

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

EDUARDO TOSHIO MURAKAMI

Estudo e Análise do Desempenho de Códigos QC-LDPC em Sistemas OFDM sob Canal AWGN

São João da Boa Vista
2025

Eduardo Toshio Murakami

Estudo e Análise do Desempenho de Códigos QC-LDPC em Sistemas OFDM sob Canal AWGN

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 Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Eletrônica e de Telecomunicações .

Orientadora: Prof^a Dr^a. Cintya Wink de Oliveira
Benedito

Coorientadora: Prof^a M^a. Renata Mataveli Viti

São João da Boa Vista

2025

M972e

Murakami, Eduardo Toshio

Estudo e análise do desempenho de códigos QC-LDPC em sistemas OFDM sob canal
AWGN / Eduardo Toshio Murakami. -- São João da Boa Vista, 2025
85 p. : il.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Telecomunicações) -
Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, São João da Boa
Vista

Orientadora: Cintya Wink de Oliveira Benedito

Coorientadora: Renata Mataveli Viti

1. Códigos corretores de erros (Teoria da informação). 2. Multiplexação. 3. Funções
ortogonais. 4. Modulação digital. I. Título.

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

TRABALHO DE CONCLUSÃO DE CURSO

**ESTUDO E ANÁLISE DO DESEMPENHO DE CÓDIGOS QC-LDPC EM SISTEMAS
OFDM SOB CANAL AWGN**

Aluno: Eduardo Toshio Murakami

Orientador: Prof.^a Dr.^a Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientadora)
- Guilherme Simon da Rosa (Examinador)
- Ivan Aritz Aldaya Garde (Examinador)

Os formulários de avaliação e a ata da defesa, na qual consta a aprovação do trabalho, devidamente assinados pela banca encontram-se no prontuário eletrônico do aluno.

São João da Boa Vista, 04 de dezembro de 2025

Este trabalho é dedicado aos meus pais e amigos, que sempre me apoiaram diante dos desafios que a vida me impôs.

AGRADECIMENTOS

Aos meus pais, João Murakami e Sônia Yumi Aizawa Murakami, pelo apoio incondicional em todos os momentos e por estarem sempre ao meu lado diante dos desafios que a vida me impôs.

À minha orientadora, Prof^a. Dr^a. Cintya Wink de Oliveira Benedito, pela paciência, dedicação e contribuição essencial ao desenvolvimento deste trabalho.

À minha coorientadora, Prof^a. Dr^a. Renata Mataveli Viti, que me deu todo o suporte necessário para a realização deste trabalho.

Aos professores Ivan Aritz Aldaya Garde e Guilherme Simon da Rosa, por terem aceitado participar da banca examinadora deste trabalho.

“Se enxerguei mais longe, foi por estar sobre os ombros de gigantes.”
(Isaac Newton)

RESUMO

Nos últimos anos, a demanda por sistemas de comunicações de alta taxa de transmissão, alta confiabilidade e alta eficiência espectral tem aumentado significativamente. Nesse contexto, a técnica de multiplexação por divisão ortogonal de frequências (*OFDM, Orthogonal Frequency Division Multiplexing*) destaca-se devido à sua elevada robustez a canais seletivos em frequência, elevada eficiência espectral e elevada robustez ao efeito multipercurso, podendo ser empregada com diferentes esquemas de modulação, como a modulação por deslocamento de fase (*PSK, Phase Shift Keying*) e a modulação de amplitude em quadratura (*QAM, Quadrature Amplitude Modulation*), permitindo ajustar o compromisso entre taxa de transmissão e tolerância ao ruído. Ainda assim, o sistema permanece sujeito a erros devido ao ruído e interferências, tornando necessária a utilização de códigos corretores de erros. Os códigos de verificação de paridade de baixa densidade (*LDPC, Low-Density Parity-Check*), especialmente na forma quase-cíclica (*QC-LDPC, Quasi-Cyclic Low-Density Parity-Check*), apresentam excelente desempenho e implementação eficiente. Este trabalho tem como objetivo principal o estudo e a avaliação do desempenho de sistemas OFDM, com diversos esquemas de modulação, utilizando códigos QC-LDPC, operando em um canal com ruído aditivo branco Gaussiano (*AWGN, Additive White Gaussian Noise*) e aplicando diversos algoritmos de decodificação. Para uma análise comparativa de desempenho foram realizadas simulações computacionais utilizando a plataforma MATLAB. Os resultados obtidos por meio das simulações serão apresentados de forma a demonstrar como a utilização de diversos esquemas de modulação afeta o desempenho em função da taxa de erro de bit (*BER, Bit Error Rate*) de um sistema OFDM com codificação QC-LDPC. Além disso, os resultados das simulações também serão apresentados de forma a comparar as possíveis mudanças que a utilização de uma codificação QC-LDPC pode ocasionar em um sistema OFDM com canal AWGN. Também será feita uma análise comparativa entre o desempenho em função da BER de diversos algoritmos de decodificação e de como o número de iterações de um algoritmo de decodificação pode influenciar no desempenho.

PALAVRAS-CHAVE: ofdm; modulação; psk; qam; códigos ldpc; códigos qc-ldpc; canal awgn; ber; algoritmos de decodificação.

ABSTRACT

In recent years, the demand for communication systems with high data rates, high reliability, and high spectral efficiency has increased significantly. In this context, Orthogonal Frequency Division Multiplexing (OFDM) stands out due to its high robustness to frequency-selective channels, high spectral efficiency, and strong robustness to multipath effect, and it can be employed with different modulation schemes, such as Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM), allowing the adjustment of the trade-off between transmission rate and noise tolerance. Even so, the system remains subject to errors due to noise and interference, making the application of error-correction techniques necessary. Low-Density Parity-Check (LDPC) codes, especially in their quasi-cyclic form (QC-LDPC), present excellent performance and efficient implementation. The main objective of this work is the study and evaluation of the performance of OFDM systems, with various modulation schemes, using QC-LDPC codes, operating in an Additive White Gaussian Noise (AWGN) channel and applying different decoding algorithms. For a comparative performance analysis, computational simulations were carried out using MATLAB platform. The results obtained through the simulations will be presented in order to demonstrate how the use of different modulation schemes affects the performance in terms of Bit Error Rate (BER) of an OFDM system with QC-LDPC coding. In addition, the simulation results will also be presented in order to compare the possible changes that the use of QC-LDPC coding may cause in an OFDM system with an AWGN channel. A comparative analysis will also be carried out between the performance in terms of BER of several decoding algorithms and how the number of iterations of a decoding algorithm can influence the performance.

KEYWORDS: ofdm; modulation; psk; qam; ldpc codes; qc-ldpc codes; awgn channel; ber; decoding algorithms.

LISTA DE ILUSTRAÇÕES

Figura 1 – Constelação BPSK.	24
Figura 2 – Constelação QPSK.	24
Figura 3 – Sistema QAM.	25
Figura 4 – Modulador QAM utilizando a forma complexa.	26
Figura 5 – Constelação 16-QAM.	26
Figura 6 – Mapeamento das amplitudes das componentes em fase e em quadratura.	27
Figura 7 – Constelação 64-QAM.	27
Figura 8 – PSD de um sinal modulado.	28
Figura 9 – PSD de um sinal FDM.	29
Figura 10 – Esquema de um sistema FDM.	30
Figura 11 – Período de símbolo de um sistema monoportadora e de um sistema multiportadora.	31
Figura 12 – Espectro de um sinal FDM.	32
Figura 13 – Espectro de um sinal OFDM.	32
Figura 14 – Transmissor OFDM.	33
Figura 15 – Transmissor OFDM utilizando a IDFT.	34
Figura 16 – Receptor OFDM.	35
Figura 17 – Espectro das diferentes portadoras.	35
Figura 18 – Canal Gaussiano.	38
Figura 19 – Estrutura de um bloco.	39
Figura 20 – Grafo bipartido.	44
Figura 21 – Grafo de Tanner de um código LDPC $(n, k) = (12, 6)$	46
Figura 22 – Árvore do grafo de Tanner de um código LDPC $(n, k) = (12, 6)$	46
Figura 23 – Protógrafo.	48
Figura 24 – Cópias do Protógrafo.	49
Figura 25 – Grafo baseado em um protógrafo.	49
Figura 26 – Protógrafo com arestas paralelas.	50
Figura 27 – Grafo obtido a partir de um protógrafo.	50
Figura 28 – Grafo de um código LDPC $(8, 4)$	59
Figura 29 – Gráfico da função $f(x) = \left \ln \left(\tanh \frac{ x }{2} \right) \right $	62
Figura 30 – Diagrama de blocos do modelo simulado.	71
Figura 31 – Curvas da BER para um sistema OFDM utilizando uma modulação BPSK sem codificação e com codificação QC-LDPC.	73
Figura 32 – Curvas da BER para um sistema OFDM utilizando uma modulação 16-QAM sem codificação e com codificação QC-LDPC.	74
Figura 33 – Curvas da BER para um sistema OFDM com codificação QC-LDPC utilizando a matriz $\mathbf{P}_1$ e diferentes modulações.	75

Figura 34 – Curvas da BER para um sistema OFDM com codificação QC-LDPC utilizando a matriz P_2 e diferentes modulações.	75
Figura 35 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_1 , comparando diferentes algoritmos de codificação com 5 iterações.	76
Figura 36 – Curvas da BER para um sistema OFDM com modulação 16-QAM e com codificação QC-LDPC utilizando a matriz P_1 comparando diferentes algoritmos de codificação com 5 iterações.	76
Figura 37 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_2 comparando diferentes algoritmos de codificação com 5 iterações.	77
Figura 38 – Curvas da BER para um sistema OFDM com modulação 16-QAM e com codificação QC-LDPC utilizando a matriz P_2 comparando diferentes algoritmos de codificação com 5 iterações.	78
Figura 39 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_2 comparando diferentes algoritmos de codificação com 25 iterações.	78
Figura 40 – Curvas da BER para um sistema OFDM com modulação 16-QAM e com codificação QC-LDPC utilizando a matriz P_2 comparando diferentes algoritmos de codificação com 25 iterações.	79
Figura 41 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_1 e algoritmo de decodificação SPA comparando diferentes números de iterações.	79
Figura 42 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_2 e algoritmo de decodificação SPA comparando diferentes números de iterações.	80

LISTA DE ABREVIATURAS E SIGLAS

5G	<i>Fifth Generation of Mobile Networks</i>
ADSL	<i>Asymmetric Digital Subscriber Line</i>
AWGN	<i>Additive White Gaussian Noise</i>
BER	<i>Bit Error Rate</i>
BPA	<i>Belief Propagation Algorithm</i>
BPSK	<i>Binary Phase Shift Keying</i>
DAC	<i>Digital-to-Analog Converter</i>
DFT	<i>Discrete Fourier Transform</i>
DVB-C2	<i>Digital Video Broadcasting – Cable, 2nd Generation</i>
DVB-S2	<i>Digital Video Broadcasting – Satellite, 2nd Generation</i>
FDM	<i>Frequency Division Multiplexing</i>
FFT	<i>Fast Fourier Transform</i>
IDFT	<i>Inverse Discrete Fourier Transform</i>
IFFT	<i>Inverse Fast Fourier Transform</i>
LDPC	<i>Low-Density Parity-Check</i>
LLR	<i>Log-Likelihood Ratio</i>
MS	<i>Min-Sum</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
PAPR	<i>Peak-to-Average Power Ratio</i>
PSK	<i>Phase Shift Keying</i>
PSD	<i>Power Spectral Density</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QC	<i>Quasi-Cyclic</i>
QC-LDPC	<i>Quasi-Cyclic Low-Density Parity-Check</i>
QPSK	<i>Quadrature Phase Shift Keying</i>

SPA	<i>Sum-Product Algorithm</i>
SPC	<i>Single Parity Check</i>
Wi-Fi	<i>Wireless Fidelity</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>

LISTA DE SÍMBOLOS

M	Ordem da modulação
$\phi_{PSK}(t)$	Sinal PSK
T_b	Período de duração de um bit
ω_c	Frequência angular da portadora
A	Amplitude do sinal PSK
θ_m	Fase do m -ésimo símbolo da constelação do sinal PSK
$p(t)$	Pulso base
$\phi_{QAM}(t)$	Sinal QAM
a_i	Amplitude da componente em fase de um sinal QAM
b_i	Amplitude da componente em quadratura de um sinal QAM
$\xi_{QAM}(t)$	Sinal QAM complexo em banda base
$i_{QAM}(t)$	Componente em fase em banda base do sinal QAM
$q_{QAM}(t)$	Componente em quadratura em banda base do sinal QAM
$\Re\{\cdot\}$	Parte real de um número complexo
f_c	Frequência da portadora
T_s	Período de um símbolo
$\mathcal{R}_g(\tau)$	Função de autocorrelação
$g(t)$	Sinal no domínio do tempo
$\mathcal{S}_g(f)$	Densidade espectral de potência
$FT\{\cdot\}$	Transformada de Fourier
$\text{triang}(\cdot)$	Função triangular
$\text{sinc}(\cdot)$	Função seno cardinal
B_s	Banda ocupada pelo sinal modulado
α_{ro}	Fator de roll-off
τ_m	Espalhamento por atraso

R_b	Taxa de bits
N_{mp}	Número de subportadoras
Δt_s	Intervalo de símbolo
$x_{OFDM}(t)$	Sinal OFDM na saída do transmissor
X_k	Sinal transmitido modulado com a k -ésima portadora
f_k	Frequência da k -ésima portadora
Δf_s	Espaçamento entre portadoras
$x_{OFDM_s}(t)$	Sinal OFDM amostrado
n	Número da amostra
t_n	Instante de tempo da n -ésima amostra
Δt_s	Diferença de tempo entre duas amostras adjacentes
X	Variável aleatória
$\mathcal{X}$	Alfabeto
p_k	Probabilidade
$p(x)$	Função massa de probabilidade
$I(x_k)$	Quantidade de Informação
$H(X)$	Entropia
$\mathbb{E}[\cdot]$	Esperança
Y	Variável aleatória
$\mathcal{Y}$	Alfabeto
$p(x, y)$	Distribuição de probabilidade conjunta
$p(x y)$	Distribuição de probabilidade condicional
$H(X, Y)$	Entropia conjunta
$H(Y X)$	Entropia condicional
$I(X; Y)$	Informação mútua
$h(X)$	Entropia diferencial
$f(x)$	Função densidade de probabilidade

S	Conjunto de suporte da variável aleatória
$i(X; Y)$	Informação mútua diferencial
$f(x, y)$	Função densidade de probabilidade conjunta
C	Capacidade do canal
$\max$	Maior elemento
X_i	Variável aleatória de entrada do canal Gaussiano
Y_i	Variável aleatória de saída do canal Gaussiano
Z_i	Ruído do canal Gaussiano
N_{n_0}	Variância do ruído AWGN
P	Restrição de potência
B_c	Banda do canal
k	Número de bits de informação
n	Número de bits da palavra-código
R	Taxa do código
C_{LBC}	Código de Bloco Linear
$GF(q)$	Corpo finito com q elementos
$d(\mathbf{c}_1, \mathbf{c}_2)$	Distância de Hamming
d_{min}	Distância mínima de Hamming
$\min$	Menor elemento
$w(\mathbf{c})$	Peso de Hamming
w_{min}	Peso mínimo de Hamming
$\mathbf{G}$	Matriz Geradora
S_c	Subespaço vetorial
$\mathbf{I}_k$	Matriz identidade de dimensões $k \times k$
$\mathbf{P}_s$	Matriz de paridade
$\mathbf{H}$	Matriz Controle de Paridade
$(\cdot)^T$	Matriz transposta

$D(V, A)$	Grafo
V	Conjunto de nós
A	Conjunto de arestas
m	Número de equações de paridade
f_i	Nó de função
c_j	Nó de variável
u_k	Bit da mensagem
x_{n-k}	Bit de paridade
u	Vetor de bits da mensagem
v	Vetor de bits da palavra-código
w_c	Número de 1's em uma coluna
w_r	Número de 1's em uma linha
d	Densidade de um código LDPC
$\lambda(x)$	Distribuição de grau dos nós de variável
$\rho(x)$	Distribuição de grau dos nós de função
λ_i	Fração de arestas no grafo que está conectada a um nó de variável de grau i
ρ_i	Fração de arestas no grafo que está conectada a um nó de paridade de grau i
Z_c	Fator de levantamento
H_P	Matriz protógrafo
π_{ij}	Matrizes de permutações
P	Matriz protótipo
$P(i, j)$	Elemento da matriz protótipo
r	Vetor de valores recebidos no decodificador
L	Conjunto de LLRs
l_i^i	LLR intrínseca
l_i^e	LLR extrínseca

$P_r(c_i = 0 r_i)$	Probabilidade de um bit enviado ser 0 dado o valor do bit recebido no decodificador
$f(r_i c_i = 0)$	Função densidade de probabilidade condicional de r_i dado que $c_i = 0$
$P_r(c_i = 0)$	Probabilidade de $c_i = 0$
$f(r_i)$	Função densidade de probabilidade de r_i
$P_r(c_i = 1 r_i)$	Probabilidade de um bit enviado ser 1 dado o valor do bit recebido no decodificador
$f(r_i c_i = 1)$	Função densidade de probabilidade condicional de r_i dado que $c_i = 1$
$P_r(c_i = 1)$	Probabilidade de $c_i = 1$
ι_i^i	Razão de verossimilhança intrínseca
l_i^i	Razão logarítmica de verossimilhança intrínseca
$N(0, \sigma^2)$	Ruído adicionado pelo canal AWGN
σ^2	Variância do ruído
m	Vetor de bits da mensagem a ser codificada
c	Vetor da palavra-código
ρ	Bit de verificação de paridade
p_i	Probabilidade de um bit enviado ser 0 dado o valor do bit recebido no decodificador
tanh	Tangente hiperbólica
sign	Parte de sinal
c_n	Nó de variável
f_m	Nó de função
$C(f_m)$	Conjunto de nós de variável conectados a um nó de função
$F(c_n)$	Conjunto de nós de função conectados a um nó de variável
$C(f_m) \setminus c_n$	Conjunto de nós de variável conectados a um nó de função f_m , com exceção do nó de variável c_n
$F(c_n) \setminus f_m$	Conjunto de nós de função conectados a um nó de variável c_n , com exceção do nó de função f_m
L_n	n -ésimo vetor de LLR

$c_n \rightarrow f_m$	Informações enviadas dos nós de variável para os nós de função
$r_{f_m \rightarrow c_n}$	Informações enviadas dos nós de função para os nós de variável
$\mathbf{H}^j$	Matriz da j -ésima camada de $\mathbf{H}$
L_{QAM}	LLRs do passo de inicialização em modulações de ordem maior que 2
C_0	Conjunto de todos os símbolos na constelação onde o k -ésimo bit é $b_k = 0$
C_1	Conjunto de todos os símbolos na constelação onde o k -ésimo bit é $b_k = 1$
s	Símbolo em uma constelação
E_b	Energia média utilizada para a transmissão de um bit
N_0	Densidade espectral de potência do ruído AWGN

SUMÁRIO

1	INTRODUÇÃO	19
2	PRINCÍPIOS DE COMUNICAÇÕES DIGITAIS E OFDM	22
	2.1 – Princípios Básicos de Sistemas de Comunicações Digitais	23
	2.1.1 – Modulações Digitais de Portadora Única	23
	2.1.2 – Densidade Espectral de Potência e Transmissão Multiportadora	28
	2.2 – OFDM	31
3	TEORIA DA INFORMAÇÃO E CODIFICAÇÃO	36
	3.1 – Medidas de Informação	36
	3.2 – Códigos de Bloco Linear	39
4	CÓDIGOS LDPC	42
	4.1 – Definições e Conceitos Básicos em Códigos LDPC	42
	4.2 – Códigos QC-LDPC Baseados em Protógrafos	48
	4.3 – Decodificação em Códigos LDPC	53
	4.3.1 – Razão Logarítmica de Verossimilhança (LLR)	53
	4.3.2 – Algoritmo Soma e Produto (SPA)	57
	4.3.3 – Algoritmo Mínima Soma (MS)	61
	4.3.4 – Decodificação em Camadas	63
	4.3.5 – LLRs em Modulações de Ordem Maior que 2	70
5	RESULTADOS E DISCUSSÕES	71
6	CONCLUSÃO	81
	REFERÊNCIAS	83

1 INTRODUÇÃO

Nas últimas décadas, o volume de dados transmitidos por redes de comunicação e a demanda por sistemas de comunicação de maior confiabilidade cresceram de forma significativa devido a diversos fatores, como a popularização da internet, a expansão do uso de dispositivos móveis e o aumento de serviços que exigem alta capacidade de tráfego, como plataformas de *streaming*, redes sociais e aplicações em tempo real (TANENBAUM; WETHERALL, 2017). A eficiência espectral também é um aspecto importante, uma vez que o espectro de frequências disponível é um recurso limitado e altamente disputado. Com o aumento do número de dispositivos conectados e de serviços que dependem de comunicação em alta velocidade, torna-se necessário transmitir mais informação utilizando a mesma faixa de frequência, o que exige o desenvolvimento e a adoção de sistemas de comunicação com maior eficiência espectral. Assim, é fundamental utilizar e aperfeiçoar técnicas que proporcionem maior eficiência espectral, elevadas taxas de transmissão de dados e maior robustez diante das limitações físicas do canal de comunicação (LATHI; DING, 2012).

Nesse contexto, as técnicas de modulação digital desempenham um papel fundamental, pois são responsáveis por mapear a informação binária em símbolos adequados ao sistema de comunicação, possibilitando, assim, atingir características desejadas na transmissão de dados. Modulações como a modulação por deslocamento de fase binária (BPSK, *Binary Phase Shift Keying*) e a modulação por deslocamento de fase em quadratura (QPSK, *Quadrature Phase Shift Keying*) apresentam boa resistência ao ruído, sendo utilizadas em situações em que a relação sinal-ruído é reduzida. Por outro lado, modulações em amplitude e fase, como a modulação de amplitude em quadratura (QAM, *Quadrature Amplitude Modulation*), permitem transmitir um número maior de bits por símbolo, aumentando a eficiência espectral, embora exijam condições de canal mais favoráveis (SKLAR, 2001).

Outra estratégia de modulação é a multiplexação por divisão de frequência ortogonal (OFDM, *Orthogonal Frequency Division Multiplexing*), proposta inicialmente por Chang em (CHANG, 1966). O OFDM consiste na divisão da banda total disponível em diversas subportadoras ortogonais, cada uma transmitindo uma parte da informação total. Essa abordagem reduz a taxa de transmissão por subportadora, aumentando a resistência ao desvanecimento e à seletividade em frequência do canal. Entretanto, a implementação prática do OFDM só se tornou viável após a demonstração de Weinstein e Ebert em (WEINSTEIN; EBERT, 1971), de que a modulação e demodulação *multicarrier* poderiam ser realizadas utilizando a Transformada Discreta de Fourier (DFT, *Discrete Fourier Transform*) e sua inversa (IDFT, *Inverse Discrete Fourier Transform*). Em 1965, James W. Cooley e John W. Tukey introduziram, em (COOLEY; TUKEY, 1965), o algoritmo da Transformada Rápida de Fourier (FFT, *Fast Fourier Transform*), que reduziu drasticamente o número de operações necessárias para a realização da DFT. Com a introdução dos algoritmos para o cálculo da FFT e sua inversa (IFFT, *Inverse Fast Fourier Transform*), o OFDM passou a ser computacionalmente mais eficiente, viabilizando sua aplicação em cenários de comunicação modernos.

Mesmo com o uso da técnica OFDM e da seleção adequada da modulação, os sistemas de comunicação permanecem sujeitos a perturbações, como ruído, interferência e desvanecimento, que

podem provocar erros na recepção dos símbolos transmitidos. De acordo com o Teorema da Capacidade de Canal de Shannon (SHANNON, 1948), é possível transmitir informação com probabilidade de erro arbitrariamente pequena, desde que se utilize um esquema de codificação adequado e que a taxa da fonte seja inferior ou igual à capacidade do canal. Esse princípio fundamenta o uso de técnicas de codificação de canal, que adicionam redundância estruturada aos dados transmitidos, permitindo a detecção e correção de erros na recepção.

Nesse cenário, os códigos de verificação de paridade de baixa densidade (LDPC, *Low-Density Parity-Check*), introduzidos por Gallager em (GALLAGER, 1963), destacam-se por oferecer desempenho próximo ao limite teórico estabelecido por Shannon. Após um longo período de pouca exploração, o estudo de códigos LDPC foi retomado em meados da década de 1990, quando MacKay e Neal demonstraram, em (MACKAY; NEAL, 1996), seu alto desempenho. Uma classe particularmente relevante é a dos códigos LDPC quase-cíclicos (QC-LDPC, *Quasi-Cyclic Low-Density Parity-Check*), caracterizados por matrizes de paridade compostas por blocos estruturados obtidos por rotações cíclicas de submatrizes. Essa organização facilita a análise, o desenvolvimento e a implementação prática desses códigos (RYAN; LIN, 2009). Padrões modernos, como o IEEE 802.11 (*Wireless Fidelity*, Wi-Fi) (IEEE Standards Association, 2020) e o 5G (*Fifth Generation*) (3rd Generation Partnership Project (3GPP), 2024), utilizam códigos QC-LDPC devido à sua eficiência.

Os códigos LDPC podem utilizar diferentes algoritmos de decodificação, cada um com suas devidas propriedades. Um dos algoritmos mais conhecidos é o algoritmo de Soma e Produto (SPA, *Sum-Product Algorithm*) que utiliza operações baseadas em probabilidades para a realização da decodificação e permite alcançar um desempenho próximo ao limite de Shannon, embora com custo computacional relativamente elevado. Com o objetivo de reduzir esse custo computacional, o algoritmo mínima soma (MS, *Min-Sum*) substitui algumas operações do SPA por aproximações, tornando o processamento mais simples, porém com uma leve degradação no desempenho de correção. Além disso, ambos os algoritmos podem ser calculados em camadas o que permite uma convergência do processo iterativo mais rápida.

Neste trabalho serão analisados os códigos LDPC, com foco especial nos códigos LDPC quase-cíclicos (QC-LDPC), com o objetivo de investigar seu desempenho em sistemas OFDM submetidos a um canal AWGN. Será investigado o desempenho desses códigos quando aplicados a um sistema OFDM com diferentes esquemas de modulação, a fim de avaliar a influência das diferentes modulações no desempenho de um sistema OFDM codificado. Também serão comparados os desempenhos de diferentes algoritmos de decodificação, incluindo Soma-Produto (SPA), Min-Sum (MS) e suas abordagens em camadas, com o objetivo de identificar as vantagens e limitações de cada algoritmo. Além disso, será analisada a influência do número de iterações do decodificador sobre o desempenho do sistema. Todas as análises serão realizadas por meio de simulações computacionais desenvolvidas na plataforma MATLAB, permitindo uma avaliação detalhada do desempenho dos códigos QC-LDPC em diferentes condições de modulação e ruído.

O presente trabalho está organizado da seguinte forma: no Capítulo 2 será apresentada a técnica OFDM, discutindo seus princípios básicos, estrutura de transmissão e recepção, além das vantagens relacionadas à utilização dessa técnica. No Capítulo 3 serão revisados os conceitos essenciais para a

realização deste trabalho com relação à Teoria da Informação e aos códigos de bloco lineares, incluindo entropia, capacidade de canal e as bases para codificação e correção de erros. O Capítulo 4 será dedicado ao estudo dos códigos LDPC, apresentando suas características estruturais, representações em grafos, métodos de construção, propriedades da codificação e propriedades da decodificação, como a utilização de diversos algoritmos de decodificação, além de apresentar os códigos QC-LDPC e suas aplicações em sistemas modernos. No Capítulo 5 será apresentado o modelo a ser utilizado nas simulações e os resultados obtidos, juntamente com as análises de desempenho dos códigos QC-LDPC em diferentes cenários e com diferentes parâmetros. Por fim, no Capítulo 6, serão apresentadas as conclusões gerais do trabalho e possíveis direções para trabalhos futuros.

2 PRINCÍPIOS DE COMUNICAÇÕES DIGITAIS E OFDM

O desenvolvimento de um bom projeto de sistema de comunicação deve levar em conta diversos fatores para que a transmissão ocorra de forma confiável e atenda a requisitos de qualidade pré-estabelecidos. Um dos principais fatores é a largura de banda, a qual, em geral, é limitada. Outro fator importante é a taxa de erro de bit (BER), que deve ser a menor possível. Mais um fator relevante é a potência do sinal a ser transmitido, a qual não deve ser muito alta, visando a uma melhor eficiência energética. Na maioria dos casos, o sinal a ser transmitido pode sofrer espalhamento no domínio do tempo, e o canal, no domínio da frequência, em geral, não é plano em uma banda muito larga, o que pode causar distorções no sinal. Em sistemas de alta taxa de transmissão de dados, o espalhamento pode causar interferência intersimbólica entre os símbolos transmitidos, e a alta seletividade em frequência do canal em bandas muito largas pode inviabilizar a transmissão de dados utilizando apenas uma única portadora.

Uma possível solução para os problemas citados é a utilização de sistemas com abordagem multiportadora. Uma técnica com essa abordagem, amplamente empregada, é a multiplexação por divisão de frequência ortogonal (OFDM), na qual um fluxo de dados de alta taxa é dividido em subfluxos de menores taxas, cada um modulado por uma portadora diferente. Vale destacar que a técnica utiliza portadoras ortogonais entre si, com a separação frequencial mínima que garante a ortogonalidade entre as portadoras, permitindo que os espectros associados a cada portadora se sobreponham. Essa característica reduz drasticamente a banda ocupada pelo sinal gerado pelo sistema OFDM.

