uma modulação BPSK antes de ser transmitido pelo canal AWGN. A modulação BPSK foi escolhida por diminuir a complexidade do algoritmo de detecção e correção de erros, visto que como no caso do algoritmo soma-produto apresentado na Subseção 4.3.2, a inicialização do decodificador ser as probabilidade de receber um determinado símbolo. Para a modulação BPSK o decodificador recebe duas probabilidades $g_n^{(0)}$ e $g_n^{(1)}$ dadas pelas Equações 4.11 e 4.12, enquanto que para a modulação QPSK o decodificador receberia probabilidades referentes aos quatro símbolos possíveis $g_n^{(00)}$, $g_n^{(01)}$, $g_n^{(10)}$, $g_n^{(11)}$ conforme as Equações 4.13, 4.14, 4.15 e 4.16.

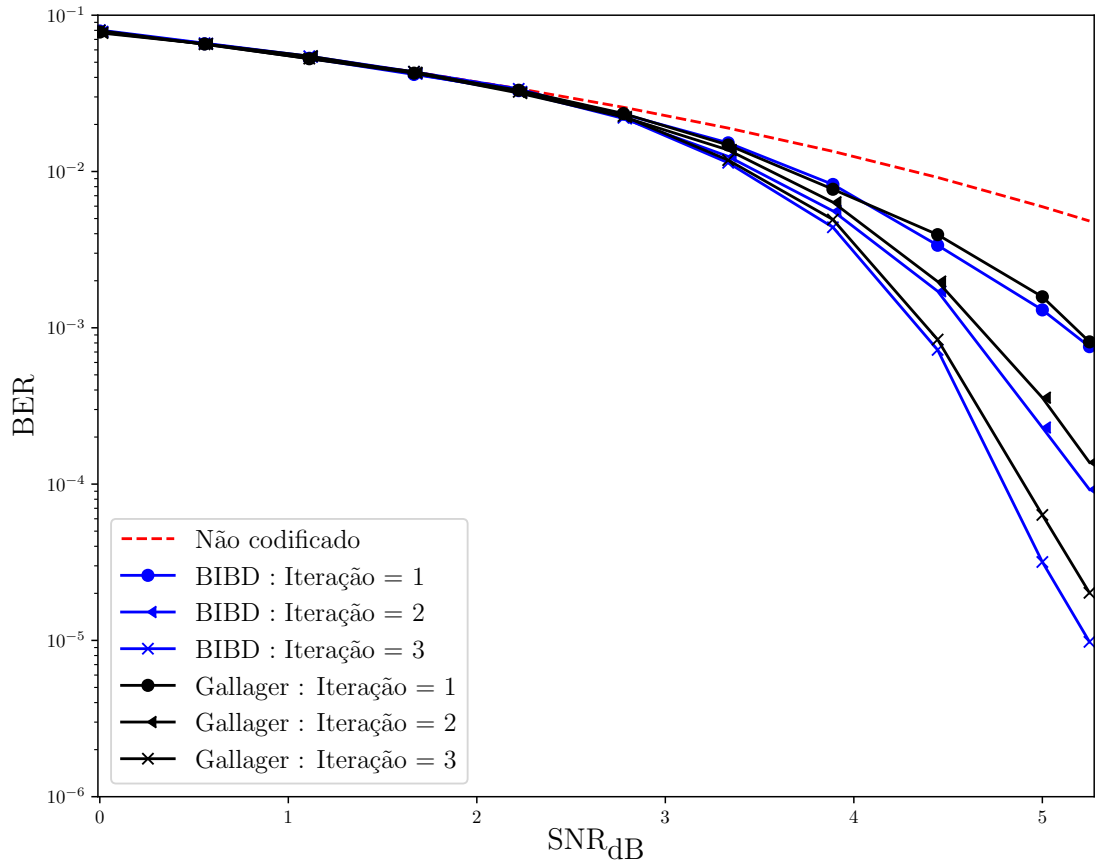
Apesar de implementar a modulação BPSK, todos os resultados obtidos podem ser estendidos à modulação QPSK, visto que a probabilidade de erro de ambas as modulações são iguais. Ao verificar as equações das probabilidades iniciais descritas na Subseção 4.3.2, verifica-se que a probabilidade para os símbolos da modulação QPSK é a soma de duas probabilidades BPSK, este fato se dá através da ortogonalidade dos sinais enviados. Em uma modulação QPSK os sinais são defasados em $90°$ e não em $180°$ como no BPSK, ou seja, em uma mesma banda em que uma modulação BPSK é transmitida, ao utilizar a modulação QPSK a taxa de bits transmitidos será o dobro [RIBEIRO].

