

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Cíntia Porcher de Oliveira

**Análise comparativa do desempenho de códigos
corretores de erros para redes móveis**

Porto Alegre

2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Cíntia Porcher de Oliveira

**Análise comparativa do desempenho de códigos corretores
de erros para redes móveis**

Projeto de Diplomação II, apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Engenheira Eletricista

UFRGS

Orientador: Prof. Dr. Ivan Müller

Porto Alegre

2023

Cíntia Porcher de Oliveira

Análise comparativa do desempenho de códigos corretores de erros para redes móveis

Projeto de Diplomação II, apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Engenheira Eletricista

BANCA EXAMINADORA

Prof. Dr. Hamilton Duarte Klimach
UFRGS

Me. Max Feldman
UFRGS

Prof. Dr. Ivan Müller
Orientador - UFRGS

Aprovado em 11 de Maio de 2023.

Agradecimentos

O interesse por esse projeto surgiu durante o semestre que estudei comunicação digital na universidade IMT Atlantique, por isso gostaria de agradecer inicialmente à oportunidade oferecida pela UFRGS de realizar esse intercâmbio. Essa universidade pública, gratuita e de qualidade me trouxe muitas oportunidades e vivências pelas quais sou grata.

Agradeço ao Professor Ivan Müller, meu orientador, e aos doutorandos do Laboratório de Comunicações pelo auxílio no desenvolvimento do projeto.

Agradeço também aos amigos que ingressaram comigo no curso no semestre 17/1, sem o apoio e a companhia deles essa trajetória seria muito mais árdua.

Por fim, agradeço especialmente ao meu parceiro Pedro, que revisou todo o texto, ouviu as minhas ideias e reclamações e que fez lanches para mim nos dias difíceis.

Resumo

A codificação para correção de erros é uma técnica amplamente aplicada em sistemas de comunicação e possui um papel vital para o aumento da tolerância ao ruído. Existem diversas técnicas de codificação que se adaptam a diferentes aplicações, a aplicação estudada nesse trabalho é a das redes móveis, onde a alta taxa de dados, a baixa latência e a mobilidade são algumas das particularidades. Esse trabalho visa analisar o desempenho de duas técnicas, sendo elas os códigos convolucionais e códigos LDPC. Os códigos LDPC (do inglês, *Low-Density Parity-Check*) foram designados para canais de dados de redes 5G NR pelo 3GPP TS 38.212 e os códigos convolucionais foram atribuídos aos canais de controle de redes 4G pelo TS 36.212. As duas codificações foram simuladas com o auxílio do MATLAB e analisadas experimentalmente através do GNU Radio e de rádios definidos por software, seguindo os parâmetros definidos pelo 3GPP (do inglês, *3rd Generation Partnership Project*). A modulação e o seu impacto no desempenho dos decodificadores foi outro aspecto analisado nesse trabalho, foram utilizadas modulações QPSK, 16-QAM, 64-QAM e 256-QAM. Os desempenhos em relação à capacidade de correção de erros foram comparados, visando compreender as particularidades de cada técnica. Os resultados evidenciaram a importância da codificação e a sua capacidade de correção de erros. Também foi mostrado que modulações com menos símbolos, como a QPSK, são menos suscetíveis a erros. A codificação LDPC definida pelo 3GPP apresentou um desempenho superior em relação aos códigos convolucionais.

Palavras-chave: Códigos LDPC, Códigos convolucionais, 5G NR, LTE, MATLAB, GNU Radio, SDR.