A técnica OFDM é utilizada em comunicações com fio, como o *Asymmetric Digital Subscriber Line* (ADSL) e o padrão *Digital Video Broadcasting – Cable, 2nd Generation* (DVB-C2), devido às diferentes atenuações sofridas pelos sinais ao longo do cabo de cobre ou coaxial. Em comunicações sem fio, porém, o OFDM encontra suas aplicações mais relevantes, pois apresenta elevada robustez ao desvanecimento por multipercurso. Essa técnica integra padrões amplamente difundidos, como o Wi-Fi (*Wireless Fidelity*), o WiMAX (*Worldwide Interoperability for Microwave Access*) e diversas configurações do 5G (*Fifth Generation of Mobile Networks*), especialmente em cenários sem linha de visada direta e que exigem altas taxas de transmissão.

Neste capítulo, serão apresentados os conceitos fundamentais de sistemas de comunicações digitais que servirão de base para o desenvolvimento deste trabalho, bem como a descrição da técnica OFDM. Inicialmente, na Seção 2.1, serão discutidos princípios essenciais para a compreensão da técnica OFDM. Em seguida, na Seção 2.2, será apresentada a técnica OFDM propriamente dita. Os conteúdos abordados neste capítulo podem ser encontrados em (LATHI; DING, 2012; PROAKIS, 2001; SKLAR, 2001; PRASAD, 2004; SCHULZE; LÜDERS, 2005; GONÇALVES, 2010; VITI, 2010; CLAIR, 2012).

2.1 PRINCÍPIOS BÁSICOS DE SISTEMAS DE COMUNICAÇÕES DIGITAIS

A técnica OFDM é um esquema de transmissão multiportadora, no qual um fluxo de dados em série de alta taxa é dividido em vários subfluxos em paralelo de menor taxa, sendo cada um deles transmitido por meio de uma subportadora distinta. Além disso, o OFDM pode ser interpretado como um caso particular da multiplexação por divisão em frequência (FDM, *Frequency Division Multiplexing*). Também é importante destacar que a técnica OFDM pode utilizar diferentes esquemas de modulação, cuja escolha depende das demandas de desempenho do sistema em questão e de sua complexidade.

Nesta seção serão apresentados conceitos fundamentais de sistemas de comunicações digitais que servirão de base para o entendimento da técnica OFDM. Na Subseção 2.1.1, serão discutidos esquemas de modulação digital de portadora única utilizados ao longo deste trabalho, incluindo as modulações BPSK, QPSK, 16-QAM e 64-QAM. Em seguida, na Subseção 2.1.2, serão apresentados conceitos relacionados à densidade espectral de potência, à técnica FDM e à transmissão multiportadora.

2.1.1 Modulações Digitais de Portadora Única

Diferentes esquemas de modulação podem ser utilizados a fim de se obter determinadas características desejadas na transmissão de um sinal. O aumento na ordem da constelação empregada para o mapeamento dos símbolos pode aumentar significativamente a taxa de transmissão de dados, elevando, assim, a eficiência espectral do sistema. Contudo, a utilização de constelações mais densas pode acarretar no aumento da BER do sistema, uma vez que os pontos da constelação ficam mais próximos entre si. Outras constelações com diferentes propriedades podem ser utilizadas para minimizar problemas específicos, como, por exemplo, a aplicação de constelações Asterisk-PSK, que podem reduzir a razão de pico para média de potência (PAPR, *Peak-to-Average Power Ratio*) em sistemas OFDM.

Um tipo de modulação muito utilizado é a modulação PSK, pois apresenta elevada robustez frente ao ruído de amplitude, uma vez que a informação é transmitida apenas na fase do sinal, tornando-se menos sensível a variações de amplitude. Além disso, essa modulação apresenta uma implementação relativamente simples. A equação de um sinal PSK de ordem M pode ser dada conforme a equação

$$\phi_{PSK}(t) = a_m \sqrt{\frac{2}{T_b}} \cos(\omega_c t) + b_m \sqrt{\frac{2}{T_b}} \sin(\omega_c t), \quad (2.1.1)$$

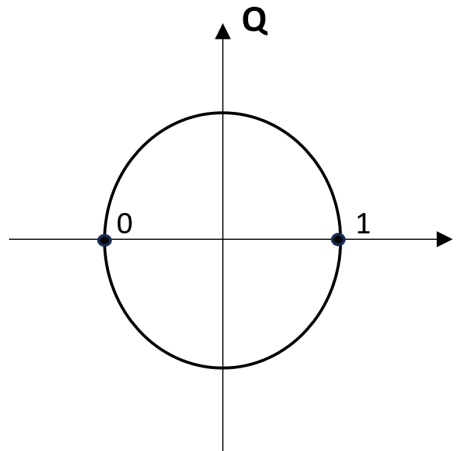
em que $a_m = A \cos(\theta_m)$ e $b_m = -A \sin(\theta_m)$, sendo A uma constante. Nessas equações, o valor de θ_m corresponde à fase do m -ésimo símbolo da constelação do sinal, T_b corresponde ao período de duração de um bit, e ω_c corresponde à frequência angular da portadora. Na Equação 2.1.1, também é possível observar duas componentes: a componente em fase, modulada por um cosseno, e a componente em quadratura, modulada por um seno.

Uma modulação muito utilizada devido à sua simplicidade é a modulação BPSK. Nessa modulação, os bits são transmitidos conforme a equação

$$\begin{cases} 0 \rightarrow p(t) \cos(\omega_c t) \\ 1 \rightarrow -p(t) \cos(\omega_c t), \end{cases} \quad (2.1.2)$$

em que $p(t)$ é um pulso base. A constelação de uma modulação BPSK pode ser observada na Figura 1.

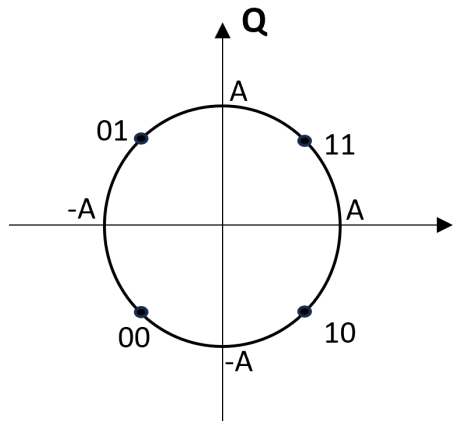
Figura 1 – Constelação BPSK.



fonte: Produção do próprio autor.

Outra modulação PSK amplamente utilizada é a modulação PSK de ordem 4 ou 4-PSK, pois essa permite transmitir uma taxa até duas vezes maior quando comparada à modulação BPSK. Além disso, essa modulação apresenta a mesma probabilidade de erro de bit em um canal AWGN que a modulação BPSK. A constelação de uma modulação 4-PSK pode ser observada na Figura 2.

Figura 2 – Constelação QPSK.



fonte: Produção do próprio autor.

Um sinal 4-PSK pode ser interpretado como a soma de dois sinais BPSK binários de mesma frequência, um com portadora $\cos(\omega_c t)$ e outro com portadora $\sin(\omega_c t)$. O sinal BPSK modulado pelo cosseno constitui a componente em fase, enquanto o sinal BPSK modulado pelo seno constitui a componente em quadratura. Por esse motivo, o sinal 4-PSK também é chamado de QPSK (*Quadrature Phase Shift Keying*).

Na modulação QAM de ordem M , é possível alcançar uma taxa de transmissão ainda mais alta, pois um único símbolo pode carregar a informação de $\log_2 M$ bits. Porém, quanto maior a ordem da modulação, maior a densidade da constelação, tornando o sistema mais suscetível a erros causados por

perturbações no canal. A equação geral de um sinal QAM de ordem M é dada por

$$\phi_{QAM}(t) = a_i p(t) \cos \omega_c t + b_i p(t) \sin \omega_c t, \quad (2.1.3)$$

em que $p(t)$ representa um pulso base e os coeficientes a_i e b_i correspondem às amplitudes das componentes em fase e em quadratura, respectivamente.

O sinal QAM banda base pode ser representado por um sinal equivalente complexo, como apresentado na equação

$$\xi_{QAM}(t) = i_{QAM}(t) + j \cdot q_{QAM}(t), \quad (2.1.4)$$

em que $i_{QAM}(t) = a_i p(t)$ é a componente em fase banda base e $q_{QAM}(t) = b_i p(t)$ é a componente em quadratura banda base. O sinal modulado transmitido pode ser obtido multiplicando-se o sinal banda base $\xi_{QAM}(t)$ por uma exponencial complexa, conforme a equação

$$\phi_{QAM}(t) = \Re \{ \xi_{QAM}(t) \cdot e^{-j2\pi f_c t} \}. \quad (2.1.5)$$

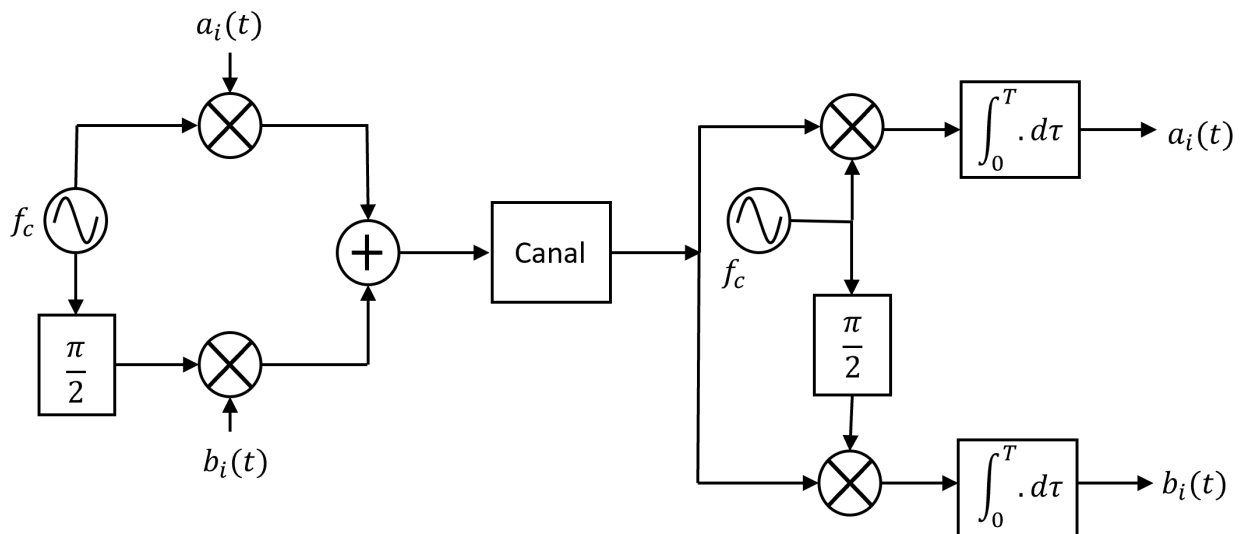
A partir da Equação 2.1.5, obtém-se a equação

$$\phi_{QAM}(t) = i_{QAM}(t) \cos(\omega_c t) + q_{QAM}(t) \sin(\omega_c t), \quad (2.1.6)$$

que é equivalente à Equação 2.1.3.

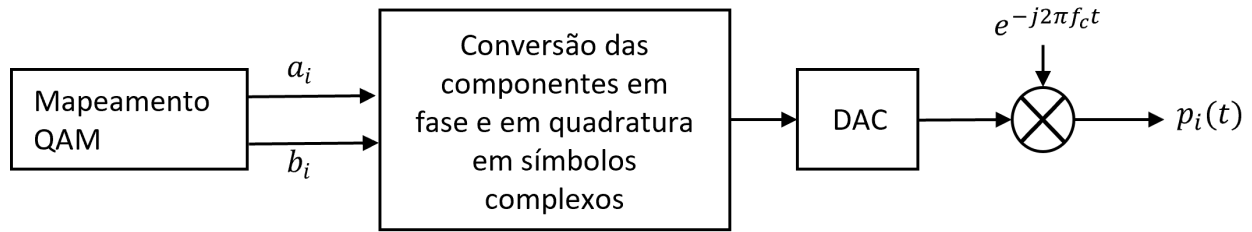
A Figura 3 mostra um sistema de transmissão QAM. É importante citar que, no demodulador, o sinal é primeiro multiplicado por um cosseno ou por um seno e, em seguida, integrado em um período de símbolo, assim, a componente indesejada se anula, restando apenas a componente desejada. A Figura 4 apresenta um modulador em que as componentes em fase e em quadratura são mapeadas, representadas na forma complexa, convertidas em sinal analógico por um conversor digital-analógico (DAC, *Digital-to-Analog Converter*) e, por fim, multiplicadas pela exponencial complexa.

Figura 3 – Sistema QAM.



fonte: Adaptado de (LATHI; DING, 2012).

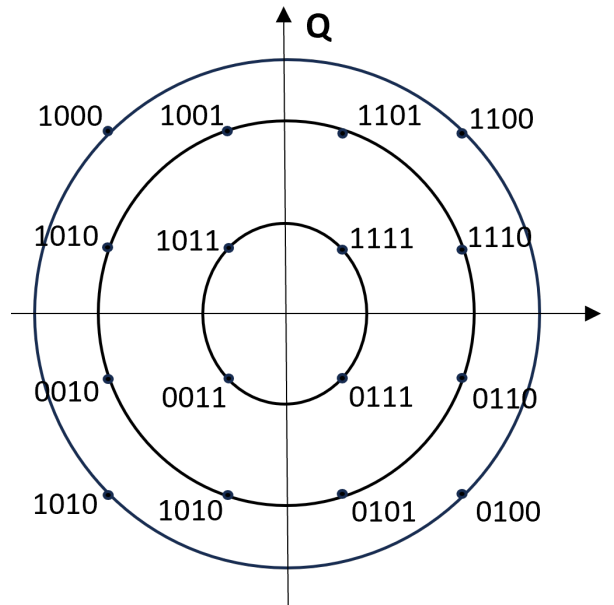
Figura 4 – Modulador QAM utilizando a forma complexa.



fonte: Adaptado de (PROAKIS, 2001).

A constelação de uma modulação QAM de ordem 16, com constelação quadrada, pode ser observada na Figura 5. A partir da figura, é possível observar que cada símbolo carrega a informação de $\log_2 16 = 4$ bits. Assim, em comparação com a modulação BPSK, a taxa de transmissão de bits de uma modulação 16-QAM pode ser até quatro vezes maior.

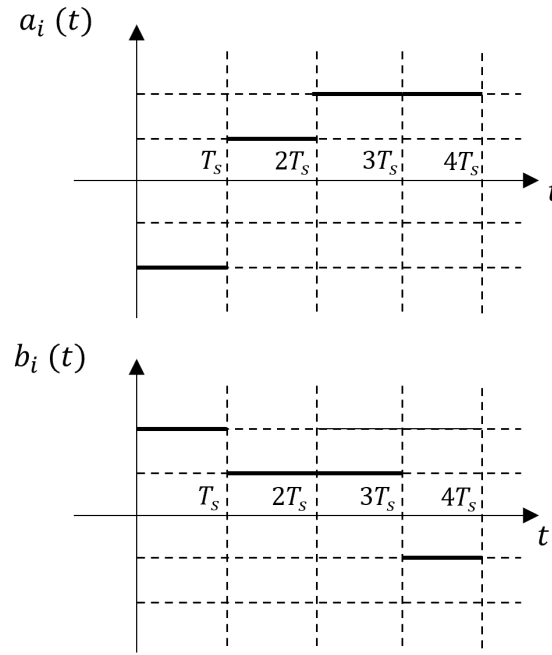
Figura 5 – Constelação 16-QAM.



fonte: Produção do próprio autor.

Para a transmissão de um sinal QAM, os bits são primeiramente mapeados nas amplitudes das componentes em fase e em quadratura. A Figura 6 mostra o mapeamento das amplitudes das componentes em fase e em quadratura da sequência de bits 1000 1111 1110 0110, utilizando a constelação da Figura 5, em que T_s representa o período de um símbolo.

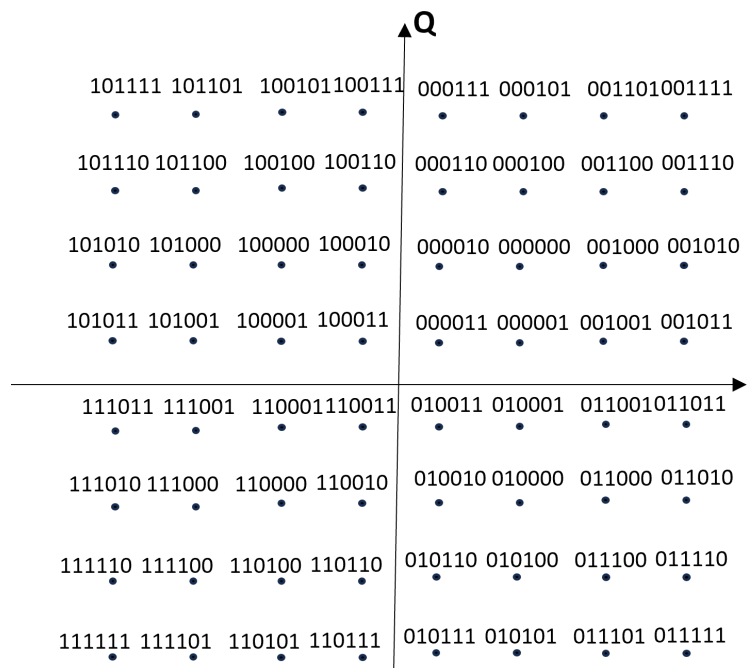
Figura 6 – Mapeamento das amplitudes das componentes em fase e em quadratura.



fonte: Produção do próprio autor.

Uma constelação QAM pode possuir um número maior de símbolos. A constelação QAM de ordem 64 é apresentada na Figura 7. Na modulação 64-QAM, cada símbolo pode carregar $\log_2 64 = 6$ bits, permitindo uma taxa de transmissão seis vezes maior em comparação com a modulação BPSK. A constelação 64-QAM possui uma maior probabilidade de erro de bit quando comparada à modulação 16-QAM, uma vez que a constelação 64-QAM é mais densa.

Figura 7 – Constelação 64-QAM.



fonte: Produção do próprio autor.

2.1.2 Densidade Espectral de Potência e Transmissão Multiportadora

Uma importante ferramenta para a análise de um sinal no domínio da frequência é a densidade espectral de potência (PSD, *Power Spectral Density*). De acordo com o Teorema de Wiener–Khinchin, a PSD de um sinal pode ser calculada a partir da função de autocorrelação de um sinal que é dada por

$$\mathcal{R}_g(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} g(t)g(t - \tau)dt. \quad (2.1.7)$$

Para calcular a PSD de um sinal basta calcular a transformada de Fourier da função de autocorrelação, conforme equação

$$\mathcal{S}_g(f) = FT\{\mathcal{R}_g(\tau)\}. \quad (2.1.8)$$

Considerando que os pulsos a serem transmitidos são retangulares, a função de autocorrelação do sinal modulado pode ser expressa pela equação

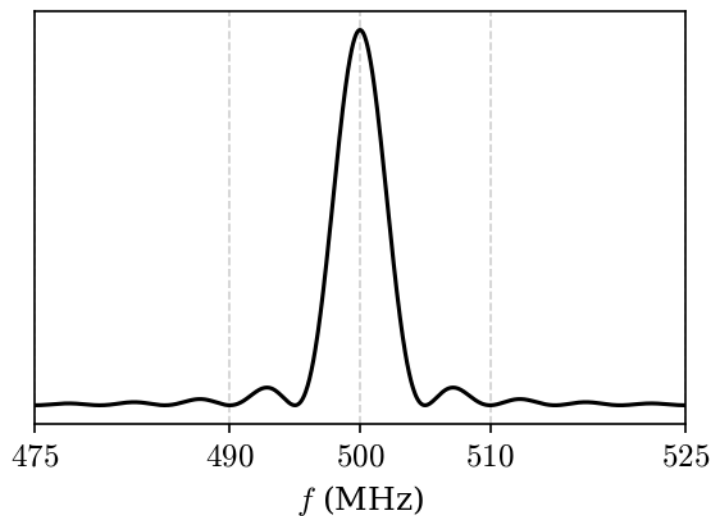
$$\mathcal{R}_g(\tau) = \text{triang}\left(\frac{\tau}{2T_s}\right) \cos(2\pi f_c \tau), \quad (2.1.9)$$

em que T_s representa o período de um símbolo. A PSD desse sinal pode ser calculada a partir da transformada de Fourier da Equação 2.1.9, conforme demonstrado na equação

$$\mathcal{S}_g(f) = FT\{\mathcal{R}_g(\tau)\} = \text{sinc}^2(T_s(f - f_c)). \quad (2.1.10)$$

A Figura 8 mostra o gráfico da PSD da Equação 2.1.10 para $f_c = 500$ MHz e $T_s = 0,2 \mu\text{s}$. Nessa figura, é possível observar que, na frequência da portadora f_c , o sinal assume seu valor máximo, enquanto nas frequências $f = f_c \pm n/T_s$, para $n \neq 0$, a função é nula. Essa propriedade é de grande importância em sistemas multiportadora, como o OFDM, pois permite transmitir e receber sinais que se sobrepõem em frequência sem interferência entre portadoras.

Figura 8 – PSD de um sinal modulado.

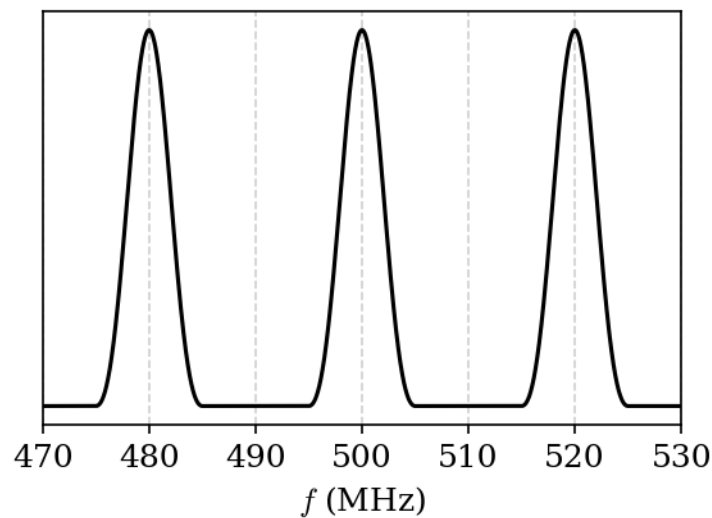


fonte: Produção do próprio autor.

A partir do conceito de densidade espectral de potência é possível compreender o funcionamento da multiplexação por divisão de frequência (FDM), técnica que permite a transmissão de diversos sinais por meio de um único canal. Nessa técnica, cada sinal é modulado por uma frequência portadora f_c distinta e, em seguida, todos os sinais modulados são somados e transmitidos. As portadoras são adequadamente espaçadas para evitar a sobreposição dos espectros de cada sinal modulado.

O sinal da Figura 8 pode ser filtrado de forma que apenas o lóbulo principal seja mantido, já que grande parte da energia do sinal está concentrada nesse lóbulo. O mesmo procedimento pode ser aplicado a sinais modulados com outras frequências de portadora, e, em seguida, todos os sinais podem ser somados, como ilustrado no gráfico da Figura 9.

Figura 9 – PSD de um sinal FDM.

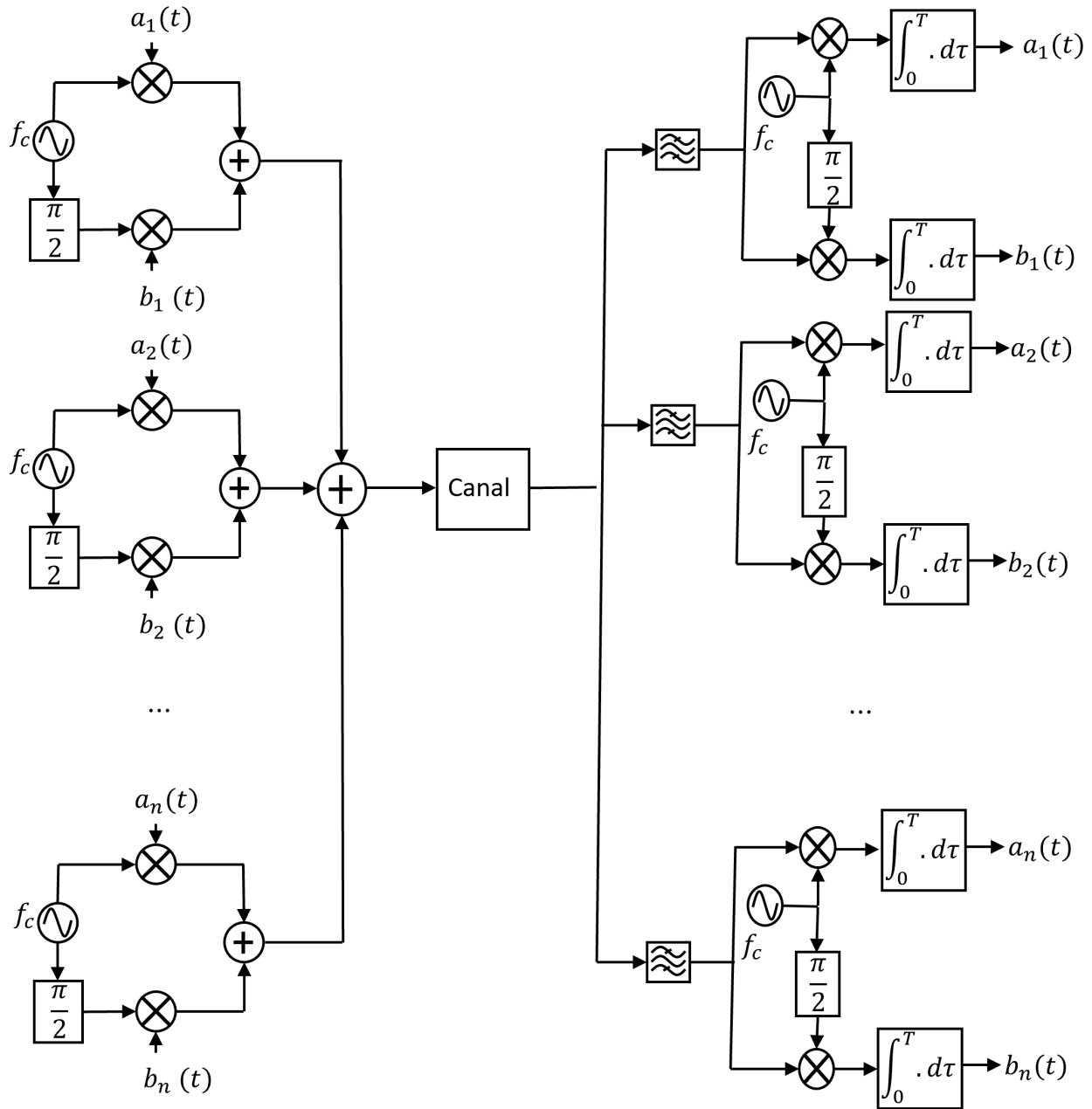


fonte: Produção do próprio autor.

Na Figura 9, é possível observar que as frequências das portadoras dos sinais são ortogonais, pois são espaçadas de n/T_s , em n é um número inteiro. No entanto, não possuem o espaçamento mínimo necessário, logo não se trata de um sinal OFDM. Também pode-se notar que os espectros dos sinais não se sobrepõem, o que possibilita filtrá-los e separar os diferentes sinais. Além disso, a figura evidencia a presença de uma banda de guarda entre os espectros, evitando interferência entre portadoras, embora essa banda de guarda reduza a eficiência espectral do sistema.

A Figura 10 mostra um esquema de um sistema FDM utilizando modulação QAM. A partir da figura, é possível observar que cada sinal é modulado por um modulador QAM e, em seguida, todos os sinais são somados e transmitidos pelo canal. No receptor, o sinal recebido é primeiramente filtrado para isolar o espectro do sinal desejado e, em seguida, cada sinal passa por um demodulador QAM.

Figura 10 – Esquema de um sistema FDM.



fonte: Adaptado de (LATHI; DING, 2012).

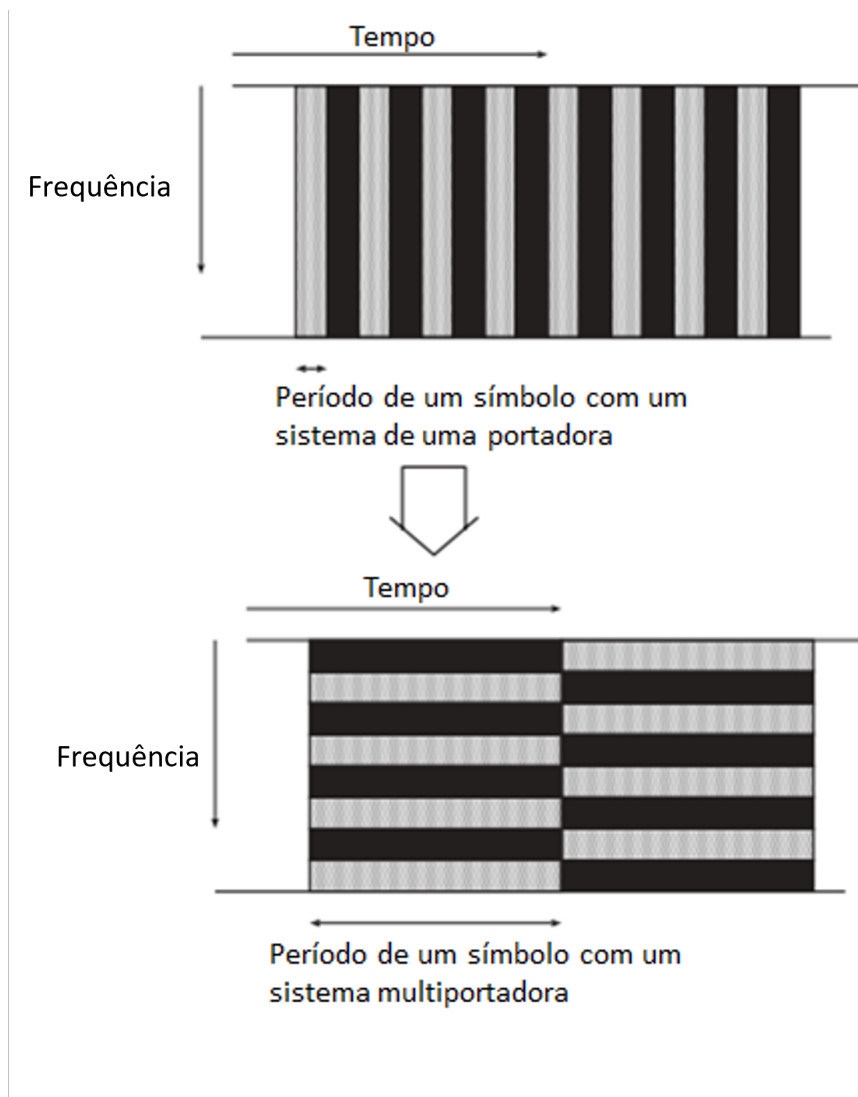
A transmissão multiportadora é uma técnica muito utilizada em sistemas de comunicações digitais modernos, que se baseia nos princípios da multiplexação por divisão de frequência para transmitir dados de forma eficiente e robusta em canais sujeitos ao desvanecimento seletivo em frequência. Diferentemente dos sistemas de portadora única, nos quais todos os dados são transmitidos em uma única portadora de alta taxa, a transmissão multiportadora divide o fluxo de dados em várias subportadoras, sendo que cada uma transmite apenas uma fração da taxa total. Essa abordagem reduz a influência do espalhamento por atraso, melhorando o desempenho em canais dispersivos.