A análise inicial se deu a cerca da influência do método escolhido para obtenção da matriz verificadora de paridade. Conforme citado no Capítulo 4, uma matriz controle de paridade pode ser obtida de inúmeras formas. Neste trabalho foram apresentados dois métodos: o tradicional proposto por Gallager (apresentado na Subseção 4.2.1), e o método que emprega o conceito de design combinatório denominado como BIBD (apresentado na Sebseção 4.2.2). Ao projetar um código LDPC, busca-se uma matriz H com o maior *girth* possível, pois como vimos, principalmente em códigos curtos ocorre degradação da performance na presença de um *girth* de tamanho 4 [CARRASCO e JOHNSTON 2008].

Para comparar a duas estratégias de codificação, foi realizada uma simulação utilizando o código de parâmetros $C(1369, 1260)$ com $w_c = 3$ e $w_r = 37$, como descrito na Tabela 9. De modo a estabelecer

uma comparação entre os métodos, foram utilizadas 3 iterações no decodificador aplicando o algoritmo Soma-Produto e uma SNR variando de 0 a 5.25 dB. O resultado obtido está apresentado na Figura 19.

Figura 19 – Simulação da BER para um código LPDC $C(1369, 1260)$ obtido utilizando os métodos de Gallager e BIBD



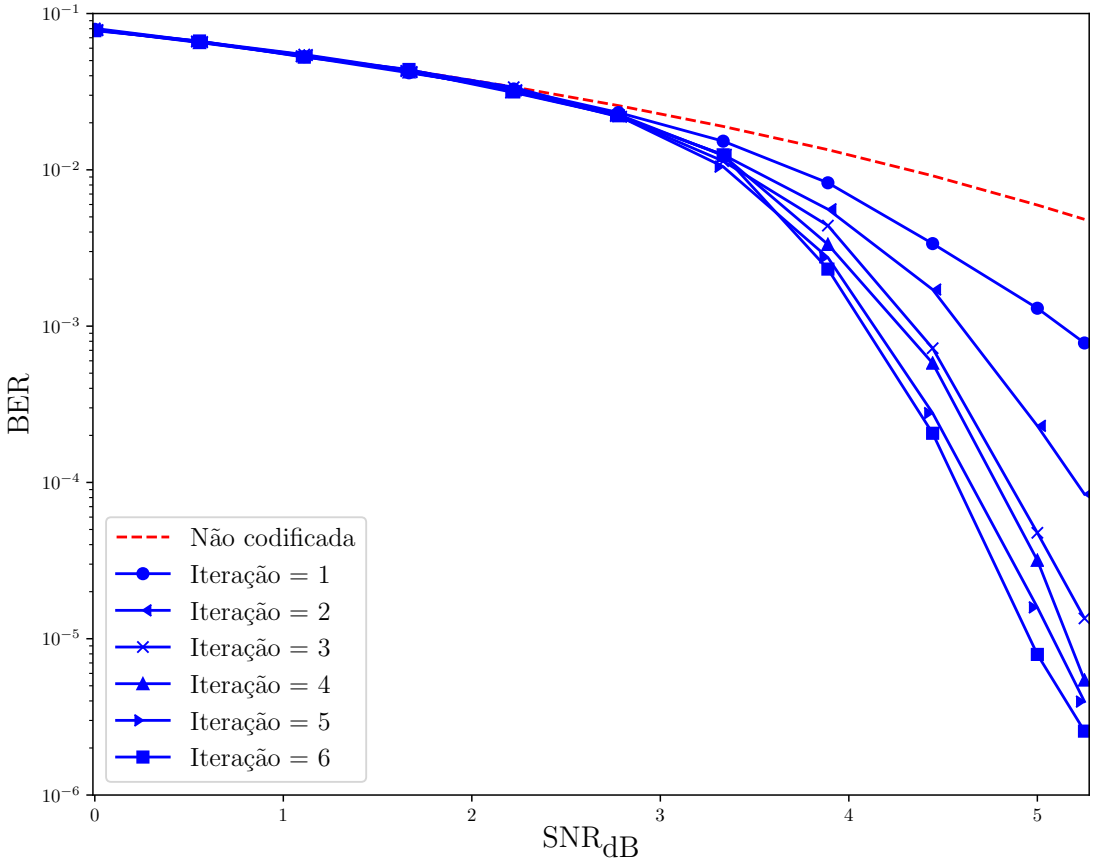
fonte: Próprio Autor

A partir da Figura 19, verifica-se que o desempenho do método BIBD mostra-se superior a Gallager. Ao investigar o motivo dessa melhora, a partir das duas matrizes controle de paridade H obtidas nos dois métodos foi implementado um algoritmo para identificar a presença de *girths* de tamanho 4. O algoritmo baseia-se em analisar o principal requisito para evitar *girth* de tamanho 4, onde verificam-se as colunas de H e checa-se se quaisquer duas colunas de H apresentam valores não-nulos em mais de uma posição em comum. A partir do algoritmo implementado, verificou-se que o método proposto por Gallager apresentou ciclos de tamanho 4, ao contrário do método BIBD que não apresentou ciclos de tamanho 4. Ao analisar os demais códigos apresentados na Tabela 9, $C(529, 462)$ e $C(4489, 4158)$, observou-se o mesmo resultado anterior. As matrizes obtidas pelo método BIBD se mostram livres de *girths* de tamanho 4, e o método proposto por Gallager apresentou *girth* de tamanho 4 em todos os códigos. Por este motivo o método BIBD foi a estratégia de codificação para códigos LDPC escolhida para realizar as demais simulações deste trabalho.