Abstract

Error correction coding is a widely applied technique in communication systems and has a vital role in increasing noise tolerance. This particular study focuses on coding techniques that are suitable for mobile networks, which have unique characteristics such as high data rates, latency, and mobility. It's worth noting that there are various coding techniques available that can be tailored to different applications. This work aims to analyze the performance of two techniques: convolutional codes and LDPC (Low-Density Parity-Check) codes. LDPC codes were assigned to shared channels of 5G NR networks by 3GPP TS 38.212 and convolutional codes were assigned to control channels of LTE (4G) networks by TS 36.212. The two coding techniques were simulated with the software MATLAB and experimentally analyzed using GNU Radio and software-defined radios, following the parameters defined by 3GPP (3rd Generation Partnership Project). The modulation and its impact on the performance of the decoders was another aspect analyzed in this work, where QPSK, 16-QAM, 64-QAM and 256-QAM modulations were used. The performance in relation to error correction capacity was compared, aiming to understand the particularities of each technique. The results highlighted the importance of channel coding and its ability to correct errors. It was also shown that modulations with fewer symbols, such as QPSK, are less susceptible to errors. The LDPC code defined by 3GPP showed superior performance over the convolutional code.

Keywords: LDPC codes, Convolutional codes, 5G NR, LTE, MATLAB, GNU Radio, SDR.

Lista de Figuras

Figura 1 – Sistema de comunicação digital.	13
Figura 2 – Gráfico de Tanner.	15
Figura 3 – Comparação da taxa de erro (BER, eixo vertical) obtidas por código LDPC regular ((3,6)-regular LDPC), Turbo (<i>Turbo Code</i>) e LDPC irregular (<i>Irregular LDPC</i>) otimizado em função da relação entre a energia do sinal e do ruído (E_b/N_0), do desvio padrão (σ) e da probabilidade de erros nos bits (P_b).	16
Figura 4 – Desempenho de três decodificadores para o código de Gallager (252, 504), expresso pelo BER em função da probabilidade de erros (p).	18
Figura 5 – Exemplo de decodificação por inversão de bits com gráfico de Tanner.	19
Figura 6 – Diagrama de bloco de um codificador convolucional.	20
Figura 7 – Diagrama de estados de um codificador convolucional.	21
Figura 8 – Diagrama de treliça com a evolução temporal do decodificador.	22
Figura 9 – Contelações de modulações digitais.	23
Figura 10 – Sistema de modulação de uma portadora.	24
Figura 11 – Evolução do número de usuários por tecnologia (em bilhões).	25
Figura 12 – Comparação do BER para diferentes taxas de códigos, tamanho de mensagens e métodos.	26
Figura 13 – Conceito de um código fotográfico.	29
Figura 14 – Princípio de encurtamento do código LDPC com <i>filler</i> bits (direita) em comparação com o LDPC padrão (esquerda).	30
Figura 15 – Estrutura da matriz de paridade dos códigos LDPC para 5G NR.	31
Figura 16 – Codificador convolucional <i>tail biting</i> definido pelo 3GPP para redes LTE.	32
Figura 17 – Adição de bits finais do tipo <i>tail biting</i>	33
Figura 18 – Diagrama do experimento utilizando um cabo e um atenuador.	34
Figura 19 – Fotografia da montagem do sistema, mostrando os componentes da Figura 18.	34
Figura 20 – Diagrama de blocos do GRC para validação.	36
Figura 21 – Diagrama de blocos do GRC para realização do experimento utilizando o rádio USRP com a adição do bloco “ <i>Polyphase Clock Sync</i> ”.	37
Figura 22 – Diagrama de blocos do GRC para gerar arquivos com a mensagem codificada e modulada.	37
Figura 23 – Diagrama do experimento utilizando antenas.	38
Figura 24 – Fotografia da montagem do sistema, mostrando os componentes da Figura 23.	38