Considerando um esquema de transmissão digital de portadora única com modulação PSK ou QAM de ordem M , a banda ocupada pelo sinal B_s é da ordem de T_s^{-1} . Por exemplo, para pulsos de cossenos levantados com fator de roll-off α_{ro} , a largura de banda é $B_s = (1 + \alpha_{ro})T_s^{-1}$. Para um canal com espalhamento por atraso (Delay Spread) τ_m , uma recepção livre de interferência intersimbólica só

é possível se $\tau_m \ll T_s$. Dessa forma, a taxa de bits $R_b = \log_2(M)T_s^{-1}$ em um esquema de portadora única é limitada pelo espalhamento por atraso.

A ideia de transmissão multiportadora traz uma solução para a limitação imposta pelo espalhamento por atraso. Dividindo um fluxo de dados em série de alta taxa em N_{mp} subfluxos em paralelo de menor taxa e transmitindo cada um com uma portadora distinta, o período de símbolo pode ser aumentado por um fator N_{mp} , como ilustrado na Figura 11. Esse aumento do período de símbolo permite uma taxa de bits N_{mp} vezes maior para um mesmo espalhamento por atraso, quando comparado a um esquema de portadora única. No entanto, é importante destacar que o fator N_{mp} não pode ser aumentado de forma arbitrária, pois um período de símbolo muito longo torna o sistema mais sensível ao tempo de coerência do canal.

Figura 11 – Período de símbolo de um sistema monoportadora e de um sistema multiportadora.



fonte: Adaptado de (SCHULZE; LÜDERS, 2005).

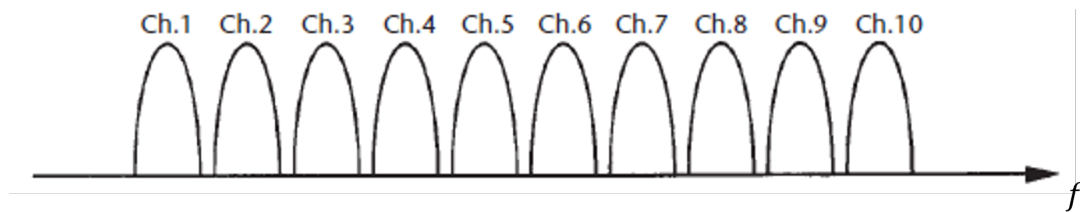
2.2 OFDM

A técnica OFDM é um caso especial de transmissão multiportadora, em que um fluxo de alta taxa de dados em série é dividido em subfluxos de menor taxa em paralelo, e cada subfluxo é modulado por

uma portadora diferente, sendo que as portadoras são ortogonais entre si, com espaçamento mínimo em frequência que assegura a condição de ortogonalidade. Uma das principais vantagens do OFDM é sua extrema robustez ao espalhamento por atraso e ao desvanecimento seletivo em frequência, além de ser resistente às possíveis interferências intersimbólicas. Outro motivo importante para o uso dessa técnica é a maior eficiência espectral que ela oferece em comparação com outras técnicas de multiplexação.

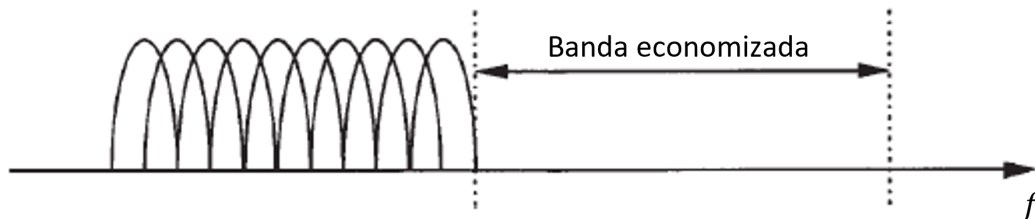
Em um esquema básico de transmissão em paralelo, utilizando a técnica FDM, a banda total do sinal é dividida em N_{mp} subcanais em frequência que não se sobrepõem e são separados por uma banda de guarda, como pode ser observado na Figura 12. Porém, como visto na seção anterior, isso causa uma baixa eficiência espectral. Em um sistema OFDM, por outro lado, o espectro dos diferentes subcanais pode se sobrepor, como ilustrado na Figura 13. Dessa forma, é possível reduzir a largura de banda total do sinal em até 50% quando comparado à técnica FDM.

Figura 12 – Espectro de um sinal FDM.



fonte: Adaptado de (PRASAD, 2004).

Figura 13 – Espectro de um sinal OFDM.



fonte: Adaptado de (PRASAD, 2004).

A palavra ortogonal em OFDM indica que os sinais modulados por diferentes portadoras devem ser ortogonais durante o intervalo de tempo de um símbolo. Dois sinais $f_1(t)$ e $f_2(t)$ são ortogonais em um intervalo de símbolo Δt_s se satisfizerem a condição

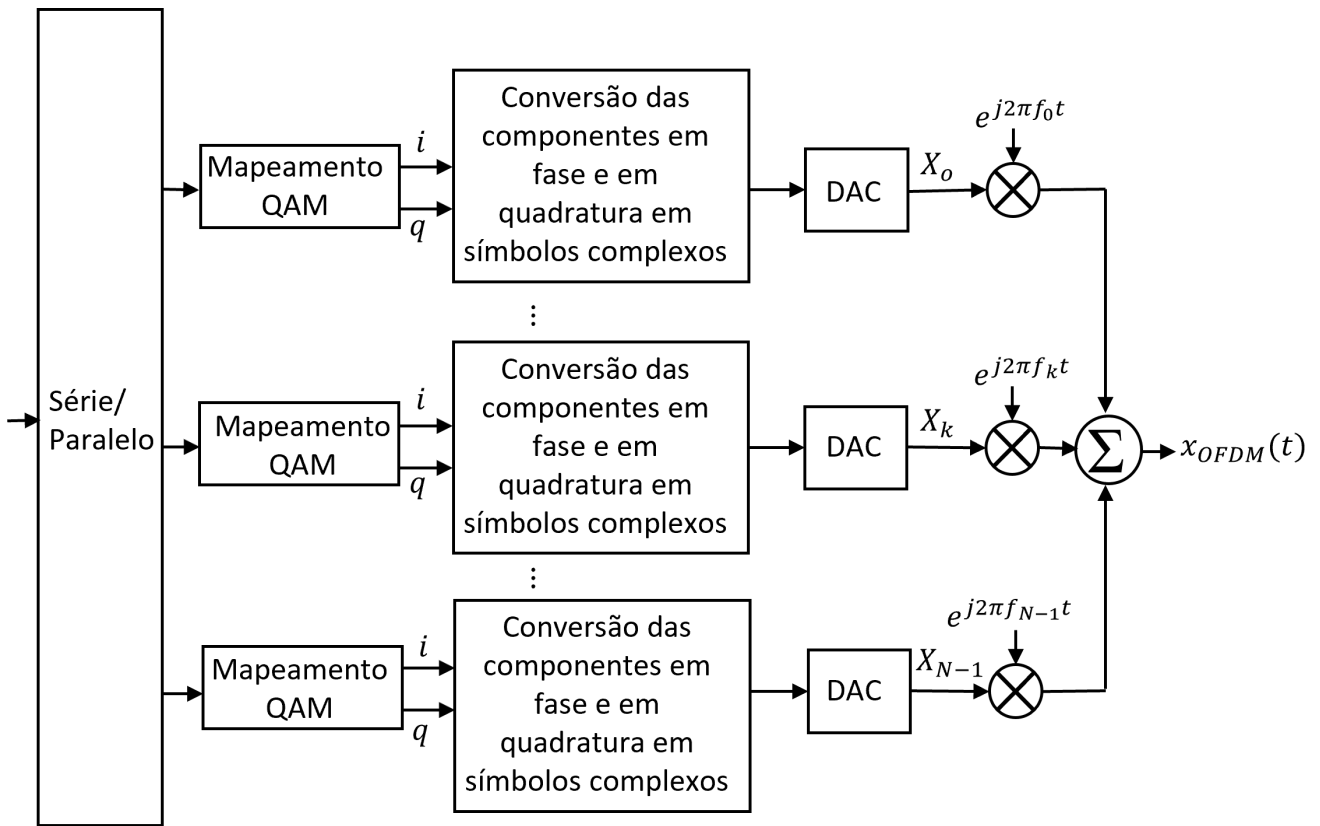
$$\int_{t_0}^{t_0 + \Delta t_s} f_1(t) f_2^*(t) dt = 0. \quad (2.2.1)$$

Se cada portadora tiver, no domínio do tempo, um número inteiro de ciclos dentro do intervalo de símbolo Δt_s , então a condição da Equação 2.2.1 é satisfeita. A partir disso, para atingir a condição de ortogonalidade e a menor separação em frequência entre duas portadoras, no OFDM cada portadora é espaçada de $1/T_s$ em relação às suas adjacentes.

A Figura 14 mostra o esquema de um transmissor OFDM, no qual um fluxo de alta taxa de dados é dividido em N subfluxos de menor taxa. Cada um desses subfluxos é modulado utilizando um QAM com uma portadora diferente e, em seguida, todos os sinais correspondentes a cada subfluxo

são somados e transmitidos pelo canal. Vale ressaltar que as portadoras são ortogonais entre si, e o espaçamento entre elas corresponde ao menor valor que garante a condição de ortogonalidade.

Figura 14 – Transmissor OFDM.



fonte: Adaptado de (SCHULZE; LÜDERS, 2005).

O sinal $x_{OFDM}(t)$ na saída do transmissor OFDM pode ser dado por

$$x_{OFDM}(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k t}, \quad (2.2.2)$$

em que f_k representa a frequência da k -ésima portadora. Considerando que $f_k = k\Delta f_s$ e que $\Delta f_s = \frac{1}{T_s}$ corresponde ao espaçamento entre duas portadoras adjacentes, a Equação 2.2.2 pode ser reformulada, resultando na equação

$$x_{OFDM}(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi k \frac{t}{T_s}}. \quad (2.2.3)$$

Amostrando o sinal no domínio do tempo, obtém-se

$$x_{OFDM_s}(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{k}{T_s} n \Delta t_s}, \quad (2.2.4)$$

em que $t_n = n \Delta t_s$ é o instante de tempo da n -ésima amostra e Δt_s é a diferença de tempo entre duas amostras adjacentes. Considerando que $\Delta t_s = \frac{T_s}{N}$, a Equação 2.2.4 pode ser reescrita conforme a

equação

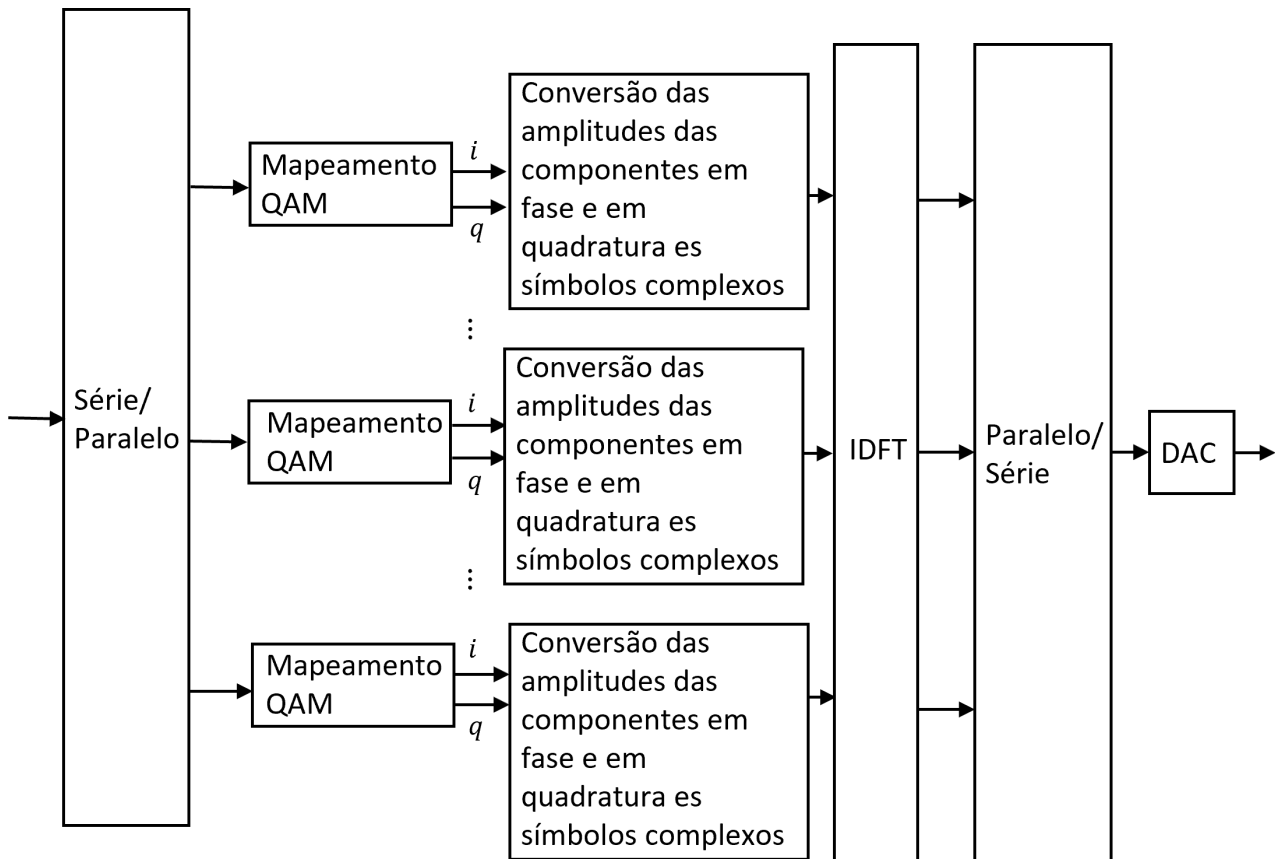
$$x_{OFDM_s}(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{k}{N} n}. \quad (2.2.5)$$

E, a IDFT é dada por

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{k}{N} n}. \quad (2.2.6)$$

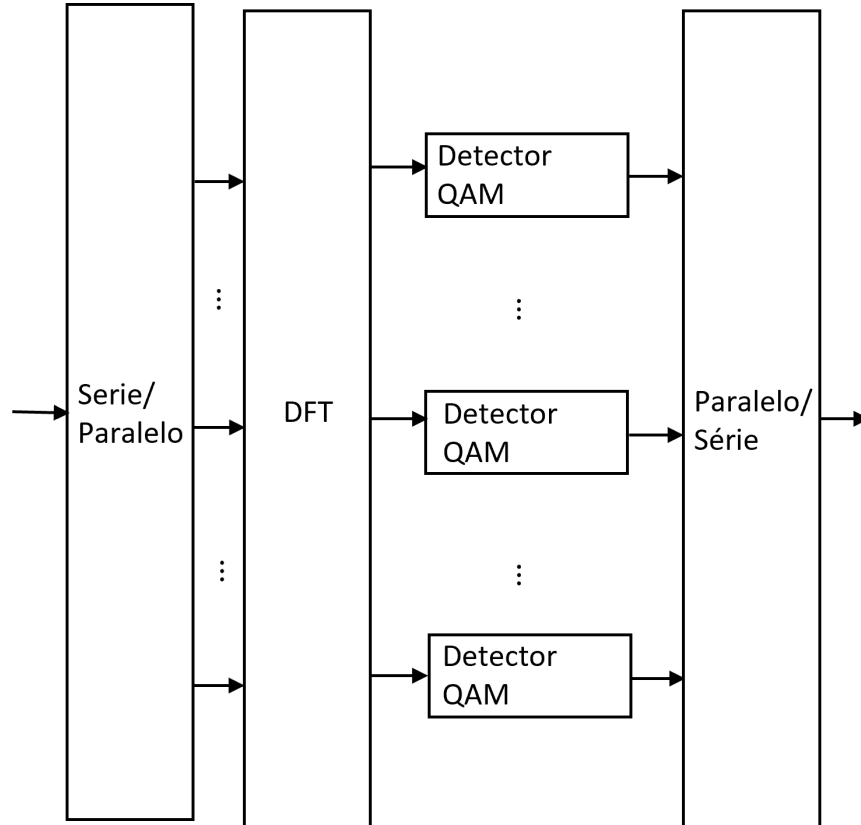
É possível notar que a Equação 2.2.5 é equivalente à equação da IDFT, sendo que a única diferença é a constante $\frac{1}{N}$, a qual pode ser desconsiderada. Dessa forma, o banco de osciladores locais, que seria necessário para modular os sinais nas frequências das portadoras, pode ser substituído pela IDFT, como ilustrado na Figura 15. No receptor, a demodulação dos sinais pode ser realizada por meio de uma DFT, como mostrado na Figura 16. Para reduzir o custo computacional da IDFT e DFT, essas podem ser calculadas de forma mais eficiente por meio da transformada rápida de Fourier (FFT) e sua inversa (IFFT). Enquanto a DFT requer um número de operações da ordem de N^2 , a FFT reduz esse número para a ordem de $N \log(N)$ operações.

Figura 15 – Transmissor OFDM utilizando a IDFT.



fonte: Adaptado de (PRASAD, 2004).

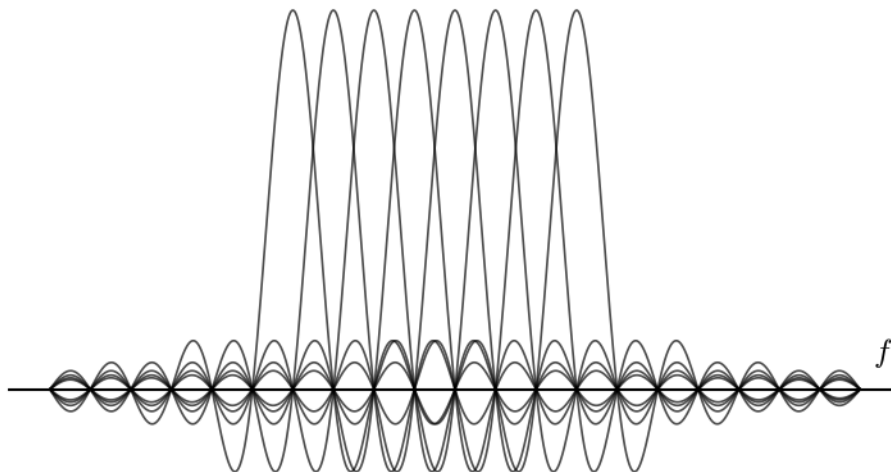
Figura 16 – Receptor OFDM.



fonte: Adaptado de (PRASAD, 2004).

Analisando a PSD da Figura 8, observa-se que, na frequência da portadora f_c , o sinal assume valores diferentes de zero, enquanto em $f = f_c \pm n/T_s$, para $n \neq 0$, a função é nula. A partir dessa propriedade, nota-se, na Figura 17, que, se o sinal OFDM for amostrado corretamente no domínio da frequência, é possível recuperá-lo sem interferência interportadora. Por exemplo, se a amostragem for realizada exatamente nas frequências das portadoras, a influência das portadoras adjacentes é nula, permitindo a recuperação do sinal sem interferência interportadora.

Figura 17 – Espectro das diferentes portadoras.



fonte: Adaptado de (VITI, 2010).

3 TEORIA DA INFORMAÇÃO E CODIFICAÇÃO

A Teoria da Informação estuda a modelagem e a análise matemática dos sistemas de comunicação, buscando responder duas questões fundamentais: qual é o limite mínimo de compressão de um sinal e qual é a taxa máxima de transmissão confiável em um canal ruidoso. Para isso, são introduzidas medidas essenciais, como entropia e informação mútua, que permitem o estudo de diferentes tipos de canais de comunicação, com destaque para o canal Gaussiano, referência central em análises teóricas e práticas. A Teoria da Informação também abrange o estudo de códigos corretores de erro, entre os quais se destacam os códigos de bloco linear, classe que inclui os códigos LDPC considerados neste trabalho.

Na Seção 3.1 serão abordados os principais conceitos e definições relacionados a medidas de informação, com foco na apresentação do canal Gaussiano e do cálculo da sua capacidade. A Seção 3.2 apresentará conceitos e definições sobre códigos de bloco linear que serão importantes para o desenvolvimento deste trabalho. As definições e conceitos utilizados neste capítulo podem ser encontrados em (HAYKIN, 2001; COVER; THOMAS, 2006; LIN; COSTELLO, 2004).

3.1 MEDIDAS DE INFORMAÇÃO

Dado um experimento aleatório modelado por uma variável aleatória discreta e sem memória X , a qual tem o alfabeto

$$\mathcal{X} = \{x_0, x_1, \dots, x_{K-1}\}, \quad (3.1.1)$$

com probabilidades

$$p(X) = P(X = x_k) = p_k, \quad k = 0, 1, \dots, K - 1, \quad (3.1.2)$$

e função massa de probabilidade $p(x)$. A quantidade de informação, que é a quantidade de incerteza de um processo que ocorre com alguma probabilidade, pode ser dada em função de $p(x)$ e é definida de maneira mais formal a seguir.

Definição 3.1.1. A quantidade de informação $I(X_k)$ ganhada após a observação do evento $X = x_k$, que ocorre com probabilidade $p_k \neq 0$ é definida como

$$I(x_k) = \log \left(\frac{1}{p_k} \right). \quad (3.1.3)$$

A entropia mede o conteúdo médio de informação por símbolo da fonte e pode ser dada pela esperança de $I(x_k)$, ou seja, ela é a média da quantidade de informação.

Definição 3.1.2. A entropia $H(X)$ de uma variável aleatória X com alfabeto $\mathcal{X}$ e função massa de probabilidade $p(x)$ é definida por

$$H(X) = \mathbb{E}[I(x_k)] = - \sum_{k=0}^{K-1} p_k \log \left(\frac{1}{p_k} \right). \quad (3.1.4)$$

Quando consideramos duas variáveis aleatórias discretas com alfabetos $\mathcal{X}$ e $\mathcal{Y}$, sua distribuição de probabilidade conjunta $p(x, y)$ e distribuições de probabilidade condicionais $p(x|y)$ e $p(y|x)$, é possível calcular a entropia conjunta e as entropias condicionais. A entropia conjunta mede a incerteza total associada ao resultado do par (X, Y) .

Definição 3.1.3. *Sejam X e Y duas variáveis aleatórias discretas com alfabetos $\mathcal{X}$ e $\mathcal{Y}$, respectivamente. Seja $p(x, y)$ a distribuição de probabilidade conjunta do par (X, Y) , onde $p(x, y) = P(X = x, Y = y)$. A entropia conjunta $H(X, Y)$ é definida como*

$$H(X, Y) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y). \quad (3.1.5)$$

Já a entropia condicional $H(Y|X)$ mede a incerteza que resta sobre a variável Y , após observar o resultado da variável X .

Definição 3.1.4. *Se $(X, Y) \sim p(x, y)$, a entropia condicional $H(Y|X)$ é definida como*

$$H(Y|X) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x). \quad (3.1.6)$$

Outra medida de informação importante é a informação mútua, que é a quantidade de informação que uma variável aleatória X contém sobre outra variável aleatória Y .

Definição 3.1.5. *Considerando duas variáveis aleatórias X e Y com uma função de massa de probabilidade conjunta $p(x, y)$ e funções de massa de probabilidade marginal $p(x)$ e $p(y)$. A informação mútua $I(X; Y)$ é definida como*

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \quad (3.1.7)$$

Considerando que $H(Y)$ é a incerteza inicial sobre Y , e $H(Y|X)$ é a incerteza restante sobre Y depois que se descobre X , então a informação mútua $I(X; Y)$ será exatamente a quantidade de incerteza que foi removida pelo conhecimento de X , ou seja,

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \quad (3.1.8)$$

Os conceitos e definições apresentados anteriormente consideram que as variáveis aleatórias são discretas. A entropia, considerando uma variável aleatória contínua, também é chamada de entropia diferencial e sua definição pode ser observada a seguir.

Definição 3.1.6. *A entropia diferencial $h(X)$ de uma variável aleatória contínua X com função densidade de probabilidade $f(x)$ é definida como*

$$h(X) = - \int_S f(x) \log f(x) dx, \quad (3.1.9)$$

onde S é o conjunto de suporte da variável aleatória.

Assim como no caso discreto, é possível estender a definição de entropia diferencial de uma única variável aleatória para duas variáveis aleatórias. A informação mútua também pode ser calculada utilizando variáveis aleatórias contínuas, nesse caso, essa medida de informação também é conhecida como informação mútua diferencial e é definida formalmente a seguir.

Definição 3.1.7. A informação mútua diferencial $i(X; Y)$ entre duas variáveis aleatórias com função densidade de probabilidade conjunta $f(x, y)$ é definida como

$$i(X; Y) = \int \int f(x, y) \log \frac{f(x, y)}{f(x)f(y)} dx dy. \quad (3.1.10)$$

É importante notar que nas Equações 3.1.3, 3.1.4, 3.1.5, 3.1.6, 3.1.7, 3.1.9 e 3.1.10 o logaritmo pode ser expresso em qualquer base, sendo que a escolha da base apenas altera a unidade resultante. Quando se utiliza a base 2, a informação é medida em *bits*, com a base e , é expressa em *nats*, e, ao empregar a base 10, a unidade correspondente é *hartleys*.

A partir do conceito de informação mútua, podemos calcular a capacidade de um canal. Considerando um canal com uma variável aleatória de entrada X e uma variável aleatória de saída Y , a capacidade de um canal é a informação mútua máxima sobre todas as funções densidade de probabilidade de entrada $f(x)$. A capacidade de um canal é definida como segue.

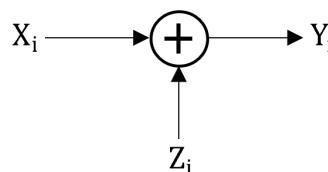
Definição 3.1.8. Seja um canal com uma variável aleatória de entrada X e uma variável aleatória de saída Y . Considerando que a informação mútua diferencial entre duas variáveis aleatórias é $I(X; Y)$, a capacidade de um canal é definida como

$$C = \max_{f(x)} I(X; Y), \quad (3.1.11)$$

onde a maximização é sobre todas as funções densidade de probabilidade de entrada $f(x)$.

Um dos canais mais importantes no caso contínuo é o canal Gaussiano, ilustrado na Figura 18. Considerando um processo aleatório ergódico de média zero e limitado em banda $X(t)$. A variável aleatória contínua X_i é obtida pela amostragem uniforme do processo $X(t)$. O canal Gaussiano é discreto no tempo e sua saída é dada pela variável aleatória contínua Y_i que é a soma da entrada X_i e do ruído Z_i . O ruído é do tipo AWGN (*Additive White Gaussian Noise*), isso significa que ele é aditivo, Gaussiano (possui distribuição normal) e branco (o que indica que sua densidade espectral de potência é constante em todas as frequências). Considera-se também que o ruído AWGN tem média 0, variância N_{n_0} , é sorteado de forma i.i.d. (independente e identicamente distribuído) de uma distribuição Gaussiana com variância N_{n_0} e é independente de X_i .

Figura 18 – Canal Gaussiano.



fonte: Produção do próprio autor.