Após definir o método BIBD como estratégia para construir a matriz controle de paridade H, realizou-se um teste para verificar a influência do número de iterações aplicadas ao decodificador

SPA, buscando um valor ótimo que apresente um desempenho satisfatório sem que haja um custo computacional elevado em termos de tempo de processamento, o resultado está representado na Figura 20. O código LDPC simulado tem parâmetros $C(1369, 1260)$ e a SNR varia entre 0 e 5.25 dB.

Figura 20 – Influência do número de iterações no algoritmo de decodificação SPA para um código LDPC $C(1369, 1260)$



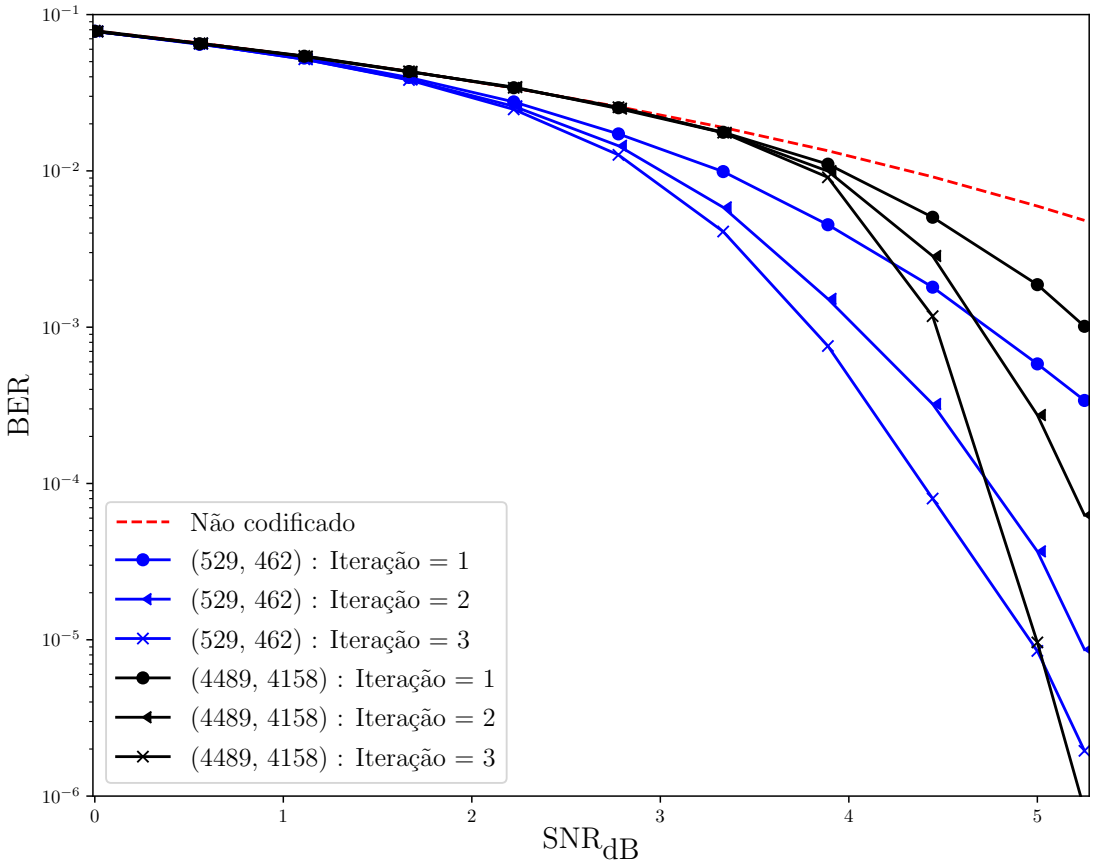
fonte: Próprio Autor

O algoritmo de decodificação SPA, utiliza as informações extrínsecas de um nó para atualizar a certeza sobre um determinado bit recebido. Conforme ocorre cada iteração, a probabilidade sobre essa certeza é refinada, aumentando o desempenho do código. Deste modo, quanto maior o número de iterações, mais vezes a probabilidade será atualizada e menor será a BER. Em contrapartida, quanto mais iterações são simuladas, maior deverá ser a SNR bem como, a quantidade de pontos simulados para atingir um número suficiente de erros a fim de estimar corretamente a BER, o que aumenta de maneira expressiva a latência da simulação para o método de paralelização utilizado. Nota-se a partir dos resultados apresentados na Figura 20 que os maiores ganhos em desempenho se dão nas 3 primeiras iterações, ao aumentar o número de iterações o código tende a chegar a um limite para os parâmetros utilizados. Para essa simulação o número ótimo de iterações foi 6, que para uma SNR de 5.25dB atingiu uma BER de 2.563×10^{-6}, conforme pode ser observado na Figura 20.