Nota-se que, caso a variância do ruído seja zero, o receptor consegue receber o símbolo transmitido perfeitamente e, como X_i pode assumir qualquer valor real, o canal pode transmitir um número real arbitrário sem erro, logo a capacidade desse canal é infinita. Se a variância do ruído for diferente de zero e não houver restrições na entrada, é possível escolher um subconjunto infinito de entradas arbitrariamente distantes entre si, de modo que sejam distinguíveis na saída com probabilidade de erro tão pequena quanto desejado, logo, a capacidade do canal também é dita como infinita. Portanto, se $N_{n0} \neq 0$, é necessária alguma restrição na entrada. A restrição mais comum na entrada do canal é a restrição de potência $\mathbb{E}[X^2] \leq P$. A capacidade do canal, em bits por uso do canal em bits/amostra, utilizando essa restrição pode ser dada pela equação

$$C = \max_{\mathbb{E}[X^2] \leq P} I(X; Y) = \frac{1}{2} \log_2 \left(1 + \frac{P}{N_{n0}} \right). \quad (3.1.12)$$

Caso o canal também seja limitado em banda, a capacidade do canal em bits por segundo em função da banda do canal B_c pode ser dada por

$$C = B_c \log_2 \left(1 + \frac{P}{N_{n0}} \right). \quad (3.1.13)$$

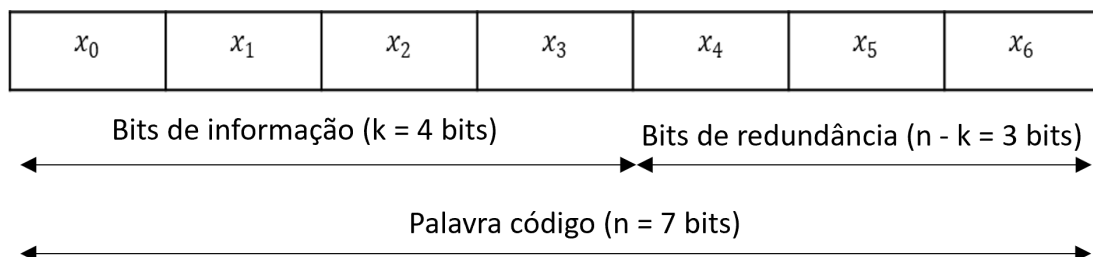
3.2 CÓDIGOS DE BLOCO LINEAR

Em sistemas de comunicações reais, a presença de possíveis perturbações no canal como, por exemplo, ruído, interferências, desvanecimentos, entre outras, pode acarretar em possíveis erros na sequência de bits recebida no receptor. Uma maneira de contornar esse problema é a utilização de uma codificação de canal que tem o objetivo central de detectar esses erros e corrigi-los.

Os códigos corretores de erro podem ser divididos em códigos de bloco e códigos convolucionais, diferenciando-se principalmente pelo fato de que os códigos de bloco codificam mensagens de comprimento fixo, enquanto os códigos convolucionais operam sobre sequências de comprimento variável. Neste trabalho, estamos interessados nos códigos de bloco.

Na codificação por bloco, o transmissor mapeia uma sequência de dados de entrada em outra sequência que será enviada pelo canal. Nesse processo, a sequência de dados de entrada é dividida em blocos de k bits e são adicionados a cada um desses blocos $n - k$ bits de paridade, dando origem a uma palavra-código de n bits. A Figura 19 mostra um bloco com 4 bits de informação x_0, x_1, x_2 e x_3 e 3 bits de paridade x_4, x_5 e x_6 .

Figura 19 – Estrutura de um bloco.



fonte: Produção do próprio autor.

Na etapa de decodificação, o receptor faz o mapeamento inverso da sequência de bits obtida na saída do canal. A presença de bits de paridade auxilia na detecção e correção de possíveis erros introduzidos pelo canal.

A taxa de um código é uma razão adimensional e representa a quantidade média de bits de informação transmitidos.

Definição 3.2.1. A taxa do código R para um código de bloco com parâmetros (n, k) é definida como

$$R = \frac{k}{n}. \quad (3.2.1)$$

De acordo com o Teorema da Codificação de Canal de Shannon, dado um canal com capacidade C e uma fonte com taxa R , então, 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.

Uma classe dos códigos de bloco são os códigos de bloco linear. Um código é linear se a soma binária de quaisquer duas palavras-código que pertencem ao código produz uma terceira palavra, a qual também pertence ao código. A definição de códigos de bloco linear pode ser observada a seguir.

Definição 3.2.2. Um código de bloco linear C_{LBC} de comprimento n e dimensão k , sobre o corpo finito $GF(q)$ com q elementos, é um subespaço vetorial k -dimensional do espaço vetorial n -dimensional $GF(q)^n$.

Um código de bloco linear pode ser denotado pelos seus parâmetros (n, k) . Se um vetor $\mathbf{c}$ pertence ao código, ou seja $\mathbf{c} \in C_{LBC}$, esse vetor é uma palavra-código. O número de palavras-código possíveis de um C_{LBC} de parâmetros (n, k) sobre $GF(q)$ é de q^k palavras-código e o número de palavras possíveis com comprimento n é de q^n .

Um dos parâmetros de grande importância em códigos de bloco linear é a distância mínima de Hamming, pois esse parâmetro determina as capacidades de detecção e correção de erros em um código de bloco linear.

Definição 3.2.3. Sejam $\mathbf{c}_1 = (c_{1,1}, c_{1,2}, \dots, c_{1,n})$ e $\mathbf{c}_2 = (c_{2,1}, c_{2,2}, \dots, c_{2,n})$ duas n -uplas (vetores) de n símbolos. A distância de Hamming entre $\mathbf{c}_1$ e $\mathbf{c}_2$, denotada por $d(\mathbf{c}_1, \mathbf{c}_2)$, é o número de posições de símbolo nas quais $\mathbf{c}_1$ e $\mathbf{c}_2$ se diferenciam.

A distância mínima de Hamming de um código C_{LBC} é a menor distância de Hamming possível entre duas palavras que pertencem ao código e é definida a seguir.

Definição 3.2.4. Seja $\mathbf{c}_1 = (c_{1,1}, c_{1,2}, \dots, c_{1,n})$ e $\mathbf{c}_2 = (c_{2,1}, c_{2,2}, \dots, c_{2,n})$ duas n -uplas (vetores) de n símbolos e $\mathbf{c}_1, \mathbf{c}_2 \in C_{LBC}$ a distância mínima de Hamming d_{min} é definida como

$$d_{min} = \min\{d(\mathbf{c}_i, \mathbf{c}_j) \mid \mathbf{c}_i, \mathbf{c}_j \in C, \mathbf{c}_i \neq \mathbf{c}_j\}. \quad (3.2.2)$$

Analisar a distância de Hamming entre todas as palavras-código pode se tornar muito trabalhoso, devido a isso é determinado o peso mínimo de Hamming o qual é igual a distância mínima de Hamming. O peso de hamming de uma palavra-código é o número de elementos diferentes de zero nessa palavra e é definido a seguir.

Definição 3.2.5. Para um código de bloco linear C_{LBC} com parâmetros (n, k) , o *Peso de Hamming* de uma palavra-código $\mathbf{c} = (c_1, c_2, \dots, c_n)$, denotado por $w(\mathbf{c})$, é definido como o número de símbolos não-nulos na palavra-código.

O peso mínimo de Hamming é o menor peso de Hamming possível entre todas as palavras-código de um código de bloco linear e é definido de maneira formal na Definição 3.2.6.

Definição 3.2.6. O *peso mínimo* de um código de bloco linear C_{LBC} , denotado por w_{min} , é definido o menor peso de Hamming encontrado entre todas as palavras-código não-nulas do código.

A capacidade de detecção e correção de erros em um código de bloco linear pode ser dada em função da distância mínima de Hamming. Um código de bloco linear pode detectar até $d_{min} - 1$ erros dentro de um bloco e pode corrigir até $\lfloor \frac{d_{min}-1}{2} \rfloor$ erros dentro de um bloco.

O processo de codificação de uma mensagem $\mathbf{u}$ em uma palavra-código pode ser realizado através da multiplicação de $\mathbf{u}$ por uma matriz geradora, definida a seguir.

Definição 3.2.7. Seja um código de bloco linear C_{LBC} de parâmetros (n, k) , que é um subespaço k -dimensional S_c do espaço vetorial n -dimensional V_n sobre $GF(2)$, uma matriz geradora $\mathbf{G}$ é uma matriz de dimensão $k \times n$ cujas linhas formam um conjunto de vetores-base para o subespaço S_c .

A matriz geradora $\mathbf{G}$ pode ser dada na forma padrão ou sistemática conforme a equação

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}_s], \quad (3.2.3)$$

em que $\mathbf{I}_k$ é uma matriz identidade de dimensões $k \times k$ e a matriz $\mathbf{P}_s$ é conhecida como matriz de paridade e tem dimensões $k \times (n - k)$.

No receptor, a verificação dos blocos recebidos é realizada por meio da matriz controle de paridade, a qual é definida a seguir.

Definição 3.2.8. Para um código de bloco linear C_{LBC} com parâmetros (n, k) , a matriz controle de paridade $\mathbf{H}$ é uma matriz de dimensão $(n - k) \times n$ que define o espaço nulo do código.

Essa matriz tem a propriedade de que para qualquer palavra-código válida $\mathbf{c}$ pertencente ao código C_{LBC} , deve ser satisfeita a condição

$$\mathbf{c} \cdot \mathbf{H}^T = \mathbf{0}, \quad (3.2.4)$$

em que $\mathbf{H}^T$ é a matriz controle de paridade transposta.

A matriz controle de paridade também pode ser dada na forma padrão ou sistemática da forma

$$\mathbf{H} = [-\mathbf{P}_s^T \mid \mathbf{I}_{n-k}], \quad (3.2.5)$$

em que $\mathbf{I}_{n-k}$ é uma matriz identidade de ordem $(n - k) \times (n - k)$ e $\mathbf{P}_s^T$ é a matriz $\mathbf{P}_s$ transposta.

4 CÓDIGOS LDPC

Os códigos de verificação de paridade de baixa densidade (LDPC) são uma classe de códigos de bloco linear. O nome LDPC vem de uma das principais características dessa classe, que é ter uma matriz controle de paridade com uma baixa densidade de 1's em comparação com a quantidade de 0's (LEINER, 2005). Tais códigos foram inicialmente propostos por Robert G. Gallager em sua dissertação de doutorado publicada em 1963 (GALLAGER, 1963). Porém, esses códigos foram amplamente ignorados por aproximadamente 35 anos. O estudo de códigos LDPC ressurgiu em meados dos anos 90 com o trabalho de David J.C. MacKay e Radford M. Neal (MACKAY; NEAL, 1996), no qual demonstraram que os códigos LDPC podem alcançar uma performance substancialmente melhor do que a dos códigos convolucionais e concatenados padrão e também podem alcançar uma performance quase tão próxima do limite de Shannon quanto a dos códigos Turbo.

O projeto e a implementação de códigos LDPC de forma aleatória podem ser extremamente complexos e até inviáveis quando as matrizes de paridade $\mathbf{H}$ possuem dimensões muito grandes. Os códigos QC-LDPC baseados em protógrafos oferecem uma solução ao substituir a construção direta de uma grande matriz $\mathbf{H}$ pelo projeto de uma matriz protógrafo muito menor, cuja análise, projeto e implementação são muito mais simples.

A decodificação utilizando códigos LDPC pode ser realizada por diferentes algoritmos, que variam em termos de complexidade e desempenho. Um algoritmo muito utilizado é o algoritmo Soma e Produto (SPA) que apresenta uma alta eficiência em termos de correção de erros, porém apresenta alta complexidade computacional. Para reduzir esse custo é possível utilizar o algoritmo Min-Sum que tem uma menor complexidade e apresenta um menor custo computacional. A convergência dos algoritmos de decodificação também pode ser mais rápida utilizando a decodificação em camadas, na qual a matriz de verificação de paridade é dividida em camadas que são processadas sequencialmente.

Neste capítulo serão apresentados conceitos referentes aos códigos LDPC e aos códigos QC-LDPC. Inicialmente, na Seção 4.1 serão apresentados os conceitos e definições básicos utilizados em códigos LDPC. A Seção 4.2 apresentará os códigos QC-LDPC baseados em protógrafos. E, por fim, a Seção 4.3 apresentará os algoritmos de decodificação utilizados nesse trabalho. Os conteúdos abordados neste capítulo podem ser encontrados em (LEINER, 2005; LEGG, 2007; MATTOS, 2012; NAZARETH, 2010; CORDEIRO, 2010; JOHNSON, 2006; RYAN; LIN, 2009; LIN; COSTELLO, 2004; HUFFMAN; PLESS, 2003; THORPE, 2003; MITCHELL; SMARANDACHE; COSTELLO, 2014).

4.1 DEFINIÇÕES E CONCEITOS BÁSICOS EM CÓDIGOS LDPC

Para o entendimento de códigos LDPC, são necessários alguns conceitos prévios. Um deles é o conceito de densidade de uma fonte, outro conceito importante é o de *overlap*. Tais conceitos são definidos a seguir.

Definição 4.1.1. *A densidade de uma fonte binária é a fração esperada de bits iguais a 1 na sequência.*

Definição 4.1.2. *O overlap de duas sequências binárias é a fração esperada das posições de bits onde ambas as sequências possuem um 1.*

Assim, podemos definir os códigos LDPC da seguinte forma.

Definição 4.1.3. *Um código LDPC é definido como o espaço nulo de uma matriz controle de paridade $\mathbf{H}$ que possui um número de 1's em cada linha e em cada coluna da matriz $\mathbf{H}$ pequeno quando comparado ao comprimento do código.*

De acordo com a Definição 4.1.3, a matriz $\mathbf{H}$ de um código LDPC apresenta uma baixa densidade, ou seja, é uma matriz esparsa. No projeto de uma matriz $\mathbf{H}$ é desejado que o overlap entre duas linhas dessa matriz não seja maior que 1, pois esse fator pode afetar significativamente o algoritmo de decodificação a ser implementado.

Uma das ferramentas fundamentais para representar códigos LDPC é o grafo de Tanner, introduzido por Robert M. Tanner em 1981 (TANNER, 1981). Um grafo de Tanner é um tipo especial de grafo bipartido no qual os vértices são organizados em dois conjuntos distintos, conectados entre si por arestas. As definições formais de grafo, grafo bipartido e grafo de Tanner são apresentadas a seguir.

Definição 4.1.4. *Um grafo $D(V, A)$ é um conjunto de pontos, chamados de nós ou vértices, conectados por linhas, chamadas de arestas. Onde V é um conjunto finito e não vazio que representa os nós e A é um conjunto de pares não ordenados de elementos de V que representa o conjunto de arestas.*

Definição 4.1.5. *Um grafo é bipartido se existe uma bipartição do seu conjunto de nós tal que cada aresta tem uma ponta em uma das partes da bipartição e a outra ponta na outra parte.*

Definição 4.1.6. *Um grafo de Tanner, para um código de bloco linear de parâmetros (n, k) , é um grafo bipartido em que os dois tipos de nós são os nós de variável, que correspondem aos n bits da palavra-código e os nós de paridade (ou nós de função), que correspondem às $m = n - k$ equações de paridade. Um nó de variável é conectado a outro nó de paridade por uma aresta somente se o bit correspondente ao nó de variável fizer parte da equação de paridade correspondente ao nó de paridade.*

Os grafos de Tanner podem ser representados na forma matricial, a partir de uma matriz de adjacência, a qual pode ser vista como a matriz controle de paridade $\mathbf{H}$ do código LDPC.

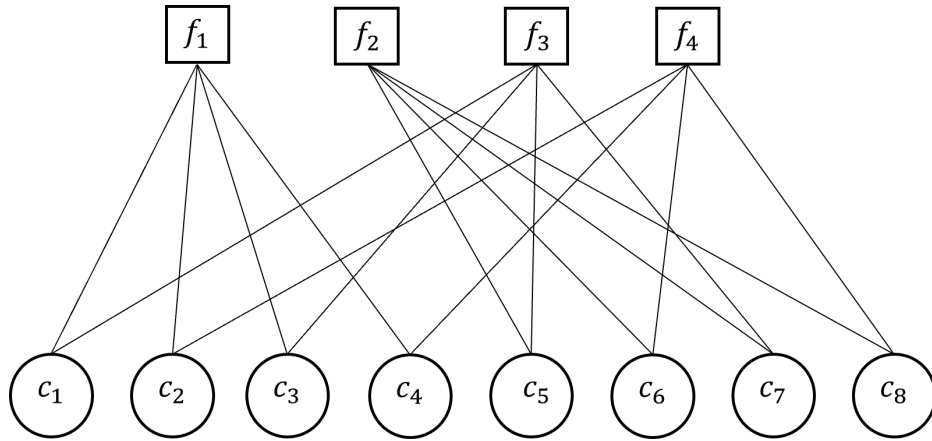
Definição 4.1.7. *Considere um grafo bipartido $D(V, A)$ com subconjuntos disjuntos de vértices $F = \{f_1, f_2, f_3, \dots, f_m\}$ e $C = \{c_1, c_2, c_3, \dots, c_n\}$. Então, este grafo bipartido pode ser representado por uma matriz de adjacência $\mathbf{H}$, cujas linhas correspondem aos vértices de F e cujas colunas correspondem aos vértices de C , onde cada elemento da matriz $h_{ij} = 1$ se, e somente se, o i -ésimo vértice de F for conectado ao j -ésimo vértice de C , caso contrário $h_{ij} = 0$.*

Exemplo 4.1.1. *A partir do grafo da Figura 20, utilizando a Definição 4.1.7, a matriz $\mathbf{H}$ desse grafo pode ser obtida. Para isso, em primeiro plano, iremos analisar o primeiro nó de função f_1 que está*

conectado aos nós de variável c_1, c_2, c_3, c_4 , logo a primeira linha de $\mathbf{H}$ é 11110000. Aplicando o mesmo procedimento para os outros nós de função, obtém-se a matriz

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

Figura 20 – Grafo bipartido.



fonte: Produção do próprio autor.

Por serem códigos de bloco linear, os códigos LDPC são definidos pelos parâmetros (n, k) , em que uma mensagem de k bits é transformada em uma palavra-código de n bits. No entanto, a codificação raramente é realizada por meio de uma matriz geradora $\mathbf{G}$, pois isso exigiria converter a matriz de verificação de paridade $\mathbf{H}$ para a forma sistemática, o que poderia comprometer uma de suas principais características: a baixa densidade de $\mathbf{H}$. Assim, a codificação é geralmente realizada a partir das equações de paridade obtidas das linhas de $\mathbf{H}$, em que a soma binária dos n bits associados a cada linha seja igual a zero. Um exemplo da obtenção das equações de paridade de uma matriz $\mathbf{H}$ e da codificação pode ser observada no Exemplo 4.1.2.

Exemplo 4.1.2. Um código LDPC com parâmetros $(n, k) = (12, 6)$ em que u_k correspondem aos bits da palavra da mensagem a ser codificada e x_{n-k} correspondem aos bits de paridade tem a matriz controle de paridade

$$\mathbf{H} = \left(\begin{array}{c|cccccccccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \\ \hline f_1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ f_2 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ f_3 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ f_4 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ f_5 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ f_6 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \right).$$

Pode-se observar que a primeira linha da matriz possui 1's nas posições correspondentes aos bits u_1 , u_3 , u_5 e x_7 , logo a equação de paridade dessa linha é dada por

$$u_1 \oplus u_3 \oplus u_5 \oplus x_7 = 0.$$

As demais equações de paridade podem ser obtidas da mesma forma da primeira equação de paridade e são dadas por

$$u_1 \oplus u_4 \oplus u_6 \oplus x_7 \oplus x_8 = 0,$$

$$u_2 \oplus u_3 \oplus u_6 \oplus x_8 \oplus x_9 = 0,$$

$$u_1 \oplus u_3 \oplus u_4 \oplus x_9 \oplus x_{10} = 0,$$

$$u_2 \oplus u_5 \oplus u_6 \oplus x_{10} \oplus x_{11} = 0,$$

$$u_2 \oplus u_4 \oplus u_5 \oplus x_{11} \oplus x_{12} = 0.$$

Agora, considere a mensagem $\mathbf{u} = (101110)$, com os bits de mensagem $u_1 = 1$, $u_2 = 0$, $u_3 = 1$, $u_4 = 1$, $u_5 = 1$, $u_6 = 0$. Ao substituírmos os bits de mensagem nas equações de paridade, é possível obter as equações de paridade

$$1 \oplus 1 \oplus 1 \oplus x_7 = 0,$$

$$1 \oplus 1 \oplus 0 \oplus x_7 \oplus x_8 = 0,$$

$$0 \oplus 1 \oplus 0 \oplus x_8 \oplus x_9 = 0,$$

$$1 \oplus 1 \oplus 1 \oplus x_9 \oplus x_{10} = 0,$$

$$0 \oplus 1 \oplus 0 \oplus x_{10} \oplus x_{11} = 0,$$

$$0 \oplus 1 \oplus 1 \oplus x_{11} \oplus x_{12} = 0.$$

Pode-se notar que para os bits de paridade $x_7 = 1$, $x_8 = 1$, $x_9 = 0$, $x_{10} = 1$, $x_{11} = 0$, $x_{12} = 0$, as equações de paridade são satisfeitas, logo a palavra-código é $\mathbf{v} = (101110110100)$.

Um parâmetro que influencia o desempenho de códigos LDPC, no contexto dos grafos de Tanner, é o comprimento dos ciclos que está relacionado com o conceito de *girth*.

Definição 4.1.8. Um caminho em um grafo é definido como uma sequência alternada de vértices e arestas, tal que cada aresta é incidente com os vértices que a precedem e a seguem, e nenhum vértice aparece mais de uma vez.

Definição 4.1.9. Um caminho fechado em um grafo que começa e termina no mesmo vértice é chamado de ciclo. O número de arestas no ciclo é chamado de comprimento do ciclo.

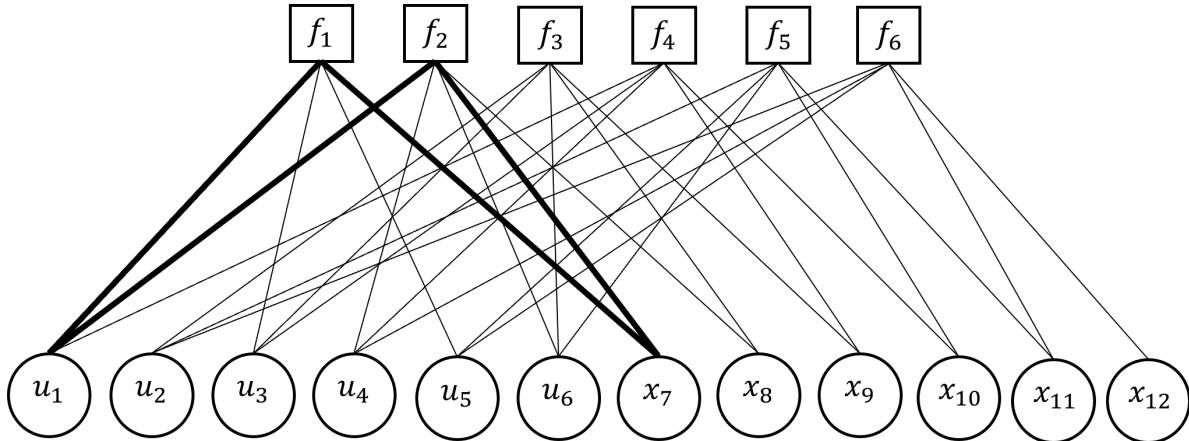
Definição 4.1.10. O Girth de um grafo é definido como o comprimento de seu ciclo mais curto.

Em um grafo de Tanner, o menor comprimento possível de um ciclo é 4, e todos os ciclos devem ter comprimento par. O ciclo de comprimento 4 é indesejável, pois compromete o desempenho do

código LDPC. Para evitá-lo, uma regra comum de construção da matriz $\mathbf{H}$ é garantir que quaisquer duas colunas tenham, no máximo, um único 1 em comum.

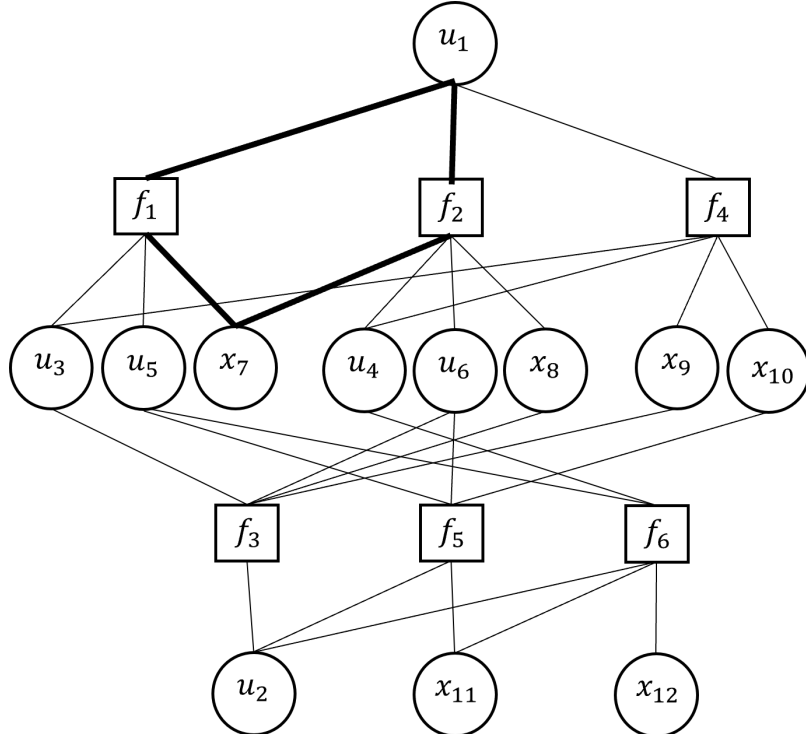
Exemplo 4.1.3. O grafo de Tanner da matriz $\mathbf{H}$ do Exemplo 4.1.2 pode ser observado na Figura 21, e a partir dele podemos construir a árvore da Figura 22. Observamos em ambas as figuras que o Girth desse grafo é 4.

Figura 21 – Grafo de Tanner de um código LDPC $(n, k) = (12, 6)$.



fonte: Produção do próprio autor.

Figura 22 – Árvore do grafo de Tanner de um código LDPC $(n, k) = (12, 6)$.



fonte: Produção do próprio autor.

Os códigos LDPC podem ser classificados em códigos LDPC regulares e códigos LDPC irregulares.

Definição 4.1.11. *Um código LDPC regular é definido como um código LDPC com parâmetros (n, w_c, w_r) , onde cada coluna contém w_c 1's e w_c é o mesmo para todas as colunas e cada linha contém w_r 1's e w_r é o mesmo para todas as linhas.*

Os códigos LDPC regulares, para garantir uma matriz esparsa, são construídos de forma que $w_c \ll m = n - k$, $w_r \ll n$ e que $w_r \geq w_c$. Além disso,

$$w_c n = w_r m. \quad (4.1.1)$$

Considerando a Equação 4.1.1 a taxa dos códigos LDPC regulares pode ser dada por

$$R = \frac{k}{n} = 1 - \frac{w_c}{w_r}, \quad (4.1.2)$$

e, a densidade por

$$d = \frac{w_r}{n} = \frac{w_c}{m}. \quad (4.1.3)$$

Um exemplo de construção de uma matriz $\mathbf{H}$ de um código LDPC regular é o método proposto por Gallager em (GALLAGER, 1963). O método de construção de Gallager é detalhado em (RYAN; LIN, 2009), e no Exemplo 4.1.1, a matriz $\mathbf{H}$ exemplifica esse método.

Richardson e Urbanke demonstraram no trabalho (RICHARDSON; SHOKROLLAHI; URBANKE, 2001) que os códigos LDPC irregulares apresentam um desempenho superior para blocos de maior comprimento, em comparação com os códigos LDPC regulares. No entanto, a maior simplicidade de implementação em hardware dos códigos regulares faz com que eles ainda sejam aplicados na prática.

Em geral, os códigos LDPC irregulares são representados pelas distribuições de grau $(\lambda(x), \rho(x))$, as quais dependem dos graus dos nós do grafo do código LDPC. A definição de grau de um nó é dada a seguir.

Definição 4.1.12. *O grau de um nó em um grafo é o número de arestas incidentes a esse nó. No contexto dos códigos LDPC, o grau de um nó de variável em um grafo de Tanner corresponde ao peso de sua coluna na matriz controle de paridade $\mathbf{H}$, enquanto o grau de um nó de paridade corresponde ao peso de sua linha.*

A distribuição de grau dos nós de variável é dada pela equação

$$\lambda(x) = \sum_i \lambda_i x^{i-1}, \quad (4.1.4)$$

em que λ_i é a fração de arestas no grafo que está conectada a um nó de variável de grau i . E a distribuição de grau dos nós de função é dada pela equação

$$\rho(x) = \sum_i \rho_i x^{i-1}, \quad (4.1.5)$$

em que ρ_i é a fração de arestas no grafo que está conectada a um nó de paridade de grau i . A soma de todos os λ_i deve ser igual a 1, pois ela representa a probabilidade total de uma aresta aleatoriamente

selecionada estar conectada a um nó de variável de algum grau e, da mesma forma, a soma de todos os ρ_i deve ser igual a 1. A partir do Exemplo 4.1.4 é possível compreender melhor como é a obtenção das distribuições de grau a partir de um grafo.

Exemplo 4.1.4. A Figura 21 apresentada no Exemplo 4.1.3 representa um grafo de Tanner de um código LDPC irregular. O grafo possui 12 nós de variável, e desses, apenas um está conectado com apenas uma aresta, 5 estão conectados com 2 arestas e 6 estão conectados a 3 arestas. Logo, a distribuição de grau dos nós de variável desse código é dada por

$$\lambda(x) = \frac{1}{12}x^{1-1} + \frac{5}{12}x^{2-1} + \frac{6}{12}x^{3-1} = \frac{1}{12} + \frac{5}{12}x + \frac{1}{2}x^2.$$

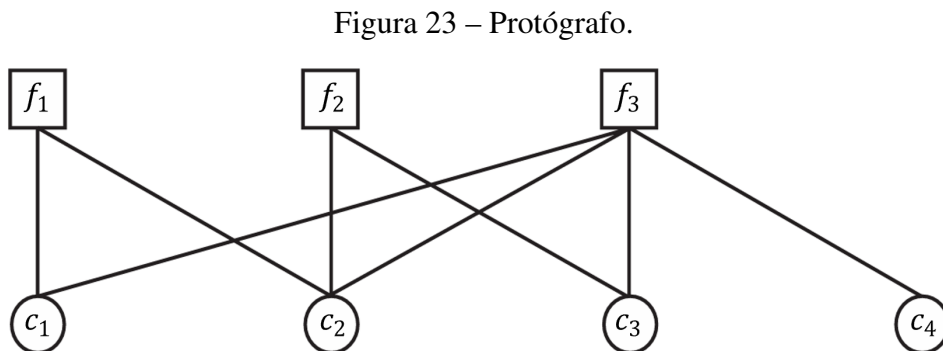
O mesmo grafo apresenta 6 nós de função, e desses, apenas um está ligado a 4 arestas e 5 estão ligados a 5 arestas, logo a distribuição de grau dos nós de função é dada por

$$\rho(x) = \frac{1}{6}x^{4-1} + \frac{5}{6}x^{5-1} = \frac{1}{6}x^3 + \frac{5}{6}x^4.$$

4.2 CÓDIGOS QC-LDPC BASEADOS EM PROTÓGRAFOS

O projeto e a análise de códigos LDPC definidos por matrizes $\mathbf{H}$ muito grandes, especialmente quando geradas aleatoriamente, podem se tornar complexos ou inviáveis, sobretudo ao tentar garantir propriedades desejadas, como a ausência de ciclos curtos. Uma solução para esse problema é o uso de protógrafos, nos quais um grafo reduzido, descrito por uma pequena matriz, é expandido para produzir uma matriz $\mathbf{H}$ de grandes dimensões. Ao concentrar-se na matriz com dimensões menores, cuja análise é mais simples devido ao tamanho reduzido, facilita a eliminação de ciclos curtos e a definição de propriedades como a distribuição de graus dos nós. Além disso, a implementação torna-se mais eficiente.

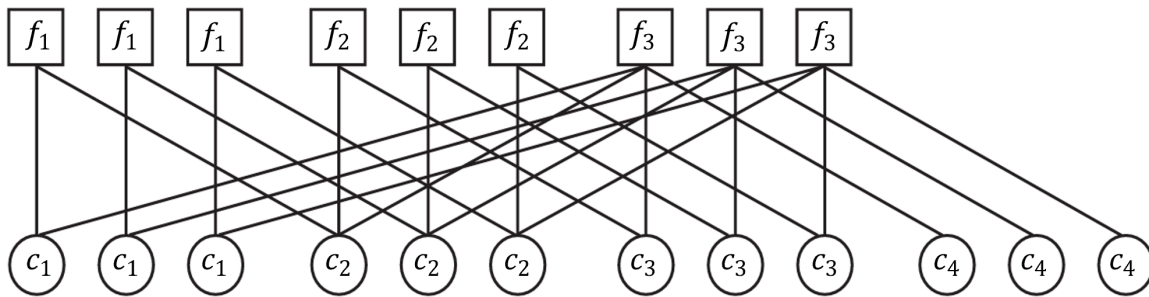
Um protógrafo é um grafo bipartido relativamente pequeno, do qual um grafo maior pode ser obtido por meio de um procedimento de cópia e permutação, que também pode ser chamado de levantamento. Nesse procedimento, o grafo é copiado Z_c vezes e, depois, as arestas das réplicas individuais são permutadas entre as Z_c réplicas para se obter um único grafo bipartido. A Figura 23 ilustra um exemplo de um protógrafo em que os nós f_1 , f_2 e f_3 são os nós de função do protógrafo e os nós c_1 , c_2 , c_3 e c_4 são os nós de variável do protógrafo.



fonte: Adaptado de (THORPE, 2003).

Na Figura 24, o protógrafo é copiado 3 vezes e as cópias estão organizadas de forma que as cópias do mesmo vértice do protógrafo estejam próximas umas das outras.

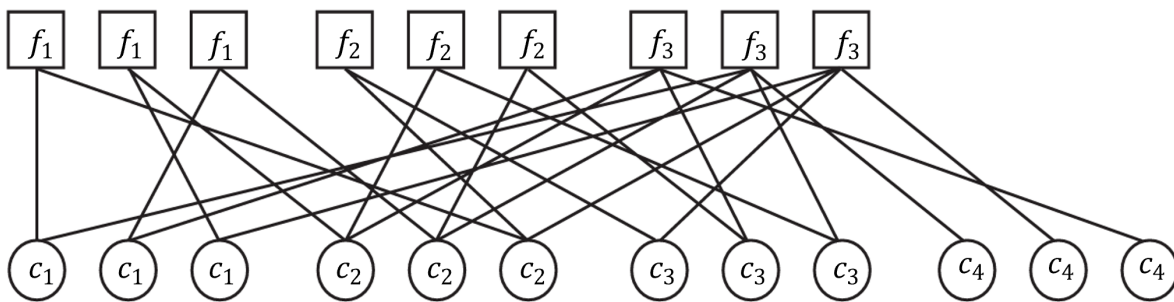
Figura 24 – Cópias do Protógrafo.



fonte: Adaptado de (THORPE, 2003).

É possível que as cópias dos protógrafos não estão conectadas, por isso é necessário permutar as arestas das réplicas individuais, o que pode ser observado na Figura 25.

Figura 25 – Grafo baseado em um protógrafo.



fonte: Adaptado de (THORPE, 2003).

Observe que, considerando que o protógrafo tem n nós de variável e m nós de função, o grafo gerado a partir do protógrafo terá Z_cn nós de variável e Z_cm nós de função. Para preservar as distribuições de grau dos nós do protógrafo original, as permutações de arestas entre as cópias não podem ser arbitrárias. Se um determinado j -ésimo nó de variável está conectado a um i -ésimo nó de função, então um nó de variável que é correspondente a uma cópia do j -ésimo nó de variável só pode se conectar a uma cópia do i -ésimo nó de função.

A seguir, definimos matriz circulante e um código quase-cíclico.

Definição 4.2.1. Uma matriz circulante é uma matriz quadrada cujas linhas são obtidas por rotações cíclicas à direita da primeira linha.

Definição 4.2.2. Um código é quase-cíclico (QC, Quasi-Cyclic) se a sua matriz de verificação de paridade $\mathbf{H}$ é um arranjo de blocos, onde cada bloco é uma matriz circulante.

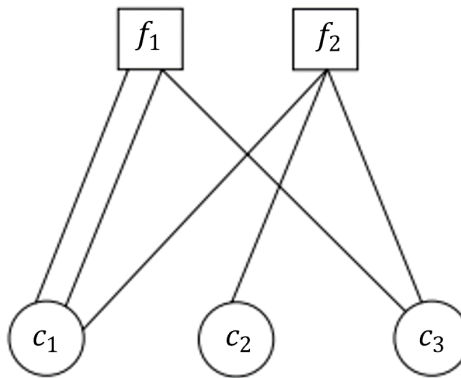
É importante citar que um código baseado em um protógrafo é quase-cíclico. Se as permutações das arestas do grafo gerado pelo protógrafo corresponderem a circulantes de peso 1, ou seja, matrizes em que cada linha é um deslocamento cíclico da linha acima dela e cada linha possui apenas um único 1, então a matriz $\mathbf{H}$ resultante do processo de cópia e permutação será um arranjo de tais matrizes

circulantes, ou seja, o código será quase-cíclico. Um exemplo de um código LDPC quase-cíclico baseado em protógrafo pode ser observado no Exemplo 4.2.1.

Exemplo 4.2.1. O protógrafo da Figura 26 tem a seguinte matriz associada

$$\mathbf{H}_P = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

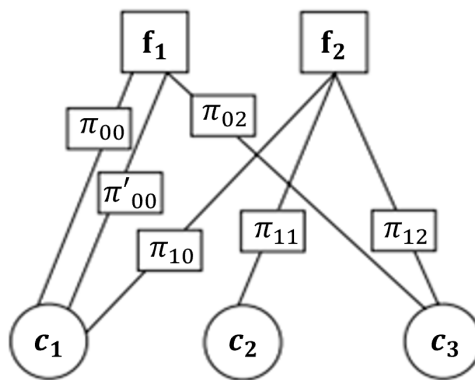
Figura 26 – Protógrafo com arestas paralelas.



fonte: Adaptado de (RYAN; LIN, 2009).

Substituindo cada nó por um conjunto de Z_c nós e cada aresta por um conjunto de Z_c arestas, é possível obter-se o grafo da Figura 27. As conexões entre os conjuntos de nós através dos conjuntos de arestas são especificadas pelas matrizes π_{ij} de ordem $Z_c \times Z_c$.

Figura 27 – Grafo obtido a partir de um protógrafo.



fonte: Adaptado de (RYAN; LIN, 2009).

Considerando um fator de levantamento $Z_c = 3$, uma possível possibilidade para as matrizes π_{ij} é

$$\pi_{00} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \pi'_{00} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \pi_{10} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

$$\pi_{11} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \pi_{02} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \pi_{12} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Assim, a matriz controle de paridade obtida por meio do protógrafo é dada por

$$\mathbf{H} = \left[\begin{array}{ccc|ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right].$$

Nessa matriz $\mathbf{H}$, é possível notar-se que π_{00} e π'_{00} foram selecionados de forma que não existam arestas paralelas no grafo correspondente à matriz $\mathbf{H}$.

No Exemplo 4.2.1, embora o primeiro bloco da matriz $\mathbf{H}$ seja uma soma de duas matrizes circulantes de peso 1, esse bloco ainda é um bloco construído a partir de matrizes circulantes. Como todos os blocos da matriz $\mathbf{H}$ são matrizes circulantes ou resultados da soma de matrizes circulantes, a matriz $\mathbf{H}$ inteira é um arranjo de matrizes circulantes, logo o código é quase-cíclico.

Embora os códigos QC-LDPC baseados em protógrafos, na teoria, permitam a utilização de arestas paralelas nos protógrafos, a implementação prática utilizada em padrões modernos, como 5G, Wi-Fi e *Digital Video Broadcasting – Satellite, 2nd Generation* (DVB-S2) adota uma convenção diferente. Nestes padrões não são utilizadas arestas paralelas, logo, um protógrafo só pode ter 0 ou 1 aresta entre os nós. Portanto, indicar o número de arestas entre os nós não é necessário, então é utilizada uma matriz protótipo $\mathbf{P}$ em que cada elemento $P(i, j)$ da matriz protótipo é substituído por um bloco de dimensões $Z_c \times Z_c$ conforme as seguintes regras.

1. Regra 1: Se $P(i, j) = -1$, é interpretado que não há conexões entre os nós correspondentes, logo esse elemento é substituído por uma matriz nula de dimensões $Z_c \times Z_c$.
2. Regra 2: Se $P(i, j) = 0$, é interpretado que há uma conexão entre os nós correspondentes e esse elemento é substituído por uma matriz identidade de ordem $Z_c \times Z_c$.
3. Regra 3: Se $P(i, j) = s$ e $s > 0$, também é interpretado que há uma conexão entre os nós correspondentes e o elemento é substituído por uma matriz identidade de ordem $Z_c \times Z_c$ com suas colunas deslocadas ciclicamente s vezes.

A matriz $\mathbf{P}$ nesta convenção funciona como um mapa de deslocamentos que define diretamente a estrutura de permutação de cada bloco na matriz $\mathbf{H}$. A geração de uma matriz $\mathbf{H}$ a partir de uma matriz $\mathbf{P}$ utilizando as regras mencionadas pode ser observada no Exemplo 4.2.2.

Exemplo 4.2.2. Considere um código QC-LDPC, baseado em um protógrafo, com taxa $R = \frac{3}{4}$ e fator de levantamento $Z_c = 3$. A matriz protótipo do código é dada por

$$\mathbf{P}_1 = \begin{pmatrix} 0 & -1 & 1 & 2 & -1 & 0 & 2 & -1 \\ -1 & 1 & 0 & -1 & 2 & 1 & -1 & 2 \end{pmatrix}.$$

Nessa matriz, os elementos $P_1(i, j) = -1$ são substituídos por matrizes nulas $\mathbf{H}_0$ de ordem 3×3 , os elementos $P_1(i, j) = 0$ são substituídos por matrizes identidade $\mathbf{H}_1$ de ordem 3×3 e os elementos $P_1(i, j) = 1$ e $P_1(i, j) = 2$ são substituídos por matrizes identidade de ordem $Z_c \times Z_c$ dadas por

$$\mathbf{H}_{\mathbf{I}_1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad e \quad \mathbf{H}_{\mathbf{I}_2} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Substituindo todos os valores dos elementos da matriz $\mathbf{P}_1$ por matrizes nulas, matrizes identidade e matrizes identidade com colunas ciclicamente deslocadas, obtêm-se a seguinte matriz

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

No padrão IEEE 802.11 (Wi-Fi), os códigos LDPC são definidos por uma matriz protótipo, também chamada de matriz base, composta por índices de deslocamento usados para gerar uma matriz $\mathbf{H}$ de maior ordem, estruturada a partir de submatrizes circulantes de tamanho fixo (IEEE Standards Association, 2020). Esse padrão fornece uma tabela com os valores de deslocamento a serem utilizados, bem como os valores de Z_c . No IEEE 802.11 podem ser utilizadas matrizes $\mathbf{P}$ com taxas de $1/2$, $2/3$, $3/4$ e $5/6$, e os valores de deslocamento são $Z_c = 27$ ou $Z_c = 54$. Um exemplo de matriz $\mathbf{P}$ utilizando um fator de levantamento $Z_c = 27$ no padrão IEEE 802.11 pode ser observado na seguinte matriz

$$\mathbf{P}_2 = \begin{bmatrix} 16 & 17 & 22 & 24 & 9 & 3 & 14 & -1 & 4 & 2 & 7 & -1 & 26 & -1 & 2 & -1 & 21 & -1 & 1 & 0 & -1 & -1 & -1 & -1 \\ 25 & 12 & 12 & 3 & 3 & 26 & 6 & 21 & -1 & 15 & 22 & -1 & 15 & -1 & 4 & -1 & -1 & 16 & -1 & 0 & 0 & -1 & -1 & -1 \\ 25 & 18 & 26 & 16 & 22 & 23 & 9 & -1 & 0 & -1 & 4 & -1 & 4 & -1 & 8 & 23 & 11 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\ 9 & 7 & 0 & 1 & 17 & -1 & -1 & 7 & 3 & -1 & 3 & 23 & -1 & 16 & -1 & -1 & 21 & -1 & 0 & -1 & -1 & 0 & 0 & -1 \\ 24 & 5 & 26 & 7 & 1 & -1 & -1 & 15 & 24 & 15 & -1 & 8 & -1 & 13 & -1 & 13 & -1 & 11 & -1 & -1 & -1 & -1 & 0 & 0 \\ 2 & 2 & 19 & 14 & 24 & 1 & 15 & 19 & -1 & 21 & -1 & 2 & -1 & 24 & -1 & 3 & -1 & 2 & 1 & -1 & -1 & -1 & -1 & 0 \end{bmatrix}. \quad (4.2.1)$$

No padrão 5G, a estrutura dos códigos QC-LDPC é um pouco diferente, pois são utilizadas duas matrizes base fixas (BG1 e BG2) e o valor de Z_c é variável (3rd Generation Partnership Project (3GPP), 2024). A matriz BG1 possui taxa $R = \frac{1}{3}$ e é empregada para *payloads* maiores e códigos mais robustos,

enquanto a matriz BG2 possui taxa $R = \frac{2}{5}$ e é utilizada para *payloads* menores e códigos com maior eficiência. Embora as taxas das matrizes BG1 e BG2 sejam fixas, o código LDPC pode assumir qualquer taxa, uma vez que a taxa final é ajustada por meio de um processo de casamento de taxa, o qual seleciona quantos bits serão transmitidos.

4.3 DECODIFICAÇÃO EM CÓDIGOS LDPC

Existem diversos algoritmos de decodificação utilizados em códigos LDPC, cada um com características e propriedades específicas. Um algoritmo muito utilizado é o SPA que apresenta um alto desempenho em termos de correção de erros, porém apresenta maior complexidade computacional devido à utilização de funções hiperbólicas. Com o objetivo de reduzir essa complexidade, o algoritmo de Mínima Soma realiza uma aproximação para evitar o cálculo das funções hiperbólicas. Além disso, para tornar a convergência dos algoritmos de decodificação mais rápida, pode-se adotar a estratégia da decodificação em camadas, na qual a matriz controle de paridade é dividida em camadas, e o processo de decodificação é aplicado de forma sequencial em cada uma delas. Os diferentes algoritmos apresentados também podem operar em sistemas com ordem de modulação maior que 2, porém são necessárias algumas modificações no algoritmo de decodificação.

Esta seção será organizada em cinco subseções. A Subseção 4.3.1 apresenta os conceitos básicos de razão logarítmica de verossimilhança (LLR, *Log-Likelihood Ratio*), fundamentais para os algoritmos discutidos neste capítulo. Em seguida, são descritos o Algoritmo Soma e Produto, na Subseção 4.3.2, e o Algoritmo Mínima Soma, na Subseção 4.3.3. A Subseção 4.3.4 trata da decodificação em camadas dos algoritmos estudados. Por fim, a Subseção 4.3.5 discute a decodificação em sistemas com ordem de modulação superior a 2.

4.3.1 Razão Logarítmica de Verossimilhança (LLR)

Um conceito amplamente utilizado em algoritmos de decodificação *soft-decision* é a razão logarítmica de verossimilhança (LLR). Entre os algoritmos que fazem uso desse conceito, destaca-se o algoritmo SPA, o qual emprega os valores de LLR para realizar o processo de decodificação. A LLR baseia-se em estimativas de probabilidade de cada um dos bits recebidos e também na relação que cada bit possui com os outros.

O conceito de razão logarítmica de verossimilhança (LLR) será demonstrado considerando uma mensagem binária qualquer, que será modulada utilizando a técnica de modulação BPSK, onde um bit 0 corresponde a uma amplitude de +1 e um bit 1 corresponde a uma amplitude de -1. Também será considerado que a mensagem modulada passará por um canal Gaussiano e o vetor de valores recebidos no decodificador é $\mathbf{r} = (r_1, r_2, r_3, \dots, r_n)$.

Na decodificação *soft-decision*, o decodificador transforma o vetor recebido do canal $\mathbf{r}$ em um conjunto de valores de estimativas de probabilidade $\mathbf{L} = [L_1, L_2, L_3, \dots, L_n]$. Cada um dos valores L_i do conjunto $\mathbf{L}$ pode ser dado como a soma das estimativas extrínsecas l_i^e e intrínsecas l_i^i , como pode ser observado na equação

$$L_i = l_i^e + l_i^i. \quad (4.3.1)$$

As estimativas intrínsecas l_i^i estão relacionadas com o valor real recebido do i -ésimo símbolo a ser analisado, e as estimativas extrínsecas l_i^e estão relacionadas a como os outros símbolos afetam o i -ésimo símbolo a ser analisado.

Considerando o Teorema de Bayes, é possível calcular a probabilidade de um bit enviado ser 0 dado o valor do bit recebido no decodificador, a qual é dada pela equação

$$P_r(c_i = 0|r_i) = \frac{f(r_i|c_i = 0)P_r(c_i = 0)}{f(r_i)}, \quad (4.3.2)$$

em que $f(r_i|c_i = 0)$ é a função densidade de probabilidade condicional de r_i dado que $c_i = 0$, $P_r(c_i = 0)$ é a probabilidade de $c_i = 0$ e $f(r_i)$ é a função densidade de probabilidade de r_i . De forma análoga, é possível calcular a probabilidade de um bit enviado ser 1 dado o valor do bit recebido no decodificador, a qual é dada por

$$P_r(c_i = 1|r_i) = \frac{f(r_i|c_i = 1)P_r(c_i = 1)}{f(r_i)}, \quad (4.3.3)$$

em que $f(r_i|c_i = 1)$ é a função densidade de probabilidade condicional de r_i dado que $c_i = 1$ e $P_r(c_i = 1)$ é a probabilidade de $c_i = 1$.

A razão de verossimilhança intrínseca é dada pela divisão de $P_r(c_i = 0|r_i)$ por $P_r(c_i = 1|r_i)$, como pode ser observado em

$$l_i^i = \frac{P_r(c_i = 0 | r_i)}{P_r(c_i = 1 | r_i)} = \frac{\frac{f(r_i | c_i = 0)^{\frac{1}{2}}}{f(r_i)}}{\frac{f(r_i | c_i = 1)^{\frac{1}{2}}}{f(r_i)}} = \frac{f(r_i | c_i = 0)}{f(r_i | c_i = 1)}. \quad (4.3.4)$$

Na Equação 4.3.4 também é considerado que os bits são equiprováveis, logo, $P_r(c_i = 0) = P_r(c_i = 1) = \frac{1}{2}$. A equação da razão de verossimilhança intrínseca também pode ser dada na forma logarítmica, como pode ser observado na equação

$$l_i^i = \ln \left(\frac{P_r(c_i = 0 | r_i)}{P_r(c_i = 1 | r_i)} \right) = \ln \left(\frac{f(r_i | c_i = 0)}{f(r_i | c_i = 1)} \right). \quad (4.3.5)$$

Considerando que o bit c_i é modulado utilizando a técnica BPSK e passa por um canal com ruído AWGN, caso o bit c_i que sai do codificador for igual a 0, o valor r_i que será recebido no decodificador será $1 + N(0, \sigma^2)$, onde $N(0, \sigma^2)$ é o ruído adicionado pelo canal e segue uma distribuição normal com média 0 e variância σ^2 , o que pode ser observado na equação

$$c_i = 0 \Rightarrow r_i = +1 + N(0, \sigma^2). \quad (4.3.6)$$

De maneira análoga, caso o bit c_i que sai do codificador for igual a 1, o valor r_i que será recebido no decodificador será $-1 + N(0, \sigma^2)$, como pode ser observado na equação

$$c_i = 1 \Rightarrow r_i = -1 + N(0, \sigma^2), \quad (4.3.7)$$

A partir das Equações 4.3.6 e 4.3.7, é possível obter as funções densidade de probabilidade condicionais $f(r_i | c_i = 0)$ e $f(r_i | c_i = 1)$ em função de r_i e de σ , as quais são dadas pelas seguintes equações

$$c_i = 0 \Rightarrow r_i = +1 + N(0, \sigma^2) \Rightarrow f(r_i | c_i = 0) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_i - 1)^2}{2\sigma^2}\right), \quad (4.3.8)$$

$$c_i = 1 \Rightarrow r_i = -1 + N(0, \sigma^2) \Rightarrow f(r_i | c_i = 0) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_i + 1)^2}{2\sigma^2}\right). \quad (4.3.9)$$

Substituindo as Equações 4.3.8 e 4.3.9 na Equação 4.3.5, é possível chegar na equação da LLR intrínseca em função de r_i e σ , que é dada por

$$l_i^i = \frac{2}{\sigma^2} r_i \quad (4.3.10)$$

A partir dessa equação, observa-se que a LLR intrínseca depende do valor recebido no decodificador r_i e de uma constante $\frac{2}{\sigma^2}$ que depende da variância do ruído do canal. Como essa constante é positiva, ela é frequentemente ignorada nos algoritmos de decodificação, pois atua apenas como um fator de escala que não altera a decisão final do sinal.

A LLR extrínseca é mais complexa, pois não depende apenas do bit a ser analisado, mas da influência dos demais bits no bit em questão. A LLR extrínseca será descrita a seguir utilizando um código de verificação de paridade única (SPC, *Single Parity Check*), que é um dos tipos mais simples de código de bloco linear.

Considerando que o código SPC tem parâmetros $n = 3$ e $k = 2$, a mensagem a ser codificada será $\mathbf{m} = (m_1, m_2)$ e a palavra-código será $\mathbf{c} = (m_1, m_2, \rho)$, em que o bit de verificação de paridade ρ pode ser dado por

$$\rho = m_1 \oplus m_2. \quad (4.3.11)$$

Assim, as palavras-códigos serão: {000, 011, 101, 110}.

Para os cálculos das LLRs, será considerado que a palavra-código será modulada utilizando BPSK e que será transmitida por um canal Gaussiano. Seja $\mathbf{r} = (r_1, r_2, r_3)$ a palavra recebida no decodificador, a qual será transformada pelo decodificador em um vetor com as LLRs $\mathbf{L} = (L_1, L_2, L_3)$, em que cada um dos valores de LLR, L_i , pode ser dado como a soma das LLRs extrínseca l_i^e e intrínseca l_i^i .

Para a simplificação dos cálculos, a LLR será calculada somente para o bit c_1 e depois será generalizada para os demais bits e para códigos com mais de um bit de verificação de paridade. Também para a simplificação dos cálculos, será considerado que $P_r(c_i = 0 | r_i) = p_i$ e $P_r(c_i = 1 | r_i) = 1 - p_i$.

Se $c_1 = 0$, as únicas palavras-código possíveis são $\mathbf{c} = (000)$ e $\mathbf{c} = (011)$. Então, se $c_1 = 0$, $c_2 = c_3 = 0$ ou $c_2 = c_3 = 1$, e a partir disso pode-se obter $P_r(c_1 = 0 | r_2, r_3)$, que é em função dos outros bits (e não do bit c_1), e que pode ser observada na equação

$$P_r(c_1 = 0 | r_2, r_3) = p_1 = p_2 p_3 + (1 - p_2)(1 - p_3). \quad (4.3.12)$$

De maneira análoga, se $c_1 = 1$, as únicas palavras-código possíveis são $\mathbf{c} = (101)$ e $\mathbf{c} = (110)$. Logo, $c_2 = 0$ e $c_3 = 1$ ou $c_2 = 1$ e $c_3 = 0$. Assim,

$$P_r(c_1 = 1|r_2, r_3) = 1 - p_1 = p_2(1 - p_3) + p_3(1 - p_2). \quad (4.3.13)$$

Subtraindo $P_r(c_1 = 0|r_2, r_3)$ por $P_r(c_1 = 1|r_2, r_3)$, utilizando a Equação 4.3.12 e a Equação 4.3.13, temos

$$P_r(c_1 = 0|r_2, r_3) - P_r(c_1 = 1|r_2, r_3) = p_1 - (1 - p_2) = (p_2 - (1 - p_2))(p_3 - (1 - p_3)). \quad (4.3.14)$$

De maneira análoga, somando $P_r(c_1 = 0|r_2, r_3)$ com $P_r(c_1 = 1|r_2, r_3)$, segue que

$$P_r(c_1 = 0|r_2, r_3) + P_r(c_1 = 1|r_2, r_3) = p_1 + (1 - p_1) = (p_2 + (1 - p_2))(p_3 + (1 - p_3)). \quad (4.3.15)$$

Dividindo $p_1 - (1 - p_1)$ por $p_1 + (1 - p_1)$, utilizando as Equações 4.3.14 e 4.3.15, e simplificando, obtemos

$$\frac{1 - \frac{1-p_1}{p_1}}{1 + \frac{1-p_1}{p_1}} = \frac{1 - \frac{1-p_2}{p_2}}{1 + \frac{1-p_2}{p_2}} \frac{1 - \frac{1-p_3}{p_3}}{1 + \frac{1-p_3}{p_3}}. \quad (4.3.16)$$

De acordo com a Equação 4.3.4, a LLR intrínseca depende somente do i -ésimo bit. Já a LLR extrínseca analisa como os outros símbolos afetam o i -ésimo símbolo, logo depende dos outros bits recebidos que não são o i -ésimo bit. Assim,

$$l_1^e = \ln \left(\frac{P_r(c_1 = 0 | r_2, r_3)}{P_r(c_1 = 1 | r_2, r_3)} \right) = \ln \left(\frac{p_1}{1 - p_1} \right). \quad (4.3.17)$$

Considerando que $e^{-\ln(x)} = \frac{1}{x}$, a partir da Equação 4.3.17, segue que

$$e^{-l_i^e} = e^{-\ln \frac{p_i}{1-p_i}} = \frac{1 - p_i}{p_i}. \quad (4.3.18)$$

E substituindo na Equação 4.3.16 temos

$$\frac{1 - e^{-l_1^e}}{1 + e^{-l_1^e}} = \frac{1 - e^{-l_2^e}}{1 + e^{-l_2^e}} \frac{1 - e^{-l_3^e}}{1 + e^{-l_3^e}}. \quad (4.3.19)$$

Como a tangente hiperbólica de $\frac{x}{2}$ pode ser dada por

$$\tanh \left(\frac{x}{2} \right) = \frac{1 - e^{-x}}{1 + e^{-x}}, \quad (4.3.20)$$

a Equação 4.3.19 pode ser reescrita como:

$$\tanh \left(\frac{l_1^e}{2} \right) = \tanh \left(\frac{l_2^e}{2} \right) \tanh \left(\frac{l_3^e}{2} \right). \quad (4.3.21)$$

O valor de l_1^e pode ser obtido isolando esse termo na Equação 4.3.21 e é dado por

$$l_1^e = 2 \tanh^{-1} \left(\tanh \left(\frac{l_2^i}{2} \right) \tanh \left(\frac{l_3^i}{2} \right) \right). \quad (4.3.22)$$

A Equação 4.3.22 é numericamente desafiadora devido à presença do produto e das funções $\tanh$ e $\tanh^{-1}$. Como a tangente hiperbólica é uma função ímpar, podemos contornar a complexidade numérica do cálculo dividindo-a em duas partes: a de sinal e a de magnitude. A parte de sinal é dada por

$$\text{sign}(l_1^e) = \text{sign}(l_2^i) \text{sign}(l_3^i), \quad (4.3.23)$$

e a parte da magnitude pode ser obtida aplicando o módulo em ambos os lados

$$|l_1^e| = 2 \tanh^{-1} \left| \tanh \left(\frac{|l_2^i|}{2} \right) \right| \left| \tanh \left(\frac{|l_3^i|}{2} \right) \right|. \quad (4.3.24)$$

Unindo as Equações 4.3.23 e 4.3.24, obtemos a equação

$$l_1^e = \text{sign}(l_2^i) \text{sign}(l_3^i) \times 2 \tanh^{-1} \left| \tanh \left(\frac{|l_2^i|}{2} \right) \right| \left| \tanh \left(\frac{|l_3^i|}{2} \right) \right|. \quad (4.3.25)$$

A equação da LLR extrínseca também pode ser observada no formato logarítmico, na forma

$$l_1^e = \text{sign}(l_2^i) \text{sign}(l_3^i) 2 \tanh^{-1} \left[\exp \left(\ln \left| \tanh \left(\frac{|l_2^i|}{2} \right) \right| + \ln \left| \tanh \left(\frac{|l_3^i|}{2} \right) \right| \right) \right]. \quad (4.3.26)$$

Vale ressaltar que a LLR extrínseca das equações 4.3.26 e 4.3.25 foram dadas para um código SPC (3, 2). Porém, elas podem ser generalizadas para códigos de bloco maiores e com mais de um bit de paridade, o que será demonstrado na próxima subseção.