Outros parâmetros que influenciam diretamente o desempenho de um código LDPC são: o tamanho n do código e a $n - k$ quantidade de bits redundantes em relação a informação. Para checar essa

influência foram simulados dois códigos distintos, o código com parâmetros $C(529, 462)$ com tamanho 529 e número de bits redundantes 67 ou seja, em relação a quantidade de bits de informação é adicionado 14.5% bits de redundância e o código com parâmetros $C(4489, 4158)$ com tamanho 4489 e número de bits redundantes 331, deste modo, os bits redundantes representam 7.96% em relação a informação. Ambos os códigos foram aplicados a uma SNR também variando de 0 a 5.25 dB e considerando 3 iterações na decodificação com o algoritmo SPA, as curvas obtidas estão dispostas na Figura 21.

Figura 21 – Comparação da BER entre códigos LDPC com parâmetros $C(529, 462)$ e $C(4489, 4158)$

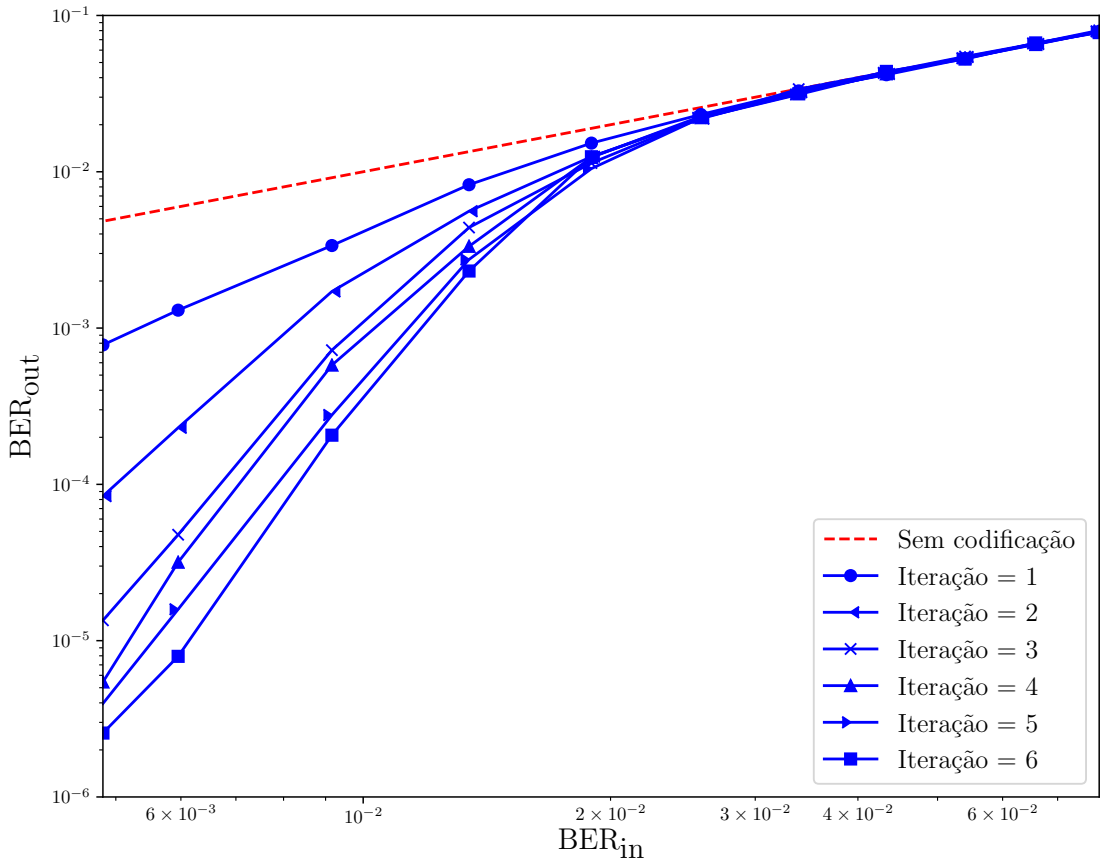


fonte: Próprio Autor

A partir dos dados da BER obtidos na simulação apresentada na Figura 20 foi possível analisar o ganho de codificação em função do número de iterações aplicadas a decodificação SPA, esta análise é realizada comparando a BER do canal não codificado, nesse caso representado por BER$_{in}$ e a BER do canal codificado BER$_{out}$. A Figura 22 traz o resultado gráfico desta simulação. Ao observar o decaimento das curvas na Figura 22 o ganho de codificação, ou seja, a melhora da BER de um canal sem codificação em relação ao canal codificado, pode ser observado de maneira que, ao fixar um valor de BER$_{in}$ e observar o valor correspondente de BER$_{out}$, essa diferença demonstra o quanto o desempenho da BER melhorou após a aplicação da FEC baseada em LDPC.