4.3.2 Algoritmo Soma e Produto (SPA)

Gallager, em seu trabalho publicado em 1963 (GALLAGER, 1963), também forneceu um algoritmo de decodificação quase ideal, que agora é chamado de Algoritmo Soma e Produto (SPA). Esse algoritmo também pode ser chamado de Algoritmo de Propagação de Crenças (BPA, *Belief Propagation Algorithm*). Esse algoritmo irá utilizar os mesmos conceitos de LLR vistos, porém aplicados a um código LDPC.

Nesse algoritmo, será considerado que o grafo de Tanner do código será composto por nós de variável c_n e nós de função f_m . Também será considerado que o conjunto de nós de variável conectados a um nó de função f_m é $C(f_m)$, e o conjunto de nós de função conectados a um nó de variável c_n é $F(c_n)$. O conjunto de nós de variável conectados a um nó de função f_m , com exceção do nó de variável c_n , é dado por $C(f_m) \setminus c_n$, e o conjunto de nós de função conectados a um nó de variável c_n , com exceção do nó de função f_m , é dado por $F(c_n) \setminus f_m$.

O algoritmo pode realizar uma ou mais iterações até atingir um critério de parada pré-estabelecido. Em geral, esse critério de parada é um número máximo de iterações ou se a palavra encontrada na iteração for uma palavra-código. Com exceção da primeira iteração, que apresenta um passo de

inicialização, cada iteração apresenta um passo horizontal, um passo vertical e um passo de decisão, que são demonstrados a seguir.

1. **Passo de Inicialização:** Esse passo ocorre nos nós de variável e é realizado somente na primeira iteração, pois são recebidos somente os valores do vetor $\mathbf{y} = (y_1, y_2, \dots, y_n)$ recebido na decodificação. O valor inicial da LLR na primeira iteração, L_1 , é igual ao vetor $\mathbf{y}$ e esses valores são as primeiras informações $q_{c_n \rightarrow f_m}$ que serão enviadas dos nós de variável para os nós de função.
2. **Passo Horizontal:** Esse passo ocorre nos nós de função. Nesse passo, as mensagens $q_{c_n \rightarrow f_m}$, do passo anterior, são utilizadas para calcular as informações que serão enviadas dos nós de função para os nós de variável, $r_{f_m \rightarrow c_n}$, utilizando a equação

$$r_{f_m \rightarrow c_n} = \prod_{c'_n \in C(f_m) \setminus c_n} \text{sign}(q_{c'_n \rightarrow f_m}) \times 2 \tanh^{-1} \prod_{c'_n \in C(f_m) \setminus c_n} \tanh\left(\frac{|q_{c'_n \rightarrow f_m}|}{2}\right). \quad (4.3.27)$$

A Equação 4.3.27 também pode ser dada no formato logarítmico, como pode ser observado em

$$r_{f_m \rightarrow c_n} = \prod_{c'_n \in C(f_m) \setminus c_n} \text{sign}(q_{c'_n \rightarrow f_m}) \times 2 \tanh^{-1} \left[\exp\left(\sum_{c'_n \in C(f_m) \setminus c_n} \left| \ln\left(\tanh\left(\frac{|q_{c'_n \rightarrow f_m}|}{2} \right) \right) \right| \right) \right]. \quad (4.3.28)$$

3. **Passo Vertical:** Esse passo ocorre nos nós de variável. Nesse passo, as mensagens $r_{f_m \rightarrow c_n}$, do passo anterior, são utilizadas em conjunto com a última LLR calculada L_n para calcular as informações que serão enviadas dos nós de variável para os nós de função, $q_{c_n \rightarrow f_m}$, a partir da equação

$$q_{c_n \rightarrow f_m} = L_n + \sum_{f'_m \in F(c_n) \setminus f_m} r_{f'_m \rightarrow c_n}. \quad (4.3.29)$$

4. **Passo de Decisão:** A partir dos valores de $r_{f_m \rightarrow c_n}$ e da última LLR calculada L_n , um novo vetor de LLRs, L_{n+1} , é calculado utilizando a equação

$$L_{n+1} = L_n + \sum_{f_m \in F(c_n)} r_{f_m \rightarrow c_n}. \quad (4.3.30)$$

Considerando uma modulação BPSK, a palavra encontrada na iteração é dada pela LLR L_{n+1} . Se $L_{n+1,n} \geq 0$, é atribuído o bit 0 para essa LLR, caso contrário, é atribuído o bit 1, como pode ser observado na equação

$$\begin{cases} L_{n+1,n} \geq 0 \Rightarrow c_n = 0 \\ L_{n+1,n} < 0 \Rightarrow c_n = 1 \end{cases}. \quad (4.3.31)$$

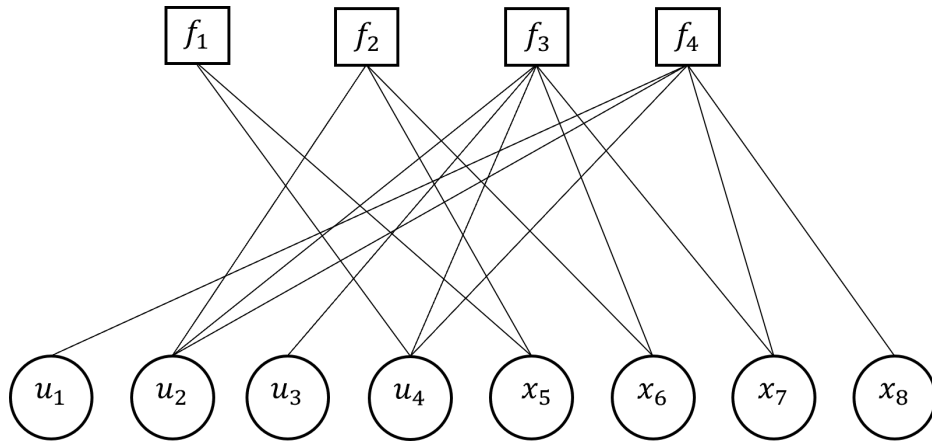
Se a palavra encontrada nesse passo pertencer ao código, essa palavra será a palavra-código resultante da decodificação e o algoritmo para, caso contrário, é iniciada uma nova iteração, porém sem o primeiro passo. O algoritmo também para caso um limite máximo de iterações seja alcançado.

O algoritmo SPA pode parecer um tanto complexo. Devido a isso, e para um melhor entendimento desse algoritmo, ele será exemplificado com um código LDPC com matriz de paridade pequena.

Exemplo 4.3.1. *Seja um código LDPC com parâmetros $(n, k) = (8, 4)$, com o grafo da Figura 28 e com a matriz*

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

Figura 28 – Grafo de um código LDPC $(8, 4)$.



fonte: Produção do próprio autor.

Considerando que a palavra-código $\mathbf{c} = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]$ é modulada por um BPSK, passará por um canal Gaussiano e a palavra recebida no decodificador é

$$\mathbf{y} = [-1,03 \quad +1,20 \quad -1,05 \quad +0,88 \quad +1,08 \quad +0,91 \quad -1,18 \quad +1,04].$$

No passo de inicialização, o valor inicial da LLR na primeira iteração, $\mathbf{L}_1$, é igual ao vetor $\mathbf{y}$, logo $\mathbf{L}_1 = [-1,03 \quad +1,20 \quad -1,05 \quad +0,88 \quad +1,08 \quad +0,91 \quad -1,18 \quad +1,04]$. E esses valores são as primeiras informações $q_{c_n \rightarrow f_m}$ que serão enviadas dos nós de variável para os nós de função. Os valores de $q_{c_n \rightarrow f_m}$ podem ser visualizados na seguinte matriz $\mathbf{Q}$:

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 & +0,88 & +1,08 & 0 & 0 & 0 \\ 0 & +1,20 & 0 & 0 & +1,08 & +0,91 & 0 & 0 \\ 0 & +1,20 & -1,05 & +0,88 & 0 & +0,91 & -1,18 & 0 \\ -1,03 & +1,20 & 0 & +0,88 & 0 & 0 & -1,18 & +1,04 \end{pmatrix}.$$

No passo horizontal da primeira iteração, são utilizados os dados da última matriz $\mathbf{Q}$ calculada para calcular as informações que serão enviadas dos nós de função para os nós de variável $r_{f_m \rightarrow c_n}$, utilizando a Equação 4.3.27. O cálculo de $r_{1 \rightarrow 4}$, $r_{1 \rightarrow 5}$ e $r_{2 \rightarrow 2}$ pode ser dado por

$$r_{1 \rightarrow 4} = (+) \times 2 \tanh^{-1} \left(\tanh \left(\frac{|+1,08|}{2} \right) \right) = +1,08,$$

$$r_{1 \rightarrow 5} = (+) \times 2 \tanh^{-1} \left(\tanh \left(\frac{|+0,88|}{2} \right) \right) = +0,88,$$

$$r_{2 \rightarrow 2} = (+)(+) \times 2 \tanh^{-1} \left(\tanh \left(\frac{|+1,08|}{2} \right) \tanh \left(\frac{+0,91}{2} \right) \right) = +0,43.$$

Os demais valores de $r_{f_m \rightarrow c_n}$ podem ser calculados da mesma forma. Todos os valores de $r_{f_m \rightarrow c_n}$ podem ser observados na matriz

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 & +1,08 & +0,88 & 0 & 0 & 0 \\ 0 & +0,43 & 0 & 0 & +0,47 & +0,54 & 0 & 0 \\ 0 & +0,09 & -0,10 & +0,12 & 0 & +0,11 & -0,09 & 0 \\ -0,11 & +0,10 & 0 & +0,13 & 0 & 0 & -0,10 & +0,11 \end{pmatrix}.$$

No passo vertical da primeira iteração, são utilizados os últimos valores de $r_{f_m \rightarrow c_n}$ para calcular os valores que serão enviados para os nós de função, $q_{c_n \rightarrow f_m}$, a partir da Equação 4.3.29. O cálculo de $q_{4 \rightarrow 1}$, $q_{5 \rightarrow 1}$ e $q_{2 \rightarrow 2}$ é dado por

$$q_{4 \rightarrow 1} = 0,12 + 0,13 + 0,88 = +1,13,$$

$$q_{5 \rightarrow 1} = 0,47 + 1,08 = +1,55,$$

$$q_{2 \rightarrow 2} = 0,09 + 0,10 + 1,20 = +1,39.$$

Os demais valores de $q_{c_n \rightarrow f_m}$ podem ser calculados da mesma forma. Todos os valores de $q_{c_n \rightarrow f_m}$ calculados nesse passo são apresentados na seguinte matriz $\mathbf{Q}$:

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 & +1,13 & +1,55 & 0 & 0 & 0 \\ 0 & +1,39 & 0 & 0 & +1,96 & +1,02 & 0 & 0 \\ 0 & +1,73 & -1,05 & +2,19 & 0 & +1,45 & -1,28 & 0 \\ -1,03 & +1,72 & 0 & +2,18 & 0 & 0 & -1,27 & +1,04 \end{pmatrix}.$$

A partir dos valores de $r_{f_m \rightarrow c_n}$ dessa iteração e das LLRs de $\mathbf{L}_1$, utilizando a 4.3.30, é calculada as novas LLRs $L_{2,n}$. O cálculo de $L_{2,1}$, $L_{2,2}$ e $L_{2,3}$ é dado por

$$L_{2,1} = -0,11 - 1,03 = -1,14,$$

$$L_{2,2} = 0,43 + 0,09 + 0,10 + 1,20 = +1,81,$$

$$L_{2,3} = -0,10 - 1,05 = -1,15.$$

As demais LLRs podem ser calculadas da mesma forma e o vetor com as novas LLRs, $\mathbf{L}_2$, é

$$\mathbf{L}_2 = \begin{bmatrix} -1,14 & +1,81 & -1,15 & +2,21 & +2,43 & +1,56 & -1,37 & +1,15 \end{bmatrix}.$$

Utilizando a regra de decisão da Equação 4.3.31, a seguinte palavra pode ser obtida:

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

Como a palavra encontrada satisfaz $\mathbf{c} \cdot \mathbf{H}^T = 0$, essa palavra é a palavra-código encontrada e o algoritmo para.

4.3.3 Algoritmo Mínima Soma (MS)

Como discutido anteriormente, o algoritmo SPA utiliza a função tangente hiperbólica em seu processo de atualização de mensagens. Entretanto, o cálculo dessas funções introduz uma complexidade computacional significativa no decodificador, especialmente para códigos LDPC de maiores dimensões. Diante desse cenário, diferentes aproximações foram propostas com o intuito de reduzir o custo computacional do algoritmo SPA. Uma das aproximações consiste em substituir o cálculo das tangentes hiperbólicas por aproximações mais simples. Contudo, essa abordagem implica um comprometimento da qualidade de decodificação, resultando em um desempenho inferior em termos de taxa de erro de bit (BER) (TU; ZHANG, 2007; EMRAN; ELSABROUTY, 2014).

A partir do gráfico da Figura 29, é possível observar que, quando $|x| \rightarrow 0$, tem-se $f(x) \rightarrow \infty$, enquanto, quando $|x| \rightarrow \infty$, tem-se $f(x) \rightarrow 0$. Dessa forma, quanto menor o valor de $|q_{c'_n \rightarrow f_m}|$ na Equação 4.3.28, maior será o valor de $\left| \ln \left(\tanh \frac{|q_{c'_n \rightarrow f_m}|}{2} \right) \right|$. Isso indica que o termo dominante no somatório da Equação 4.3.28 corresponde ao menor valor de $|q_{c'_n \rightarrow f_m}|$. O Algoritmo Mínima Soma considera apenas esse menor valor na Equação 4.3.28, ou seja,

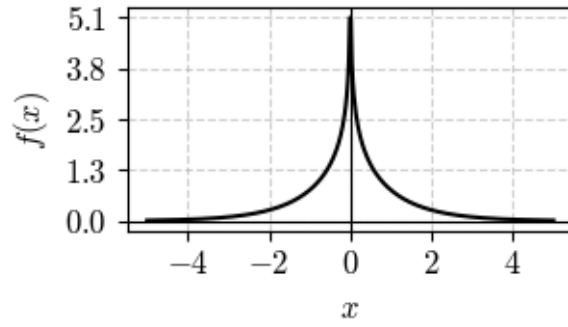
$$r_{f_m \rightarrow c_n} = \prod_{c'_n \in C(f_m) \setminus c_n} \text{sign}(q_{c'_n \rightarrow f_m}) \times 2 \tanh^{-1} \left[\exp \left(\sum_{c'_n \in C(f_m) \setminus c_n} \left| \ln \left(\tanh \left(\frac{\min(|q_{c'_n \rightarrow f_m}|)}{2} \right) \right) \right| \right) \right]. \quad (4.3.32)$$

A partir das devidas manipulações realizadas na Equação 4.3.32, obtém-se a equação

$$r_{f_m \rightarrow c_n} = \prod_{c'_n \in C(f_m) \setminus c_n} \text{sign}(q_{c'_n \rightarrow f_m}) \times \min_{c'_n \in C(f_m) \setminus c_n} (|q_{c'_n \rightarrow f_m}|). \quad (4.3.33)$$

É possível notar que a única diferença entre o algoritmo de mínima soma e o SPA está no passo horizontal, em que se utiliza a Equação 4.3.33 em vez da Equação 4.3.28.

Figura 29 – Gráfico da função $f(x) = \left| \ln \left(\tanh \frac{|x|}{2} \right) \right|$.



fonte: Produção do próprio autor.

Exemplo 4.3.2. Considerando o mesmo código do Exemplo 4.3.1, a mesma palavra-código $\mathbf{c}$ e a mesma mensagem recebida no decodificador $\mathbf{y}$, a LLR obtida no passo de inicialização é dada por

$$\mathbf{L}_1 = [-1,03 \quad +1,20 \quad -1,05 \quad +0,88 \quad +1,08 \quad +0,91 \quad -1,18 \quad +1,04].$$

A matriz $\mathbf{Q}$, com os valores de $q_{c_n \rightarrow f_m}$ do passo de inicialização, é

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 & +0,88 & +1,08 & 0 & 0 & 0 \\ 0 & +1,20 & 0 & 0 & +1,08 & +0,91 & 0 & 0 \\ 0 & +1,20 & -1,05 & +0,88 & 0 & +0,91 & -1,18 & 0 \\ -1,03 & +1,20 & 0 & +0,88 & 0 & 0 & -1,18 & +1,04 \end{pmatrix}.$$

No passo horizontal são calculadas as informações que serão enviadas dos nós de função para os nós de variável $r_{f_m \rightarrow c_n}$ utilizando a Equação 4.3.33. Os valores de $r_{1 \rightarrow 4}$, $r_{1 \rightarrow 5}$ e $r_{2 \rightarrow 2}$ são dados por

$$r_{1 \rightarrow 4} = (+) \times |1,08| = 1,08,$$

$$r_{1 \rightarrow 5} = (+) \times |0,88| = 0,88,$$

$$r_{2 \rightarrow 2} = (+)(+) \times |0,91| = 0,91.$$

Os demais valores de $r_{f_m \rightarrow c_n}$ podem ser calculados da mesma forma. Todos os valores de $r_{f_m \rightarrow c_n}$ podem ser observados na matriz

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 1,08 & 0,88 & 0 & 0 & 0 \\ 0 & 0,91 & 0 & 0 & 0,91 & 1,08 & 0 & 0 \\ 0 & 0,88 & -0,88 & 0,91 & 0 & 0,88 & -0,88 & 0 \\ -0,88 & 0,88 & 0 & 1,03 & 0 & 0 & -0,88 & 0,88 \end{bmatrix}.$$

No passo vertical da primeira iteração, a partir dos valores de $r_{f_m \rightarrow c_n}$, são calculados os valores de

$q_{c_n \rightarrow f_m}$ utilizando a Equação 4.3.29. O cálculo de $q_{4 \rightarrow 1}$, $q_{5 \rightarrow 1}$ e $q_{2 \rightarrow 2}$ é dado por

$$q_{4 \rightarrow 1} = 0,91 + 1,03 + 0,88 = 2,82,$$

$$q_{5 \rightarrow 1} = 0,91 + 1,08 = 1,99,$$

$$q_{2 \rightarrow 2} = 0,88 + 0,88 + 1,20 = 2,96.$$

O restante dos valores de $q_{c_n \rightarrow f_m}$ podem ser calculados da mesma forma. Todos os valores de $q_{c_n \rightarrow f_m}$ calculados nesse passo podem ser observados na seguinte matriz $\mathbf{Q}$:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 2,82 & 1,99 & 0 & 0 & 0 \\ 0 & 2,96 & 0 & 0 & 1,96 & 1,79 & 0 & 0 \\ 0 & 2,99 & -1,05 & 2,99 & 0 & 1,99 & -2,06 & 0 \\ -1,03 & 2,99 & 0 & 2,87 & 0 & 0 & -2,06 & 1,92 \end{bmatrix}.$$

A partir dos valores de $r_{f_m \rightarrow c_n}$ dessa iteração e das LLRs de $\mathbf{L}_1$, utilizando a Equação 4.3.30, são calculadas as novas LLRs $L_{2,n}$. O cálculo de $L_{2,1}$, $L_{2,2}$ e $L_{2,3}$ é dado por

$$L_{2,1} = -0,88 - 1,03 = -1,91,$$

$$L_{2,2} = 0,91 + 0,88 + 0,88 + 1,20 = 3,87,$$

$$L_{2,3} = -0,88 - 1,05 = -1,93.$$

As demais LLRs podem ser calculadas da mesma forma. O vetor com as novas LLRs é dado por

$$\mathbf{L}_2 = \begin{bmatrix} -1,91 & +3,87 & -1,93 & +3,90 & +2,87 & +2,87 & -2,94 & +1,92 \end{bmatrix}.$$

Utilizando a regra de decisão da Equação 4.3.31, pode-se obter a palavra

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

Como a palavra encontrada satisfaz $\mathbf{c} \cdot \mathbf{H}^T = 0$, essa palavra é a palavra-código encontrada e o algoritmo para.

4.3.4 Decodificação em Camadas

Os algoritmos SPA e Mínima Soma, abordados anteriormente, também podem ser implementados utilizando a decodificação em camadas. Embora essa abordagem adicione uma maior complexidade ao processo de decodificação, ela proporciona uma convergência mais rápida do algoritmo de decodifi-

ção. A decodificação em camadas em conjunto com um algoritmo SPA utiliza, aproximadamente, apenas metade das iterações que seriam necessárias em uma decodificação SPA sem nenhuma alteração para alcançar a mesma taxa de erro (HOCEVAR, 2004).

Na decodificação em camadas, a matriz de controle de paridade é dividida em camadas horizontais, em geral de modo que cada camada contenha apenas um único elemento igual a 1 em cada coluna, e o algoritmo de decodificação, seja SPA ou Mínima Soma, é aplicado de forma sequencial em cada camada da matriz $\mathbf{H}$. Cada iteração nas camadas da matriz de controle de paridade é denominada uma sub-iteração e, após a finalização de todas as camadas, ou seja, de todas as sub-iterações, ocorre uma super-iteração.

Os passos para a execução do algoritmo Mínima Soma em camadas são apresentados a seguir. Todos os passos são baseados no algoritmo Mínima Soma tradicional, entretanto, neste caso, a matriz $\mathbf{H}$ é dividida em camadas ou submatrizes $\mathbf{H}^j$, e o algoritmo é aplicado de forma sequencial a cada uma dessas camadas. Vale ressaltar que a decodificação em camadas também pode ser aplicada ao algoritmo SPA, basta dividir a matriz $\mathbf{H}$ da mesma forma que no caso do algoritmo Mínima Soma em camadas e aplicar o algoritmo SPA de forma sequencial em cada camada.

1. Passo de Inicialização: Esse passo é realizado da mesma forma que no algoritmo SPA. Nesse passo, o valor inicial das LLRs é igual ao vetor $\mathbf{y}$ recebido no decodificador. Esses valores de LLRs são utilizados para computar a primeira camada da matriz $\mathbf{Q}$ com os primeiros valores de $q_{c_n \rightarrow f_m}$, que serão enviados dos nós de variável para os nós de função.
2. Passo Horizontal: Esse passo é aplicado a cada camada das matrizes $\mathbf{R}$ e $\mathbf{Q}$, desde as primeiras submatrizes $\mathbf{R}^1$ e $\mathbf{Q}^1$ até as últimas submatrizes $\mathbf{R}^z$ e $\mathbf{Q}^z$. O passo é dividido nas seguintes sub-iterações:

- a) Atualização da matriz $\mathbf{R}$: nessa sub-iteração, utilizando os últimos valores de $q_{c_n \rightarrow f_m}^i$ encontrados, são atualizados os valores de cada camada j da matriz $\mathbf{R}$ utilizando a equação

$$r_{f_m \rightarrow c_n}^j = \prod_{c'_n \in C(f_m) \setminus c_n} \text{sign}(q_{c'_n \rightarrow f_m}^j) \times \min_{c'_n \in C(f_m) \setminus c_n} \left(|q_{c'_n \rightarrow f_m}^j| \right). \quad (4.3.34)$$

- b) Atualização da LLR: A partir dos valores de $r_{f_m \rightarrow c_n}^j$ calculados na sub-iteração anterior e da última LLR encontrada $\mathbf{L}_n$, é calculado um novo vetor com as novas LLRs, $\mathbf{L}_{n+1}$, a partir da equação

$$L_{n+1} = L_n + \sum_{f_m \in F(c_n)} r_{f_m \rightarrow c_n}^j. \quad (4.3.35)$$

- c) Atualização da matriz $\mathbf{Q}$: nessa sub-iteração, a próxima camada $j + 1$ da matriz $\mathbf{Q}$ é atualizada utilizando os valores das LLRs calculadas na última sub-iteração.

3. Passo Vertical: Esse passo também é aplicado para todas as camadas, assim como no passo horizontal. Esse passo é dividido nas seguintes sub-iterações:

- a) Atualização da LLR: Nessa sub-iteração, a partir da última matriz $\mathbf{R}$ obtida e do último vetor de LLRs obtido, $\mathbf{L}_l$, é calculado um novo vetor de LLRs, $\mathbf{L}_{l+1}$, a partir da equação

$$L_{l+1} = L_l - \sum_{f_m \in F(c_n)} r_{f_m \rightarrow c_n}^i. \quad (4.3.36)$$

- b) Atualização da matriz $\mathbf{Q}$: A camada j da matriz $\mathbf{Q}$ é atualizada a partir das LLRs calculadas na última sub-iteração.
- c) Atualização da matriz $\mathbf{R}$: A camada j da matriz $\mathbf{R}$ é atualizada a partir da matriz $\mathbf{Q}$ calculada na sub-iteração anterior, utilizando a equação

$$r_{f_m \rightarrow c_n}^j = \prod_{c'_n \in C(f_m) \setminus c_n} \text{sign}(q_{c'_n \rightarrow f_m}^j) \times \min_{c'_n \in C(f_m) \setminus c_n} \left(|q_{c'_n \rightarrow f_m}^j| \right). \quad (4.3.37)$$

- d) Atualização da LLR: A LLR é atualizada novamente utilizando o último vetor de LLRs calculado, $\mathbf{L}_{l+1}$, e também a última matriz $\mathbf{R}$ calculada, por meio da equação

$$L_{n+2} = L_{l+1} + \sum_{f_m \in F(c_n)} r_{f_m \rightarrow c_n}^j. \quad (4.3.38)$$

4. Passo de Decisão: A palavra encontrada na super-iteração é dada a partir do último valor encontrado para as LLRs na super-iteração. Se a LLR L_{final_n} for menor que 0, é atribuído o bit 1 para essa LLR; caso contrário, é atribuído o bit 0, como pode ser observado na equação

$$\begin{cases} L_{final_n} \geq 0 \Rightarrow c_n = 0 \\ L_{final_n} < 0 \Rightarrow c_n = 1 \end{cases}. \quad (4.3.39)$$

Caso a palavra encontrada pertença ao código, essa será a palavra-código encontrada e o algoritmo para, caso contrário, uma nova super-iteração é iniciada, porém sem o passo de inicialização. Caso algum limite pré-estabelecido de iterações seja atingido, o algoritmo também para.

O algoritmo Mínima Soma em camadas pode parecer relativamente complexo. Para facilitar a compreensão do seu funcionamento, ele será exemplificado, no Exemplo 4.3.3, passo a passo utilizando um código LDPC com uma matriz $\mathbf{H}$ de dimensões pequenas.

Exemplo 4.3.3. Considerando um código LDPC com a mesma matriz $\mathbf{H}$ do Exemplo 4.3.1, a palavra-código $\mathbf{c} = [1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]$ modulada por um BPSK e transmitida por um canal Gaussiano. A palavra recebida no decodificador é

$$\mathbf{y} = [-1,03 \quad +1,20 \quad -1,05 \quad +0,88 \quad +1,08 \quad +0,91 \quad -1,18 \quad +1,04].$$

A matriz $\mathbf{H}$ em questão não pode ser dividida em camadas de forma que em cada camada exista apenas um único 1 em cada coluna, então a matriz de controle de paridade foi dividida da seguinte

forma:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

A partir dos dados fornecidos, inicia-se o processo das iterações do algoritmo Mínima Soma em camadas, dado pelos seguintes passos:

1. Inicialização

O vetor com as LLRs iniciais, dado por $\mathbf{y}$, é

$$\mathbf{L}_1 = [-1,03 \quad +1,20 \quad -1,05 \quad +0,88 \quad +1,08 \quad +0,91 \quad -1,18 \quad +1,04].$$

Os valores de $q_{c_n \rightarrow f_m}^1$ da primeira camada da matriz $\mathbf{Q}$ são atualizados utilizando os valores de $\mathbf{L}_1$ e podem ser observados em:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0,88 & 1,08 & 0 & 0 & 0 \\ 0 & 1,20 & 0 & 0 & 1,08 & 0,91 & 0 & 0 \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix}.$$

2. Passo Horizontal (1ª camada)

Utilizando a Equação 4.3.34, são calculados os valores de $r_{f_m \rightarrow c_n}^1$ da primeira camada da matriz $\mathbf{R}$. O cálculo de $r_{1 \rightarrow 4}$, $r_{1 \rightarrow 5}$ e $r_{2 \rightarrow 2}$ pode ser observado em

$$r_{1 \rightarrow 4} = (+) \times |1,08| = 1,08,$$

$$r_{1 \rightarrow 5} = (+) \times |0,88| = 0,88,$$

$$r_{2 \rightarrow 2} = (+)(+) \times |0,91| = 0,91.$$

Os demais valores de $r_{f_m \rightarrow c_n}^1$ podem ser calculados da mesma forma e são apresentados na primeira camada da matriz

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 1,08 & 0,88 & 0 & 0 & 0 \\ 0 & 0,91 & 0 & 0 & 0,91 & 1,08 & 0 & 0 \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix}.$$

A partir da matriz $\mathbf{R}$ calculada e do valor inicial das LLRs $\mathbf{L}_1$, é calculado o novo vetor com as novas LLRs, $\mathbf{L}_2$, a partir da Equação 4.3.35, que é dado por

$$\mathbf{L}_2 = [-1,03 \quad +2,11 \quad -1,05 \quad +1,96 \quad +2,87 \quad +1,99 \quad -1,18 \quad +1,04]$$

3. Passo Horizontal (2ª camada)

Os valores de $q_{c_n \rightarrow f_m}^2$ da segunda camada são dados utilizando os valores de $\mathbf{L}_2$. A segunda camada da matriz $\mathbf{Q}$ atualizada pode ser observada na matriz

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0,88 & 1,08 & 0 & 0 & 0 \\ 0 & 1,20 & 0 & 0 & 1,08 & 0,91 & 0 & 0 \\ \hline 0 & 2,11 & -1,05 & 1,96 & 0 & 1,99 & -1,18 & 0 \\ -1,03 & 2,11 & 0 & 1,96 & 0 & 0 & -1,18 & 1,04 \end{bmatrix}.$$

Os valores de $r_{f_m \rightarrow c_n}^2$ da segunda camada da matriz $\mathbf{R}$ são atualizados utilizando a segunda camada da última matriz $\mathbf{Q}$ calculada e a Equação 4.3.34. O cálculo de $r_{3 \rightarrow 2}$, $r_{3 \rightarrow 3}$ e $r_{3 \rightarrow 4}$ é dado por

$$r_{3 \rightarrow 2} = (-)(+)(+)(-) \times |-1,05| = 1,05,$$

$$r_{3 \rightarrow 3} = (+)(+)(+)(-) \times |-1,18| = -1,18,$$

$$r_{3 \rightarrow 4} = (+)(-)(+)(-) \times |-1,05| = 1,05.$$

O restante dos valores de $r_{f_m \rightarrow c_n}^2$ podem ser calculados da mesma forma e todos os valores de $r_{f_m \rightarrow c_n}^2$ da segunda camada de $\mathbf{R}$ podem ser observados na matriz

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 1,08 & 0,88 & 0 & 0 & 0 \\ 0 & 0,91 & 0 & 0 & 0,91 & 1,08 & 0 & 0 \\ \hline 0 & 1,05 & -1,18 & 1,05 & 0 & 1,05 & -1,05 & 0 \\ -1,04 & 1,03 & 0 & 1,03 & 0 & 0 & -1,03 & 1,03 \end{bmatrix}.$$

A partir dos valores da última camada da última matriz $\mathbf{R}$ calculada e dos valores do último vetor de LLRs computado, $\mathbf{L}_2$, é calculado um novo vetor de LLRs, $\mathbf{L}_3$, utilizando a Equação 4.3.35, que é dado por

$$\mathbf{L}_3 = [-2,07 \quad +4,19 \quad -2,23 \quad +4,04 \quad +2,87 \quad +3,04 \quad -3,26 \quad +2,07].$$

4. Passo Vertical (1ª camada)

A partir da Equação 4.3.36, é calculado um novo vetor $\mathbf{L}_4$ com as LLRs para a primeira camada do passo vertical, que é dado por

$$\mathbf{L}_4 = [-2,07 \quad +2,99 \quad -2,23 \quad +3,16 \quad +0,71 \quad +2,13 \quad -3,26 \quad +2,07].$$

A partir dos valores de $\mathbf{L}_4$, são atualizados os valores de $q_{c_n \rightarrow f_m}^1$ da primeira camada da matriz $\mathbf{Q}$, que pode ser observada em

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 3,16 & 0,71 & 0 & 0 & 0 \\ 0 & 2,99 & 0 & 0 & 0,71 & 2,13 & 0 & 0 \\ \hline 0 & 2,11 & -1,05 & 1,96 & 0 & 1,99 & -1,18 & 0 \\ -1,03 & 2,11 & 0 & 1,96 & 0 & 0 & -1,18 & 1,04 \end{bmatrix}.$$

A partir da primeira camada da última matriz $\mathbf{Q}$, são atualizados os valores de $r_{f_m \rightarrow c_n}^1$ da primeira camada da matriz $\mathbf{R}$ utilizando a Equação 4.3.37. O cálculo de $r_{1 \rightarrow 4}$, $r_{1 \rightarrow 5}$ e $r_{2 \rightarrow 2}$ é dado por

$$r_{1 \rightarrow 4} = (+) \times |0,71| = 0,71,$$

$$r_{1 \rightarrow 5} = (+) \times |3,16| = 3,16,$$

$$r_{2 \rightarrow 2} = (+)(+) \times |0,71| = 0,71.$$

Os demais valores de $r_{f_m \rightarrow c_n}^1$ podem ser calculados da mesma forma e todos os valores de $r_{f_m \rightarrow c_n}^1$ podem ser observados na primeira camada da matriz

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0,71 & 3,16 & 0 & 0 & 0 \\ 0 & 0,71 & 0 & 0 & 2,13 & 2,16 & 0 & 0 \\ \hline 0 & 1,05 & -1,18 & 1,05 & 0 & 1,05 & -1,05 & 0 \\ -1,04 & 1,03 & 0 & 1,03 & 0 & 0 & -1,03 & 1,03 \end{bmatrix}.$$

Ainda no passo vertical da primeira iteração da primeira camada, é calculado um novo vetor $\mathbf{L}_5$ com novas LLRs, a partir da Equação 4.3.38, utilizando a primeira camada da última matriz $\mathbf{R}$ calculada e o último vetor de LLRs calculado, $\mathbf{L}_4$. Tal vetor pode ser dado por

$$\mathbf{L}_5 = [-2,07 \quad +3,7 \quad -2,23 \quad +3,87 \quad +6 \quad +4,29 \quad -3,26 \quad +2,07].$$

5. Passo Vertical (2ª camada)

A partir do último vetor de LLRs, $\mathbf{L}_5$, calculado na camada anterior no passo horizontal, e da segunda camada da última matriz $\mathbf{R}$ calculada, é calculado um novo vetor de LLRs, $\mathbf{L}_6$,

utilizando a Equação 4.3.36, que é dado por

$$\mathbf{L}_6 = [-1,03 \quad +1,62 \quad -2,05 \quad +1,79 \quad +6 \quad +3,24 \quad -1,18 \quad +1,04].$$

Os valores $q_{c_n \rightarrow f_m}^2$ da segunda camada da matriz $\mathbf{Q}$ são atualizados a partir de $\mathbf{L}_6$, e a segunda camada atualizada da matriz $\mathbf{Q}$ pode ser observada na matriz

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 3,16 & 0,71 & 0 & 0 & 0 \\ 0 & 2,99 & 0 & 0 & 0,71 & 2,13 & 0 & 0 \\ 0 & 1,62 & -2,05 & 1,79 & 0 & 3,24 & -1,18 & 0 \\ -1,03 & 1,62 & 0 & 1,79 & 0 & 0 & -1,18 & 1,04 \end{bmatrix}.$$

Os valores de $r_{f_m \rightarrow c_n}^2$ da segunda camada da matriz $\mathbf{R}$ são atualizados a partir da segunda camada da última matriz $\mathbf{Q}$ calculada, utilizando a Equação 4.3.37. A atualização de $r_{3 \rightarrow 2}$, $r_{3 \rightarrow 3}$ e $r_{3 \rightarrow 4}$ é dada por

$$r_{3 \rightarrow 2} = (-)(+)(+)(-) \times |-1,18| = 1,18,$$

$$r_{3 \rightarrow 3} = (+)(+)(+)(-) \times |-1,18| = -1,18,$$

$$r_{3 \rightarrow 4} = (+)(-)(+)(-) \times |-1,18| = 1,18.$$

Os demais valores de $r_{f_m \rightarrow c_n}^2$ podem ser obtidos da mesma forma e todos os valores atualizados de $r_{f_m \rightarrow c_n}^2$ podem ser observados na matriz

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0,71 & 3,16 & 0 & 0 & 0 \\ 0 & 0,71 & 0 & 0 & 2,13 & 2,16 & 0 & 0 \\ 0 & 1,18 & -1,18 & 1,18 & 0 & 1,18 & -1,62 & 0 \\ -1,04 & 1,03 & 0 & 1,03 & 0 & 0 & -1,03 & 1,03 \end{bmatrix}.$$

A partir da Equação 4.3.38, utilizando a segunda camada da última matriz $\mathbf{R}$ calculada e o último vetor de LLRs calculado, $\mathbf{L}_6$, é calculado um novo vetor de LLRs, $\mathbf{L}_7$, que é dado por

$$\mathbf{L}_7 = [-2,07 \quad +3,83 \quad -3,23 \quad +4 \quad +6 \quad +4,42 \quad -3,83 \quad +2,07].$$

6. Passo de Decisão (1ª super-iteração)

Com base no último vetor de LLRs calculado nessa super-iteração, $\mathbf{L}_7$, a palavra para esta super-iteração, $\mathbf{c}$, é encontrada utilizando a regra de decisão da Equação 4.3.39. Essa palavra é dada por

$$\mathbf{c} = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0].$$

Como a palavra $\mathbf{c}$ encontrada satisfaz $\mathbf{c} \cdot \mathbf{H}^t = \mathbf{0}$, essa é a palavra-código encontrada pelo decodificador e o algoritmo para.

4.3.5 LLRs em Modulações de Ordem Maior que 2

Quando são utilizadas modulações de ordem maior que 2, como é o caso das modulações QAM, no receptor são recebidos valores correspondentes a símbolos complexos. Para utilizar os algoritmos de decodificação citados anteriormente, é necessária a conversão dos valores dos símbolos complexos recebidos em LLRs correspondentes a cada bit, as quais serão utilizadas nos passos de inicialização dos algoritmos de decodificação. A Equação 4.3.5 pode ser reescrita como

$$L_{QAM} = \ln \left(\frac{\sum_{s \in C_0} p(r_i | s)}{\sum_{s \in C_1} p(r_1 | s)} \right), \quad (4.3.40)$$

em que C_0 é o conjunto de todos os símbolos s na constelação onde o k -ésimo bit é $b_k = 0$, e C_1 é o conjunto de todos os símbolos s na constelação onde o k -ésimo bit é $b_k = 1$. Considerando um canal Gaussiano, a Equação 4.3.40 pode ser reescrita da forma

$$L_{QAM} = \ln \left(\frac{\sum_{s \in C_0} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{|r-s|^2}{\sigma^2} \right)}{\sum_{s \in C_1} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{|r-s|^2}{\sigma^2} \right)} \right) = \ln \left(\frac{\sum_{s \in C_0} \exp \left(-\frac{|r-s|^2}{\sigma^2} \right)}{\sum_{s \in C_1} \exp \left(-\frac{|r-s|^2}{\sigma^2} \right)} \right). \quad (4.3.41)$$

Utilizar a Equação 4.3.41 para o cálculo das LLRs utilizando uma modulação QAM de ordem mais elevada é computacionalmente inviável, pois o cálculo da exponencial para cada ponto da constelação, seguido de somas e um logaritmo, é computacionalmente muito caro. Devido a isso, é utilizada a aproximação

$$\ln(e^{x_1} + e^{x_2} + \dots + e^{x_N}) \approx \max(x_1, x_2, \dots, x_N), \quad (4.3.42)$$

onde é utilizado somente o maior valor de $-\frac{|r-s|^2}{\sigma^2}$, logo a aproximação resulta na equação

$$L_{QAM} \approx \max_{s \in C_0} \left(-\frac{|r-s|^2}{\sigma^2} \right) - \max_{s \in C_1} \left(-\frac{|r-s|^2}{\sigma^2} \right). \quad (4.3.43)$$

É possível notar que a operação $\max_{s \in C} \left(-\frac{|r-s|^2}{\sigma^2} \right)$ é equivalente a $-\min_{s \in C} \left(\frac{|r-s|^2}{\sigma^2} \right)$, logo a equação para o cálculo das LLRs a partir dos símbolos recebidos pode ser dada pela equação

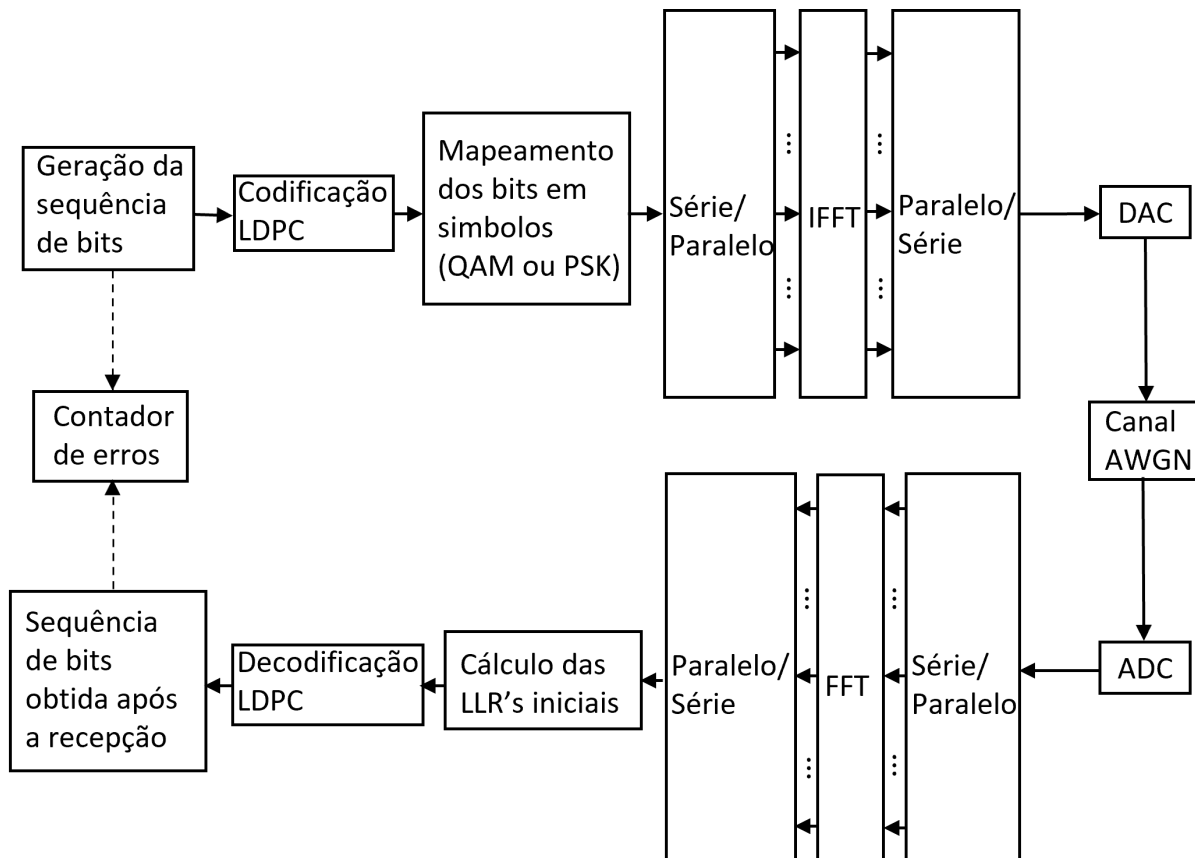
$$L_{QAM} \approx \frac{1}{\sigma^2} \left(\min_{s \in C_1} |r-s|^2 - \min_{s \in C_0} |r-s|^2 \right). \quad (4.3.44)$$

5 RESULTADOS E DISCUSSÕES

Neste capítulo será apresentado o modelo do sistema utilizado nas simulações na plataforma MATLAB, bem como os resultados obtidos a partir das simulações envolvendo códigos QC-LDPC. O objetivo é comparar o desempenho de BER dessa estratégia de codificação de canal em função de diversos parâmetros, tais como diferentes matrizes $\mathbf{H}$, distintos esquemas de modulação e vários algoritmos de decodificação com diferentes números de iterações.

O modelo das simulações pode ser representado pelo diagrama de blocos da Figura 30. Vale mencionar que, nas simulações, foi considerado um sistema OFDM em banda base. Inicialmente, é gerada a sequência de bits que será transmitida. Um valor mínimo de BER a ser simulado, considerado adequado, é de 10^{-6} , pois esse valor pode ser considerado aceitável em alguns padrões atuais de sistemas com codificação LDPC, como os padrões IEEE 802.11 (Wi-Fi) e 5G. Para obter uma BER da ordem de 10^{-6} com confiabilidade estatística, é necessário simular mais do que 10^6 bits. Além disso, como o número total de bits deve ser múltiplo do comprimento dos blocos definidos pela matriz $\mathbf{H}$, o número de bits simulados foi de 10.004.222. Após a geração dos bits a serem transmitidos, os mesmos foram codificados utilizando um código QC-LDPC com a matriz protótipo $\mathbf{P}_1$ do Exemplo 4.2.2 e com a matriz $\mathbf{P}_2$ apresentada na Equação 4.2.1.

Figura 30 – Diagrama de blocos do modelo simulado.



fonte: Produção do próprio autor.

Na Seção 2.2 vimos que em um sistema OFDM, é feita uma divisão dos bits de dados em série

para paralelo, e depois os bits são mapeados em símbolos complexos. Para uma maior simplicidade no modelo utilizado nas simulações, os bits são convertidos em símbolos complexos primeiro e, depois, o conjunto de símbolos complexos é dividido em subfluxos de símbolos complexos.

Para a comparação do desempenho entre diferentes modulações, foram utilizadas as modulações BPSK, QPSK, 16-QAM e 64-QAM (apresentadas na Subseção 2.1.1 desse trabalho) no sistema a ser simulado. Na modulação BPSK, os bits $b(i)$ foram mapeados em símbolos complexos $d(i)$ utilizando a equação

$$d(i) = \frac{1}{\sqrt{2}}[(1 - 2b(i)) + j(1 - 2b(i))]. \quad (5.0.1)$$

Para o caso QPSK, os pares de bits $b(2i)$ e $b(2i + 1)$ foram mapeados em símbolos complexos $d(i)$ utilizando a equação

$$d(i) = \frac{1}{\sqrt{2}}[(1 - 2b(2i)) + j(1 - 2b(2i + 1))]. \quad (5.0.2)$$

No caso da modulação 16QAM, quádruplas de bits $b(4i)$, $b(4i + 1)$, $b(4i + 2)$, $b(4i + 3)$ foram mapeadas em símbolos complexos $d(i)$ de acordo com a equação

$$d(i) = \frac{1}{\sqrt{10}} \{ (1 - 2b(4i))[2 - (1 - 2b(4i + 2))] + j(1 - 2b(4i + 1))[2 - (1 - 2b(4i + 3))] \}. \quad (5.0.3)$$

Para o caso 64-QAM, sêxtuplos de bits, $b(6i)$, $b(6i + 1)$, $b(6i + 2)$, $b(6i + 3)$, $b(6i + 4)$, $b(6i + 5)$ foram mapeados em símbolos complexos $d(i)$ conforme a equação

$$d(i) = \frac{1}{\sqrt{170}} \{ (1 - 2b(8i)) [8 - (1 - 2b(8i + 2)) [4 - (1 - 2b(8i + 4)) [2 - (1 - 2b(8i + 6))]]] + j(1 - 2b(8i + 1)) [8 - (1 - 2b(8i + 3)) [4 - (1 - 2b(8i + 5)) [2 - (1 - 2b(8i + 7))]]] \}. \quad (5.0.4)$$

Na Equação 5.0.1, a constelação BPSK utilizada não é a constelação padrão, pois esta está rotacionada em 45 graus. Isso se deve ao fato de ser vantajoso que todas as constelações utilizadas fiquem alinhadas em fase entre si, ou seja, um ponto BPSK, QPSK ou QAM fica nas mesmas direções de fase no plano complexo. Nas equações de mapeamento dos símbolos complexos, é possível notar um fator de normalização que serve para normalizar a energia média dos símbolos para uma energia média igual a 1.

Após a conversão dos bits codificados em símbolos complexos, é possível notar no diagrama de blocos da Figura 30 que o conjunto de símbolos complexos é dividido em subconjuntos de símbolos complexos a partir do conversor Série/Paralelo. Após isso, é aplicada uma IFFT, como apresentado na Seção 2.2. Após a IFFT, os dados resultantes são concatenados em um único vetor de dados que é transformado em um sinal contínuo e é transmitido por um canal AWGN.

Nas simulações do sistema OFDM, utilizaram-se apenas portadoras ativas, portadoras nulas e pilotos não foram consideradas. Como as portadoras nulas têm a função de evitar o vazamento de energia fora da banda desejada e esse efeito não era relevante para este trabalho, elas foram omitidas. O prefixo cíclico, normalmente empregado como intervalo de guarda para reduzir a Interferência Inter-Simbólica em canais dispersivos, também não foi utilizado, pois o foco esteve nos efeitos da codificação e das modulações em um sistema OFDM genérico sob canal AWGN.

Cada subportadora foi espaçada em 15 kHz, valor utilizado em padrões do 5G. Além disso, adotou-

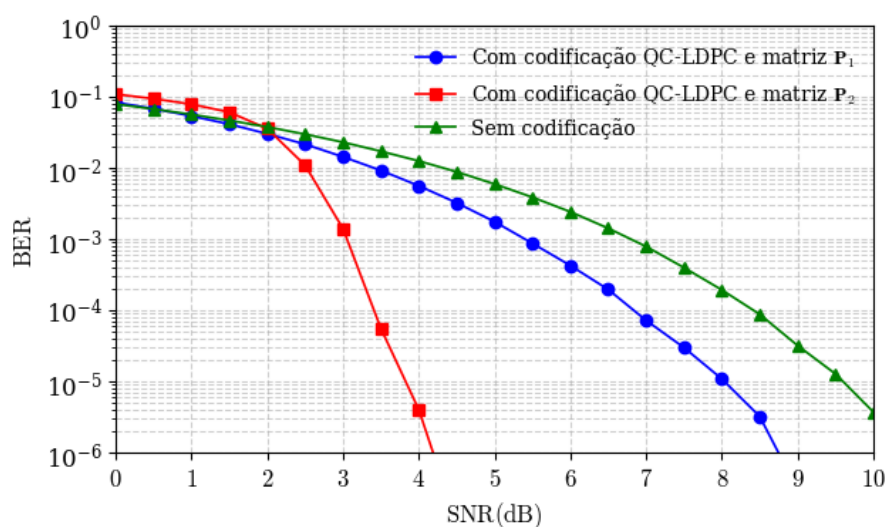
se um número de portadoras igual a 1024, potência de 2 que garante maior eficiência para os algoritmos de IFFT e FFT e representa uma largura de banda realista. Dessa forma, a largura de banda total obtida foi $B_t = 1024 \times 15 \text{ kHz} = 15,36 \text{ MHz}$.

Após a IFFT, os dados foram concatenados em um único vetor e transmitidos pelo canal Gaussiano. Em seguida, foram reconvertidos ao formato paralelo, processados pela FFT e novamente concatenados em série, resultando nos símbolos complexos recebidos.

Por fim, no modelo do sistema simulado, a partir dos valores dos símbolos complexos recebidos, os valores das LLRs do passo de inicialização do algoritmo de decodificação do sistema foram calculados da mesma maneira apresentada na Subseção 4.3.5. Em seguida, foram implementados os algoritmos de decodificação apresentados na Seção 4.3 e obteve-se a sequência de bits após a recepção.

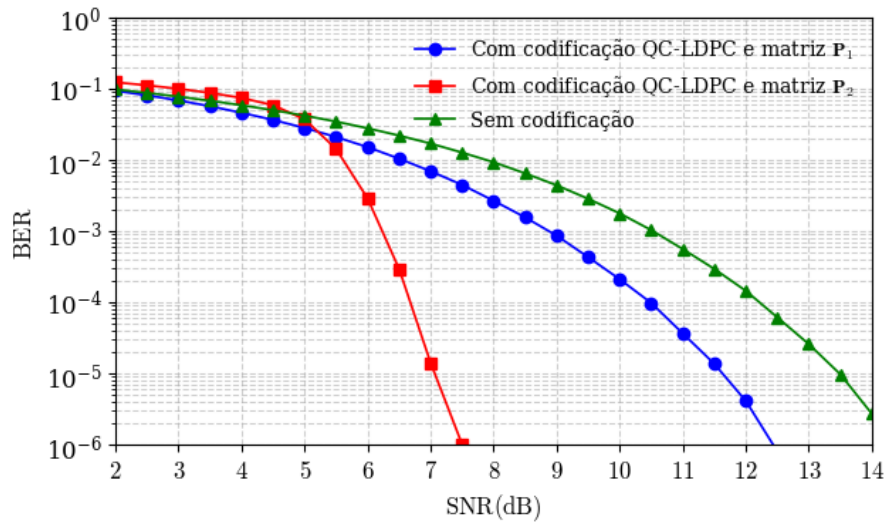
A partir do modelo apresentado acima, serão apresentados e analisados os resultados das simulações realizadas. Inicialmente, comparou-se o desempenho de um sistema OFDM sem codificação, um sistema OFDM com codificação QC-LDPC de menor complexidade utilizando a matriz P_1 e um sistema com codificação QC-LDPC com parâmetros atuais utilizando a matriz P_2 . Os sistemas com codificação QC-LDPC foram simulados utilizando o algoritmo de decodificação SPA com 25 iterações. Os resultados obtidos utilizando a modulação BPSK podem ser observados na Figura 31, e os resultados obtidos utilizando a modulação 16-QAM podem ser observados na Figura 32. Em ambos os casos, verificou-se que o sistema com codificação QC-LDPC se comportou melhor que o sistema sem codificação, o que se deve a capacidade de correção de erros da codificação QC-LDPC. Também observou-se que são obtidos melhores resultados utilizando-se a matriz protótipo P_2 . Isso se deve ao fato de a matriz de verificação de paridade gerada por essa matriz protótipo ser muito maior que a matriz H gerada a partir de P_1 . Matrizes de verificação de paridade menores tendem a ter uma maior quantidade de ciclos curtos, o que prejudica o desempenho do algoritmo de decodificação SPA.

Figura 31 – Curvas da BER para um sistema OFDM utilizando uma modulação BPSK sem codificação e com codificação QC-LDPC.



fonte: Produção do próprio autor.

Figura 32 – Curvas da BER para um sistema OFDM utilizando uma modulação 16-QAM sem codificação e com codificação QC-LDPC.



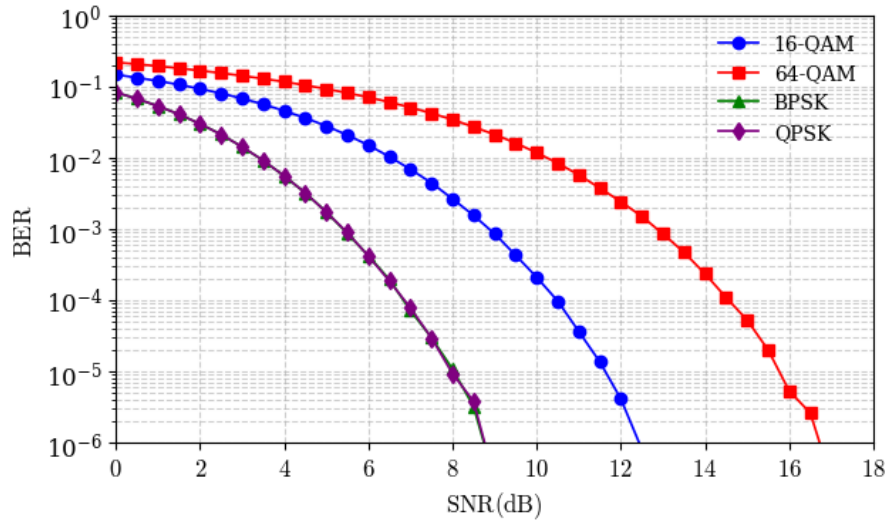
fonte: Produção do próprio autor.

Observou-se que, para valores de SNR baixos, os sistemas codificados apresentam desempenho similar ao sistema não codificado. Isso ocorre porque cada código possui um limiar de decodificação, isto é, um valor mínimo de $\frac{E_b}{N_0}$ a partir do qual o decodificador iterativo consegue convergir. Abaixo desse limiar, as mensagens trocadas no grafo tornam-se incertas e inconsistentes, impedindo a correção confiável dos erros. Além disso, em sistemas codificados, para um mesmo valor de $\frac{E_b}{N_0}$, a potência efetiva por bit de informação útil é menor do que no caso sem codificação, contribuindo para a piora da BER em baixas SNRs. Observou-se também que os valores de BER observados na Figura 32 são mais elevados quando comparados aos valores de BER da Figura 31. Isso se deve pois modulações de maior ordem, em geral, possuem uma maior densidade, o que pode resultar em valores de BER mais elevados. Porém a utilização de modulações de maior ordem permitem a transmissão de dados em taxas mais elevadas.

Conforme discutido na Subseção 2.1.1 a modulação empregada em um sistema de comunicação pode ter grande influência em seu desempenho. Para a comparação da influência de diferentes modulações na BER, realizou-se a simulação do sistema proposto utilizando as modulações BPSK, QPSK, 16-QAM e 64-QAM. Neste caso, fixamos uma codificação QC-LDPC com o algoritmo de decodificação SPA com 25 iterações. Os resultados obtidos utilizando a matriz P_1 podem ser observados na Figura 33, e os resultados obtidos utilizando a matriz P_2 podem ser observados na Figura 34. Em ambas as figuras observou-se que, embora as modulações 16-QAM e 64-QAM permitam aumentar a taxa de transmissão, elas apresentam pior desempenho em termos de BER devido à maior densidade das constelações, que reduz a distância euclidiana mínima entre símbolos e eleva a probabilidade de erro de bit. Também notou-se que as modulações BPSK e QPSK apresentaram curvas de BER idênticas, mesmo a QPSK oferecendo maior eficiência na taxa de transmissão de bits, pois essa modulação equivale à transmissão simultânea de duas sequências BPSK ortogonais cada uma utilizando metade da energia total do símbolo. Dessa forma, a energia por bit E_b é a mesma em ambas as modulações e a probabilidade de erro de bit, para um mesmo valor de $\frac{E_b}{N_0}$ é igual em ambas

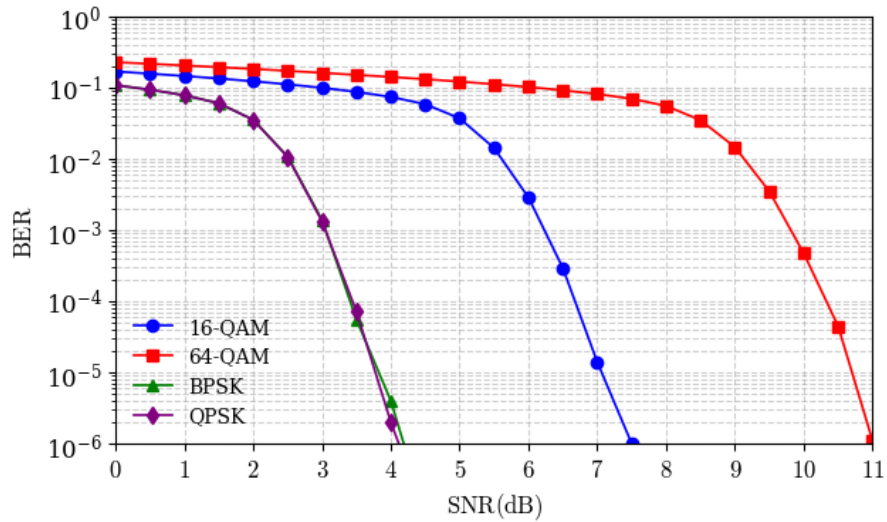
as modulações. Além disso, o ruído AWGN incide igualmente e de forma de maneira independente sobre as duas componentes, preservando a relação sinal-ruído por bit em cada dimensão.