Este ganho de codificação pode ser calculado em dB, de maneira a facilitar o entendimento do valor do ganho. Para isso a partir da Equação 2.9 e dos dados obtidos a partir da simulação que resultou

Figura 22 – Evolução de performance para cada iteração de um código LDPC $C(1369, 1260)$



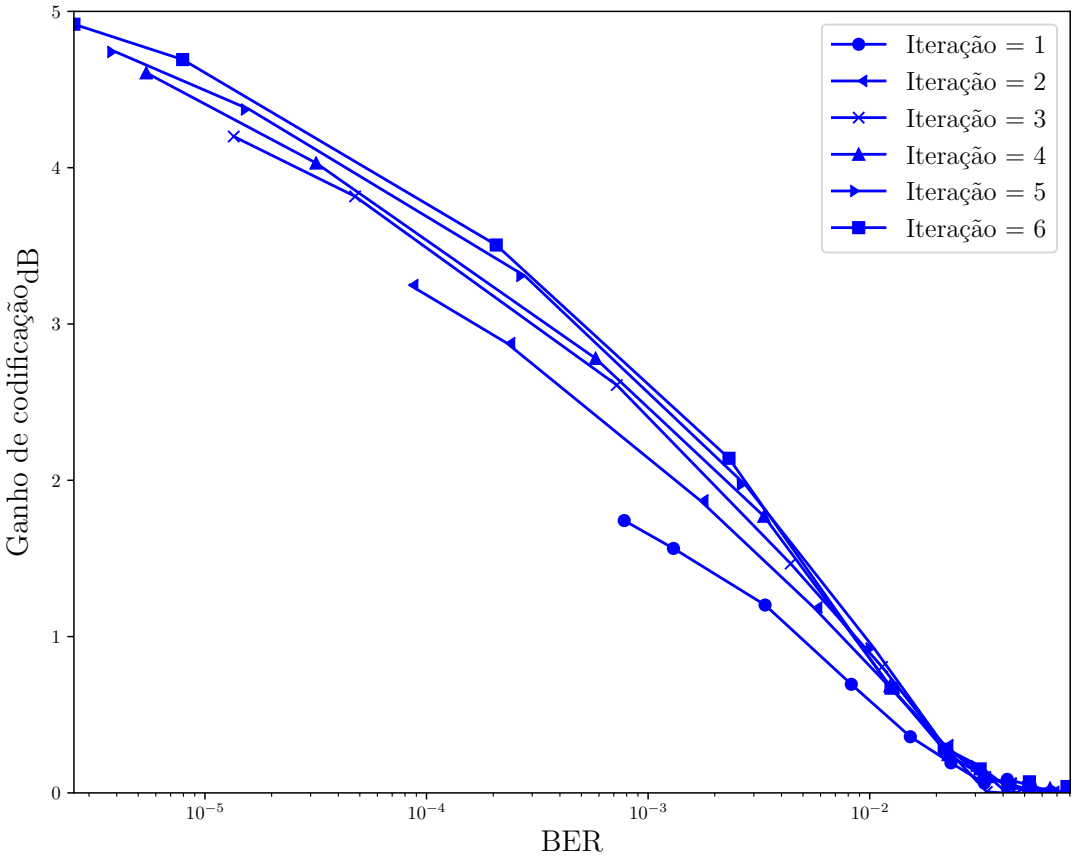
fonte: Próprio Autor

em Figura 22 , foi possível calcular o ganho de codificação CG de modo que fosse possível expor de maneira mais clara o ganho em dB para cada iteração aplicada ao decodificador SPA. Deste modo obtivemos o resultado mostrado na Figura 23. Com este resultado demonstra-se que a utilização do método FEC baseado em LDPC afeta de maneira muito positiva o desempenho do sistema, visto que para a segunda iteração e uma BER de aproximadamente 10^{-4} o ganho do código foi acima de 3 dB, ou seja, para atingir uma mesma BER um canal codificado precisa de metade da potência de SNR quando comparado a um canal sem codificação.