Figura 33 – Curvas da BER para um sistema OFDM com codificação QC-LDPC utilizando a matriz P_1 e diferentes modulações.



fonte: Produção do próprio autor.

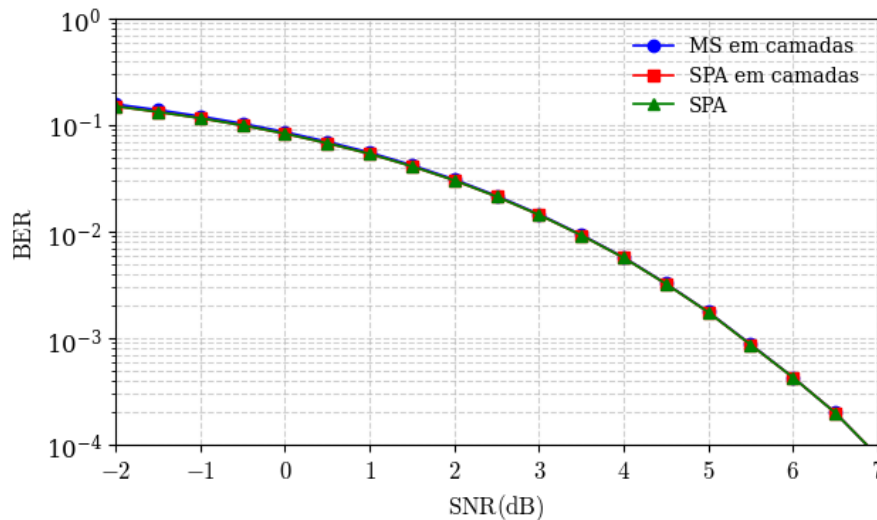
Figura 34 – Curvas da BER para um sistema OFDM com codificação QC-LDPC utilizando a matriz P_2 e diferentes modulações.



fonte: Produção do próprio autor.

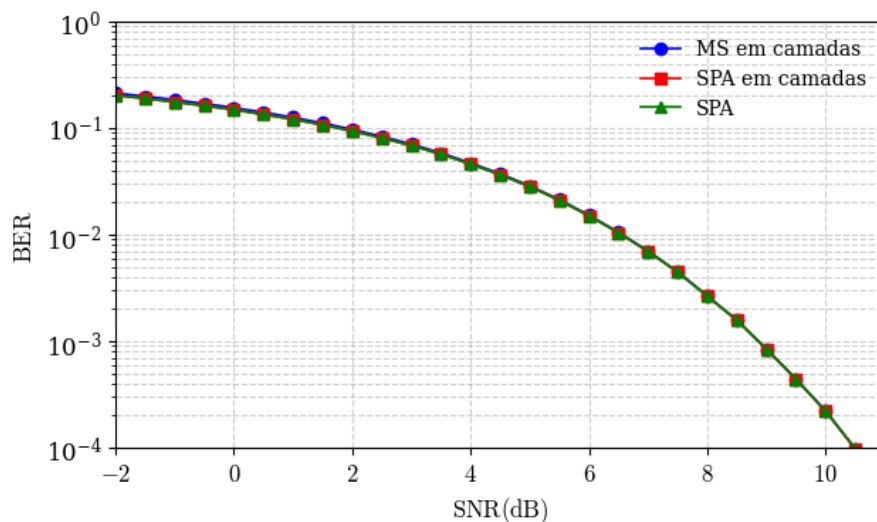
Como abordado na Seção 4.3, outro fator que pode influenciar o desempenho de um sistema é a escolha do algoritmo de decodificação a ser utilizado. Para uma comparação do desempenho de um sistema com codificação QC-LDPC com os algoritmos apresentados neste trabalho, simulou-se o sistema proposto utilizando os algoritmos SPA, SPA em camadas e MS em camadas. Inicialmente, fixamos 5 iterações no algoritmo e a matriz P_1 . Na Figura 35 foi utilizada a modulação BPSK e na Figura 36 a modulação 16-QAM.

Figura 35 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_1 , comparando diferentes algoritmos de codificação com 5 iterações.



fonte: Produção do próprio autor.

Figura 36 – Curvas da BER para um sistema OFDM com modulação 16-QAM e com codificação QC-LDPC utilizando a matriz P_1 comparando diferentes algoritmos de codificação com 5 iterações.



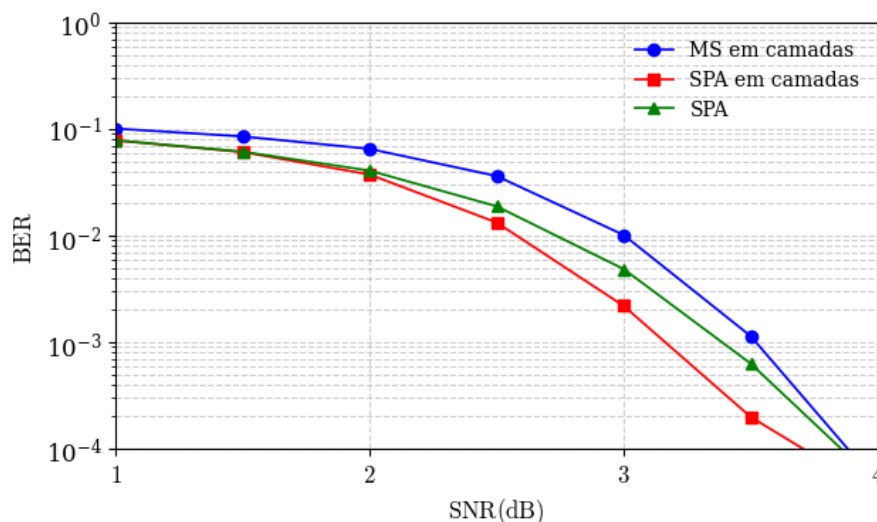
fonte: Produção do próprio autor.

Observou-se que os diferentes algoritmos de decodificação não influenciam significativamente o desempenho em termos de BER, mantendo suas curvas praticamente idênticas. Embora a decodificação

em camadas apresente uma convergência mais rápida, esse efeito não se manifesta quando a matriz $\mathbf{H}$ possui dimensões reduzidas, como a gerada a partir de $\mathbf{P}_1$, uma vez que o grafo correspondente possui poucos nós e conexões, a propagação de mensagens entre os nós ocorre de maneira completa já nas primeiras iterações, permitindo que a convergência seja atingida logo nas primeiras iterações, tanto no SPA padrão quanto no SPA em camadas. Em matrizes $\mathbf{H}$ pequenas, a aproximação feita pelo método MS tende a produzir valores de LLR próximos aos obtidos pelo SPA, pois o número de termos envolvidos na equação de atualização de mensagens em cada nó é muito pequeno. Isso justifica o fato de os métodos SPA em camadas e MS em camadas não apresentarem diferenças significativas nesse cenário.

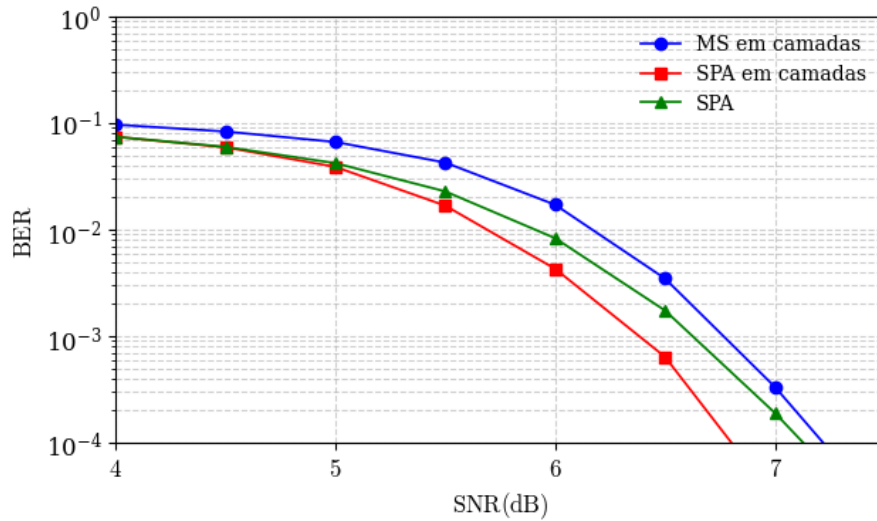
Considerando ainda 5 iterações, mas agora utilizando a matriz $\mathbf{P}_2$ com modulação BPSK na Figura 37 e 16-QAM na Figura 38, observa-se que as diferenças entre os algoritmos se tornam mais evidentes, pois o maior número de nós e de verificações de paridade no grafo associado à matriz $\mathbf{H}$ gerada por $\mathbf{P}_2$ aumenta a distância necessária para que as mensagens se propaguem por todo o código. Verificou-se que o algoritmo SPA em camadas apresenta melhor desempenho em termos de BER do que o SPA padrão, uma vez que, em códigos maiores, a atualização das mensagens com a abordagem em camadas acelera a propagação da informação entre os nós, permitindo uma convergência mais rápida e resultados superiores com o mesmo número de iterações. Além disso, notou-se que o método Min-Sum apresentou o pior desempenho em BER quando comparado aos demais algoritmos, pois sua aproximação, apesar de reduzir a complexidade computacional, gera discrepâncias maiores nas LLRs ao longo das iterações, introduzindo erros mais significativos em matrizes $\mathbf{H}$ de grande dimensão, como a obtida a partir de $\mathbf{P}_2$, comprometendo seu desempenho.

Figura 37 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz $\mathbf{P}_2$ comparando diferentes algoritmos de codificação com 5 iterações.



fonte: Produção do próprio autor.

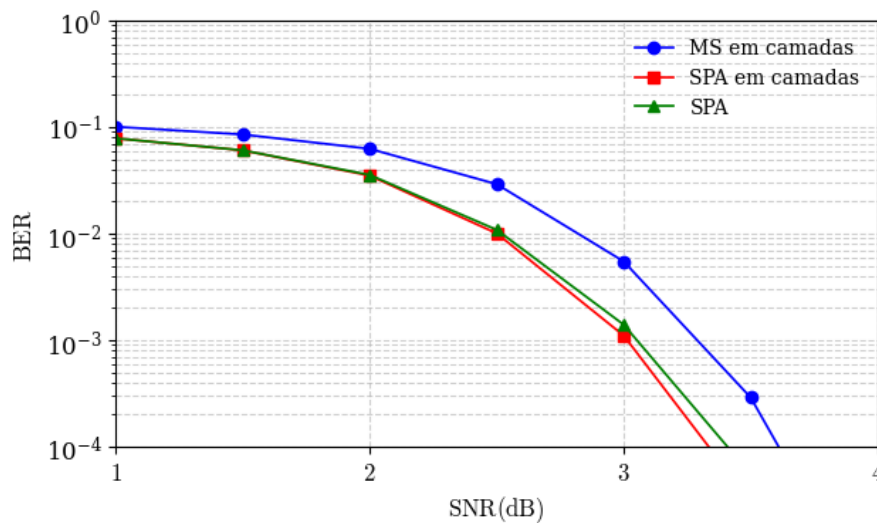
Figura 38 – Curvas da BER para um sistema OFDM com modulação 16-QAM e com codificação QC-LDPC utilizando a matriz P_2 comparando diferentes algoritmos de codificação com 5 iterações.



fonte: Produção do próprio autor.

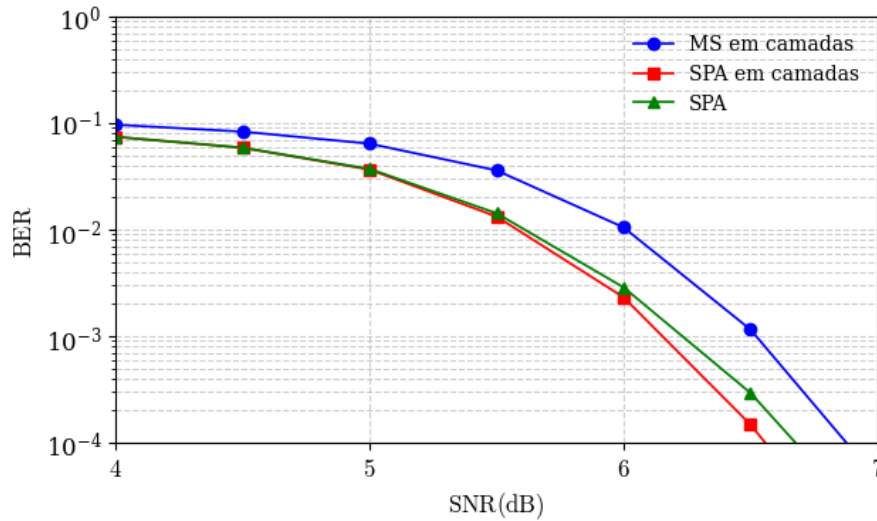
Agora, vamos fixar a matriz P_2 e realizar 25 iterações nos algoritmos de decodificação. Na Figura 39 temos os resultados utilizando a modulação BPSK e na Figura 40 utilizando a modulação 16-QAM. Notou-se que, com um maior número de iterações, as curvas de BER do SPA (padrão) e do SPA em camadas ficam mais próximas umas das outras. Isso ocorre porque ambos os algoritmos implementam o mesmo princípio de atualização de mensagens. Embora o SPA em camadas apresente uma convergência mais rápida, ou seja, alcança o mesmo desempenho com menos iterações, quando se permite um número elevado de iterações, o SPA tradicional também consegue convergir para uma mesma solução do SPA em camadas.

Figura 39 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_2 comparando diferentes algoritmos de codificação com 25 iterações.



fonte: Produção do próprio autor.

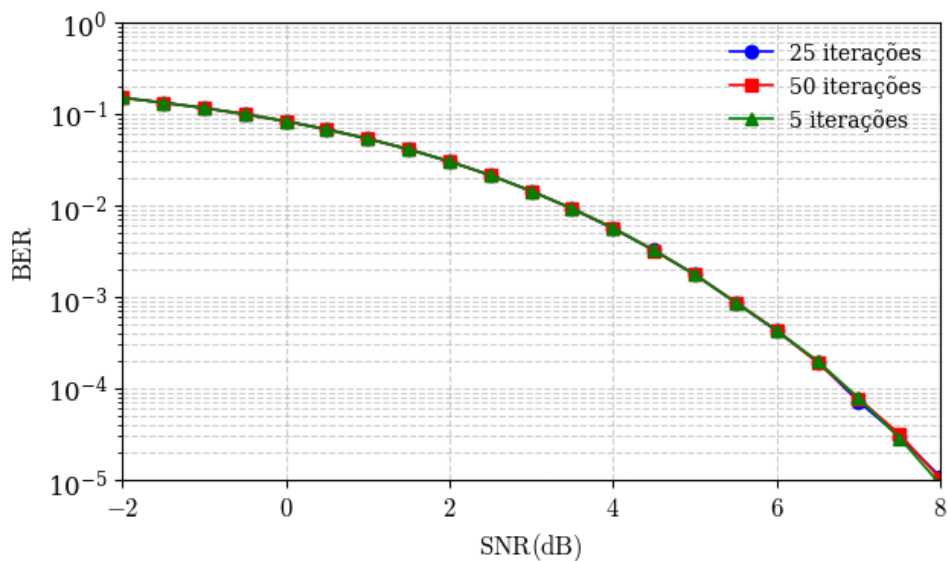
Figura 40 – Curvas da BER para um sistema OFDM com modulação 16-QAM e com codificação QC-LDPC utilizando a matriz P_2 comparando diferentes algoritmos de codificação com 25 iterações.



fonte: Produção do próprio autor.

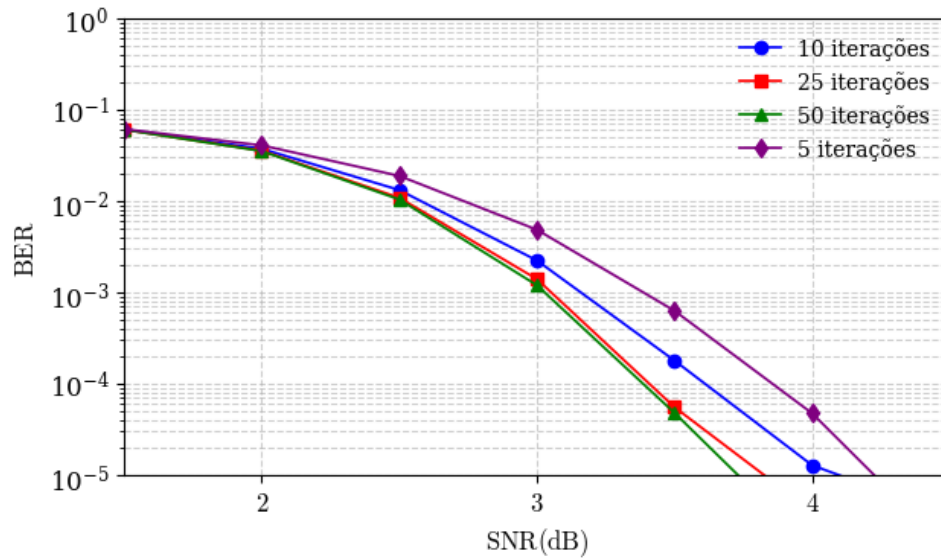
Por fim, iremos realizar a comparação do desempenho da BER de um sistema de comunicação com codificação QC-LDPC em função do número de iterações no algoritmo de decodificação. Para isso, simulou-se o sistema OFDM proposto, com modulação BPSK e utilizando o algoritmo de decodificação SPA com diferentes números de iterações. Os resultados obtidos utilizando a matriz P_1 podem ser observados na Figura 41, e os resultados obtidos utilizando a matriz P_2 podem ser observados na Figura 42.

Figura 41 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_1 e algoritmo de decodificação SPA comparando diferentes números de iterações.



fonte: Produção do próprio autor.

Figura 42 – Curvas da BER para um sistema OFDM com modulação BPSK e com codificação QC-LDPC utilizando a matriz P_2 e algoritmo de decodificação SPA comparando diferentes números de iterações.



fonte: Produção do próprio autor.

A partir da Figura 41, notou-se que, utilizando a matriz P_1 , as curvas de BER permanecem praticamente iguais independentemente do número de iterações. Isso ocorre devido às menores dimensões da matriz H gerada por P_1 , que faz com que a convergência do decodificador aconteça rapidamente, e iterações adicionais não melhoram significativamente o desempenho do sistema em função da BER. Por meio da Figura 42, notou-se que, para o código QC-LDPC gerado a partir da matriz P_2 , o número de iterações do algoritmo SPA influencia o desempenho da decodificação, reduzindo a BER a medida que aumentamos o número de iterações. Entretanto, observa-se que com 25 iterações já ocorre uma saturação da BER não sendo mais necessário aumentar o número de iterações e elevar o custo computacional.

6 CONCLUSÃO

Neste trabalho, foi apresentado um estudo detalhado sobre a utilização de códigos de correção de erros LDPC, em especial os códigos QC-LDPC, aplicados a sistemas de comunicações digitais que empregam a técnica OFDM com as modulações BPSK, QPSK, 16-QAM e 64-QAM em um canal Gaussiano com ruído AWGN. Inicialmente, foram introduzidos os conceitos fundamentais de sistemas OFDM. Em seguida, foram revisados princípios básicos de teoria da informação e codificação, essenciais para o entendimento do canal Gaussiano e dos códigos de bloco linear, classe da qual os códigos LDPC fazem parte. Posteriormente, foram apresentados conceitos e definições básicos sobre códigos LDPC, códigos QC-LDPC e sobre os algoritmos de decodificação utilizados neste trabalho.

Para facilitar a compreensão das propriedades e do funcionamento dos códigos LDPC e QC-LDPC, foram fornecidas diversas definições e exemplos ilustrativos. Da mesma forma, os algoritmos de decodificação foram detalhados passo a passo, acompanhados de exemplos didáticos utilizando matrizes de paridade reduzidas, com o objetivo de tornar o processo mais intuitivo, embora na prática sejam utilizadas matrizes de dimensões muito maiores.

O desempenho dos códigos QC-LDPC em função da BER foi avaliado por meio de simulações computacionais desenvolvidas na plataforma MATLAB, considerando um sistema OFDM utilizando modulações BPSK, QPSK, 16-QAM e 64-QAM, com constelações quadradas e transmissão sobre um canal AWGN. No modelo utilizado para as simulações, buscou-se simular um sistema OFDM com parâmetros próximos aos utilizados em padrões atuais. Também simulou-se uma quantidade de bits de maneira a alcançar uma BER de 10^{-6} , valor o qual é considerado razoável em muitos padrões modernos. Outro ponto a ser citado é que nas simulações também foi implementada uma codificação QC-LDPC com a matriz P_1 , que foi dada em um exemplo didático para explicar o funcionamento das matrizes protótipo, e também uma matriz P_2 , a qual é utilizada no padrão Wi-Fi.

Os resultados obtidos mostram que a codificação QC-LDPC melhora significativamente o desempenho do sistema em termos de BER quando comparado ao cenário sem codificação, sobretudo ao utilizar a matriz P_2 padronizada pelo IEEE 802.11, enquanto matrizes menores ou com mais ciclos curtos apresentam pior desempenho. As simulações também evidenciaram o *trade-off* entre complexidade e eficiência espectral nas modulações: BPSK e QPSK apresentaram menor BER, porém com menor taxa de transmissão, enquanto 16-QAM e 64-QAM oferecem maior eficiência espectral ao custo de maior probabilidade de erro.

Quanto aos algoritmos de decodificação, verificou-se que, para códigos curtos (matriz P_1), as diferenças de desempenho entre SPA, SPA em camadas e MS em camadas são pouco significativas, tornando o MS mais atrativo pelo menor custo computacional. Já para códigos maiores (matriz P_2), o SPA em camadas apresentou a melhor BER, seguido pelo SPA tradicional e, por último, o MS em camadas, apontando a necessidade de avaliar o compromisso entre desempenho e custo computacional. Por fim, observou-se que o número de iterações tem influência mínima em códigos pequenos, mas é determinante em códigos maiores, melhorando o desempenho até um limite de saturação. Assim, a escolha da matriz H , do algoritmo de decodificação, do esquema de modulação e do número de

iterações deve considerar o equilíbrio entre desempenho em BER, complexidade e eficiência espectral conforme os requisitos do sistema.

Este trabalho considerou exclusivamente um canal AWGN para as análises do desempenho de sistemas OFDM com codificação QC-LDPC. Contudo, sistemas OFDM são amplamente utilizados em canais dispersivos e seletivos em frequência. Assim, como sugestão para trabalhos futuros, destaca-se a investigação de modelos de canais mais realistas, como canais com desvanecimento multipercurso, que representam cenários típicos de aplicações práticas da técnica OFDM. Com a utilização de canais com desvanecimento multipercurso, também podem ser utilizadas técnicas para combater os efeitos negativos desse tipo de canal, como, por exemplo, o uso de pilotos para equalização, bem como o emprego de prefixos cíclicos, que desempenham papel fundamental no desempenho de sistemas OFDM.

REFERÊNCIAS

- 3rd Generation Partnership Project (3GPP). *3GPP TS 38.212*. Valbonne, France: 3GPP, 2024. Technical Specification. Disponível em: <https://www.3gpp.org/ftp/Specs/archive/38_series/38.212/>.
- CHANG, R. Synthesis of band-limited orthogonal signals for multichannel data transmission. **Bell System Technical Journal**, v. 45, n. 10, p. 1775–1796, 1966.
- CLAIR, M. J. J. **Códigos LDPC Quaternários Aplicados à Técnica de Transmissão OFDM**. Dissertação (Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP - Brasil, 2012.
- COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. **Mathematics of Computation**, v. 19, n. 90, p. 297–301, 1965.
- CORDEIRO, M. A. **Decodificação Iterativa de Códigos LDPC em Canais Discretos com Quantização Uniforme**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife, PE, Brasil, 2010.
- COVER, T. M.; THOMAS, J. A. **Elements of Information Theory**. 2. ed. Hoboken, NJ - USA: Wiley-Interscience, 2006.
- EMRAN, M.; ELSABROUTY, M. Improved min-sum decoding of LDPC codes based on adaptive normalization factor. **IEEE Communications Letters**, v. 18, n. 11, p. 1859–1862, 2014.
- GALLAGER, R. G. **Low-Density Parity-Check Codes**. Tese (Doutorado) — Massachusetts Institute of Technology (MIT), Cambridge, MA - USA, 1963.
- GONÇALVES, M. L. M. N. S. **Códigos LDPC Quaternários Aplicados à Técnica de Transmissão OFDM**. Dissertação (Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP - Brasil, 2010.
- HAYKIN, S. **Communication Systems**. 4. ed. New York, NY - USA: John Wiley & Sons, Inc., 2001.
- HOCEVAR, D. E. A reduced complexity decoder architecture via layered decoding of LDPC codes. In: IEEE (Ed.). **Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS 2004)**. [S.l.]: IEEE, 2004. p. 107–112.
- HUFFMAN, W. C.; PLESS, V. **Fundamentals of Error-Correcting Codes**. Cambridge, Reino Unido: Cambridge University Press, 2003.
- IEEE Standards Association. **IEEE Std 802.11-2020: IEEE Standard for Information Technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements**. Piscataway, New Jersey – USA: Institute of Electrical and Electronics Engineers, 2020. Padrão técnico.
- JOHNSON, S. J. **Introducing Low-Density Parity-Check Codes**. 2006. Disponível em: <https://www.researchgate.net/publication/228977165_Introducing_Low-Density_Parity-Check_Codes>.
- LATHI, B. P.; DING, Z. **Sistemas de Comunicações Analógicos e Digitais Modernos**. 4. ed. Rio de Janeiro, RJ - Brasil: LTC, 2012.
- LEGG, A. P. **Proposta de Códigos LDPC para Canais de Resposta Parcial**. Dissertação (Mestrado) — Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC - Brasil, 2007.

- LEINER, B. M. LDPC codes – a brief tutorial. Tutorial acadêmico não publicado. 2005. Disponível em: <<https://bernh.net/media/download/papers/ldpc.pdf>>.
- LIN, S.; COSTELLO, D. J. **Error Control Coding**. Upper Saddle River, NJ - EUA: Pearson, 2004.
- MACKAY, D. J.; NEAL, R. M. Near shannon limit performance of low density parity check codes. **Electronics Letters**, v. 32, n. 18, p. 1645–1646, 1996.
- MATTOS, W. D. de. **Códigos de Verificação de Erros de Paridade de Baixa Densidade (LDPC)**. Monografia (Especialização) — Universidade Tecnológica Federal do Paraná, Curitiba, PR, Brasil, 2012.
- MITCHELL, D. G. M.; SMARANDACHE, R.; COSTELLO, J. D. J. Quasi-cyclic LDPC codes based on pre-lifted protographs. **IEEE Transactions on Information Theory**, v. 60, n. 10, p. 5856–5870, 2014.
- NAZARETH, F. J. F. **Estudo de Códigos LDPC (Low-Density Parity Check): Matrizes Regulares e Irregulares**. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.
- PRASAD, R. **OFDM for Wireless Communications Systems**. Boston, MA - USA: Artech House, 2004.
- PROAKIS, J. G. **Digital Communications**. Boston, MA - USA: McGraw-Hill, 2001.
- 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. E.; LIN, S. **Channel Codes: Classical and Modern**. Cambridge, Reino Unido: Cambridge University Press, 2009.
- SCHULZE, H.; LÜDERS, C. **Theory and Applications of OFDMA and CDMA: Wideband Wireless Communications**. Chichester, West Sussex - UK: John Wiley & Sons, 2005.
- SHANNON, C. E. A mathematical theory of communication. **Bell System Technical Journal**, v. 27, n. 3, p. 379–423, 1948.
- SKLAR, B. **Digital Communications: Fundamentals and Applications**. Upper Saddle River, NJ - USA: Prentice Hall, 2001.
- TANENBAUM, A. S.; WETHERALL, D. J. **Redes de Computadores**. São Paulo, SP - Brasil: Pearson, 2017.
- TANNER, R. M. A recursive approach to low complexity codes. **IEEE Transactions on Information Theory**, v. 27, n. 5, p. 533–547, 1981.
- THORPE, J. **Low-Density Parity-Check (LDPC) Codes Constructed from Protographs**. Pasadena, CA, 2003. 1–7 p. Disponível em: <https://tmo.jpl.nasa.gov/progress_report/42-154/154C.pdf>.
- TU, C.; ZHANG, Z. An efficient approximation for sum-product decoding of LDPC codes. In: **2007 IEEE International Symposium on Information Theory**. Nice, France: IEEE, 2007. p. 1926–1930.
- VITI, R. M. **Desempenho de Código Turbo Quaternário em Sistemas de Multiplexação por Divisão de Frequências Ortogonais**. Dissertação (Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP - Brasil, 2010.

WEINSTEIN, S. B.; EBERT, P. M. Data transmission by frequency-division multiplexing using the discrete fourier transform. **IEEE Transactions on Communications Technology**, v. 19, n. 5, p. 628–634, 1971.