O maior ganho de codificação, conforme esperado, se deu na iteração 6 onde foi possível avaliar que para uma BER de 2.563×10^{-6} o ganho do código LDPC em comparação com o canal sem codificação foi de 4.91 dB. Para verificar o ganho de codificação líquida é necessário considerar a taxa de código, neste caso para o código utilizado nesta implementação $C(1369, 1260)$ sua taxa é $R_C = k/n = 0.9204$. Dessa forma, a partir da Equação 2.10, o ganho liquido para esta codificação foi de

$$NCG = 4.91 + (-0.3602) = 4.5498\ dB.$$

Figura 23 – Ganho de codificação para cada iteração de um código LDPC $C(1369, 1260)$



fonte: Próprio Autor

6 CONCLUSÃO

No presente trabalho foi apresentado um estudo sobre códigos LDPC regulares, apresentando suas características, métodos de obtenção da matriz controle de paridade, além de métodos de codificação e decodificação. Os parâmetros e desempenho foram analisados a fim de verificar seu comportamento para sistemas de comunicações ópticas coerentes.

Os códigos LPDC foram propostos por Gallager e são caracterizados por conter uma matriz controle de paridade H esparsa, ou seja, com baixa densidade de elementos não nulos. Podem ser definidos como códigos regulares ou irregulares. Para códigos regulares, explorou-se métodos aplicáveis capazes de gerar uma matriz H esparsa como o método definido inicialmente por Gallager e ainda um método baseado em design combinatório denominado BIBD. Para a decodificação exemplificou-se um método de decodificação por *hard-decision*, o *Bit-Flipping* e por *soft-decidion*, o algoritmo soma-produto SPA.

O desempenho dos códigos LDPC foram simulados computacionalmente, utilizando um modelo de canal gaussiano onde as não linearidades do canal eram representadas através de um ruído gaussiano aditivo, ao considerar um canal sem memória e um ruído branco local, o comportamento desse canal coincide para um canal AWGN, onde seu desempenho em termos da BER pode ser obtido por meio de fórmulas analíticas. A determinação da modulação que foi utilizada levou em consideração a decodificação através do SPA, dado que o decodificador utiliza as probabilidades de um símbolo apresentar um determinado valor, por motivos de simplicidade da decodificação a modulação utilizada foi o BPSK.

A simulação inicial foi realizada com um código $C(1369, 1260)$ e 3 iterações no decodificador SPA a fim de averiguar a eficiência em termos da BER quanto ao método utilizado para obter a matriz H, ao testar dois métodos de obtenção da matriz, sendo o método proposto por Gallager e o método BIBD, verificou-se um melhor desempenho quando a matriz de paridade foi obtida através do método BIBD. Para verificar o motivo deste desempenho superior, implementou-se um algoritmo que através da matriz controle de paridade obtida era possível verificar a presença de *girths* igual a 4. O método BIBD se mostrou ser bastante eficiente nesse quesito, pois ao analisar a matriz de paridade dos 3 tamanhos de códigos utilizados, nenhum deles apresentou *girth* de valor 4, ao contrário do método proposto por Gallager. Em seguida, após definir o método de obtenção da matriz H através do BIBD, foi realizada uma simulação para explorar qual seria o efeito do número de iterações utilizadas no algoritmo SPA no desempenho da BER, a simulação foi realizada visando atingir uma BER mínima de 10^{-6}, sem que a simulação apresentasse uma alta latência na obtenção do resultado, esse desempenho foi encontrado ao aplicar uma SNR de 5.25 dB e 6 iterações no algoritmo soma-produto. É necessário levar em consideração que ao elevar o número de iterações, a latência gerada para alcançar os resultados aumenta consideravelmente, além de que após um determinado número de iterações o desempenho não sofrerá melhora significativa.

Para analisar a influência do tamanho da palavra código e da quantidade de bits redundantes, foram simulados dois códigos de tamanho distintos, o código $C(529, 462)$ com 14.5% de bits de redundância e o código $C(4489, 4158)$ com 7.96% de bits redundantes. Ambos os códigos foram submetidos a 3

iterações na decodificação através do SPA. O código que apresentou maior porcentagem de bits redundantes obteve um melhor desempenho de BER para as SNRs menores que 5 dB, já para SNRs maiores o código de bloco maior no caso $C(4489, 4158)$ o desempenho se mostra mais satisfatório. Entretanto, vale ressaltar que a complexidade do algoritmo soma-produto aumenta proporcionalmente com o tamanho do código utilizado devido as operações necessárias para a atualização das probabilidades no passo horizontal do algoritmo.

Uma análise interessante se deu em torno do cálculo do ganho de codificação, onde ao relacionar as curvas entre a BER codificada e a BER do canal sem codificação, verificou-se que para um código $C(1369, 1260)$, com SNR variando entre 0 dB e 5.25 dB e 6 iterações no decodificador, na segunda iteração para uma BER de aproximadamente 10^{-4} o ganho de codificação foi em torno de 3 dB, ou seja, para atingir essa mesma BER um canal com codificação precisa de metade da potência de SNR quando comparado ao canal não codificado. Na sexta iteração e para uma BER de $2,563 \times 10^{-6}$ o CG foi de 4.91 dB, já o NCG que considera a taxa de código que está sendo utilizada foi de 4.5498 dB.

Apesar de todas as simulações terem sido realizadas utilizando BPSK como formato de modulação, é possível extrapolar esses resultados para sistemas de comunicações ópticas que utilizem QPSK, visto que o desempenho em termos de BER se mantém o mesmo, aumentando a eficiência espectral, ou ainda a taxa de transmissão de bits. Este comportamento ocorre pois o QPSK pode ser representado através de dois sinais BPSK ortogonais.

Este trabalho considerou que os efeitos não lineares inerentes aos sistemas de comunicações ópticas se manifestaram através de um ruído gaussiano aditivo, sendo possível a partir de algumas considerações adotar um canal AWGN para realizar as simulações, porém não considerou outros elementos comumente presentes em sistemas de comunicações ópticas como por exemplo amplificadores ou multiplexadores. Sendo assim, trabalhos futuros podem explorar a influência de tais elementos no desempenho da codificação.

REFERÊNCIAS

AMMAR, B. et al. Construction of low-density parity-check codes based on balanced incomplete block designs. **IEEE Transactions on Information Theory**, v. 50, n. 6, p. 1257–1269, June 2004. ISSN 1557-9654.

ARABACI, M. et al. Nonbinary quasi-cyclic ldpc-based coded modulation for beyond 100-gb/s transmission. **IEEE Photonics Technology Letters**, v. 22, n. 6, p. 434–436, 2010.

BLAUNSTEIN, N. et al. ***Fiber Optic and Atmospheric Optical Communication***. 5. ed. [S.l.]: IEEE Press and Willey, 2020.

CARRASCO, R. A.; JOHNSTON, M. ***Non-binary Error Control Coding For Wireless Communication And Data Storage***. 1. ed. [S.l.]: John Wiley Sons Ltd, 2008.

COVER, T. M.; THOMAS, J. A. **Elements of Information Theory**. [S.l.]: Wiley Interscience, 2006.

CUSHON, K.; LARSSON-EDEFORS, P.; ANDREKSON, P. Improved low-power ldpc fec for coherent optical systems. In: **2017 European Conference on Optical Communication (ECOC)**. [S.l.: s.n.], 2017. p. 1–3.

DECUSATIS, C. ***Handbook of Fiber Optic Data Communication***. 3. ed. [S.l.]: Elsevier, 2008.

DELANGE, O. E. *Wide-band optical communication systems: Part II—Frequency-division multiplexing*. ***Proceedings of the IEEE***, v. 58, p. 1683 – 1690, 1970.

ELEFTHERIOU, E.; ÖLÇER, S. **G.gen: LDPC codes for G.dmt.bis and G.lite.bis**. 2001. Disponível em: <http://web.stanford.edu/class/ee379b/class_reader/ibm3>.

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

GRIFFIN, R.; CARTER, A. *Optical differential quadrature phase-shift key (oDQPSK) for high capacity optical transmission*. ***Optical Fiber Communication Conference and Exhibit***, p. 367–368, 2002.

GROVER, W. *Forward error correction in dispersion-limited lightwave systems*. ***Journal of Lightwave Technology*** , v. 6, p. 643 – 654, 1988.

HAMMING, R. W. Error detecting and error correcting codes. **The Bell System Technical Journal**, v. 29, n. 2, p. 147–160, 1950.

HAYKIN, S. S. **Communication systems**. [S.l.]: Wiley Interscience, 2001.

KAHN, J. M.; HO, K.-P. *Spectral efficiency limits and modulation/detection techniques for DWDM systems*. ***IEEE Journal of Selected Topics in Quantum Electronics***, v. 10, p. 259 – 272, 2004.

KAMINOW, I. P.; LI, T.; WILLNER, A. E. ***Optical Fiber Telecommunications***. 5. ed. [S.l.]: Elsevier, 2008.

KAMIYA, N.; SHIOIRI, S. Concatenated qc-ldpc and spc codes for 100 gbps ultra long-haul optical transmission systems. In: **2010 Conference on Optical Fiber Communication (OFC/NFOEC), collocated National Fiber Optic Engineers Conference**. [S.l.: s.n.], 2010. p. 1–3.

KIKUCHI, K. *Phase-diversity homodyne detection of multilevel optical modulation with digital carrier phase estimation*. ***IEEE Journal of Selected Topics in Quantum Electronics***, v. 12, p. 563 – 570, 2006.

KIKUCHI, K. *Fundamentals of Coherent Optical Fiber Communications*. ***The Bell System Technical Journal***, v. 34, p. 157 – 179, 2016.

KUMAR, S.; DEEN, M. J. ***Fiber Optic Communications Fundamentals and Applications***. 1. ed. [S.l.]: Wiley, 2014.

LIN, S.; JR., D. J. C. ***Error Control Coding***. 2. ed. [S.l.]: Pearson, 2004.

LULU, A. Y. **Construction of Ldpc Codes Using Randomly Permutated Copies of Parity Check Matrix**. 81 p. Dissertação (Mestrado) — Islamic University of Gaza (IUG), Gaza, 2012.

MACKAY, D.; NEAL, R. Near shannon limit performance of low density parity check codes. **ELECTRONICS LETTERS**, v. 32, n. 18, p. 1645 – 1646, 1996.

MIZUOCHI, T.; MIYATA, Y. *LDPC-based advanced FEC for 100 Gbps transmission*. ***2008 Digest of the IEEE/LEOS Summer Topical Meetings***, 2008.

MOREIRA, J. R. O.; 1, L. B. L. Fundamentos de design combinatórios e aplicações em códigos. **Revista de Divulgação Científica em Ciências Exatas e Tecnológicas PORANDU**, v. 3, n. 1, p. 1 – 8, 2019.

MORI, Y. et al. *Unrepeated 200-km transmission of 40-Gbit/s 16-QAM signals using digital coherent optical receiver*. ***IEEE Journal of Selected Topics in Quantum Electronics***, v. 12, p. 563 – 570, 2006.

NYQUIST, H. Certain factors affecting telegraph speed. **Transactions of the American Institute of Electrical Engineers**, XLIII, p. 412–422, 1924.

NYQUIST, H. Certain topics in telegraph transmission theory. **Transactions of the American Institute of Electrical Engineers**, v. 47, n. 2, p. 617–644, 1928.

POGGIOLINI, P. et al. The gn-model of fiber non-linear propagation and its applications. **Journal of Lightwave Technology**, v. 32, n. 4, p. 694–721, 2014.

PUC, A. et al. Concatenated fec experiment over 5000 km long straight line wdm test bed. In: **OFC/IOOC . Technical Digest. Optical Fiber Communication Conference, 1999, and the International Conference on Integrated Optics and Optical Fiber Communication**. [S.l.: s.n.], 1999. v. 3, p. 255–258 vol.3.

RAMANUJAM, N. et al. Forward error correction (fec) techniques in long-haul optical transmission systems. In: **LEOS 2000. 2000 IEEE Annual Meeting Conference Proceedings. 13th Annual Meeting. IEEE Lasers and Electro-Optics Society 2000 Annual Meeting (Cat. No.00CH37080)**. [S.l.: s.n.], 2000. v. 2, p. 405–406 vol.2.

RIBEIRO, J. B. B. ***T*ELECOMUNICAÇÕES VOLUME 5**. [S.l.: s.n.].

RICHARDSON, T. J.; SHOKROLLAHI, M. A.; URBANKE, R. L. Design of capacity-approaching irregular low-density parity-check codes. **IEEE Transactions on Information Theory**, v. 47, n. 2, p. 619–637, 2001.

RYAN, W.; LIN, S. **Channel Codes: Classical and Modern**. [S.l.]: Cambridge University Press, 2009.

RYAN, W. E.; LIN, S. ***Channel Codes Classical and Modern***. 1. ed. [S.l.]: Cambridge University Press, 2009.

SAE-YOUNG CHUNG et al. On the design of low-density parity-check codes within 0.0045 db of the shannon limit. **IEEE Communications Letters**, v. 5, n. 2, p. 58–60, 2001.

SANTOS, L. F. D. **Decodificadores de baixa complexidade para códigos LDPC Q-ários**. 50 p. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, 2014.

SENIOR, J. M. ***Optical Fiber Communications Principles and Practice***. 3. ed. [S.l.]: Pearson, 2009.

SHANNON, C. E. *A Mathematical Theory of Communication*. **The Bell System Technical Journal**, v. 27, p. 397–423 *and* 623–656, 1948. *Reprinted in*: C.E.Shannon and W.Weaver, eds., *A Mathematical Theory of Communication*, (Univ. of Illinois Press, Urbana, Illinois, (1963)).

SMITH, B. P.; KSCHISCHANG, F. R. *Future Prospects for FEC in Fiber-Optic Communications* . ***IEEE Journal of Selected Topics in Quantum Electronics*** , v. 16, p. 1245 – 1257, 2010.

SONG, H. et al. Combined constrained code and ldpc code for long-haul fiber-optic communication systems. In: **2012 IEEE Global Communications Conference (GLOBECOM)**. [S.l.: s.n.], 2012. p. 2984–2989.

TYCHOPOULOS, A.; KOUFOPAVLOU, O.; TOMKOS, I. *FEC in optical communications - A tutorial overview on the evolution of architectures and the future prospects of outband and inband FEC for optical communications* . ***IEEE Circuits and Devices Magazine*** , v. 22, p. 79 – 86, 2006.

VASIC, B.; DJORDJEVIC, I. B. Low-density parity check codes for long-haul optical communication systems. **IEEE Photonics Technology Letters**, v. 14, n. 8, p. 1208–1210, 2002.

WERTS, A. *Propagation de la lumière cohérente dans les fibres optiques*. ***L'Onde Electrique***, v. 46, p. 967–980, 1966.