Figura 25 – Diagrama de blocos do GRC para realização do experimento com a adição do bloco Costas Loop.	40
Figura 26 – Evolução do BER com as iterações do código LDPC com modulação 16-QAM.	41
Figura 27 – Constelações dos sinais antes (em laranja) e após (em azul) um canal AWGN com um SNR de 24 dB.	42
Figura 28 – Comparação do BER com codificação LDPC e sem codificação para diversos esquemas de modulação.	43
Figura 29 – Comparação do BER com codificação Convolutiva e sem codificação para diversos esquemas de modulação.	44
Figura 30 – Comparação do BER com codificação LDPC e com codificação Convolutiva.	45
Figura 31 – Constelações produzidas pela simulação no GRC antes (em vermelho) e após (em azul) passar pelo canal AWGN com um desvio padrão de 0,6.	46
Figura 32 – Constelação recebida antes (em azul) e após (em vermelho) do processo de sincronização.	47
Figura 33 – Espectro do sinal utilizado para o experimento.	47
Figura 34 – Espectro do sinal com medidas de largura de banda e potência.	48
Figura 35 – Constelação QPSK recebida antes (em azul) e após (em vermelho) o PLL.	48
Figura 36 – Constelação 16-QAM recebida antes (em azul) e após (em vermelho) o PLL.	49
Figura 37 – Correlação entre a mensagem com codificação LDPC enviada e a mensagem recebida demodulada.	50
Figura 38 – Correlação entre a mensagem com codificação convolutiva enviada e a mensagem recebida demodulada.	50
Figura 39 – Módulo dos símbolos recebidos.	51

Lista de Tabelas

Tabela 1	– Lista de possíveis <i>codewords</i>	17
Tabela 2	– Características do rádio USRP.	28
Tabela 3	– Conjuntos de fatores de expansão Z	30
Tabela 4	– Parâmetros da simulação do código LDPC.	32
Tabela 5	– Parâmetros da simulação do código convolucional.	33
Tabela 6	– Parâmetros para codificação e modulação.	35
Tabela 7	– Parâmetros para rádio USRP.	35
Tabela 8	– Taxas de erros da simulação no GNU Radio.	45
Tabela 9	– Taxas de erros do experimento.	51

Lista de Abreviaturas

3GPP	<i>3rd Generation Partnership Project</i>
AWGN	<i>Additive White Gaussian Noise</i>
BER	<i>Bit Error Rate</i>
GRC	GNU Radio Companion
LDPC	<i>Low-Density Parity-Check</i>
LTE	<i>Long Term Evolution</i>
NR	<i>New Radio</i>
PLL	<i>Phase-Locked Loop</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
SDR	<i>Software Defined Radio</i>
SNR	<i>Signal-to-Noise Ratio</i>
USRP	<i>Universal Software Radio Peripheral</i>

Sumário

1	INTRODUÇÃO	11
2	FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA	13
2.1	Sistemas de comunicação digital	13
2.2	Técnicas de detecção e correção de erros	14
2.2.1	Códigos LDPC	15
2.2.2	Códigos convolucionais	19
2.3	Modulação digital	22
2.4	Redes móveis modernas	24
2.5	Trabalhos relacionados	25
3	MATERIAIS E MÉTODOS	27
3.1	Materiais	27
3.1.1	<i>Software</i>	27
3.1.2	<i>Hardware</i>	28
3.2	Métodos	28
3.2.1	Código LDPC NR	29
3.2.2	Código Convolucional LTE	32
3.2.3	Experimento	34
4	RESULTADOS	41
4.1	Simulação	41
4.2	Experimento	45
5	CONCLUSÕES	52
5.1	Trabalhos Futuros	52
	REFERÊNCIAS BIBLIOGRÁFICAS	54
	APÊNDICE A – ALGORITMO LDPC	58
	APÊNDICE B – ALGORITMO CONVOLUCIONAL	59
	APÊNDICE C – CÓDIGOS DESENVOLVIDOS	60

1 Introdução

A comunicação digital está cada vez mais presente na vida das pessoas e tem se inserido em diversas novas áreas da engenharia, sendo um dos principais fatores para a revolução digital. A quinta geração de redes móveis (5G), é um dos exemplos de inovação na área das telecomunicações com aplicações em diversas tecnologias de outras áreas, como carros autônomos, cirurgias à distância e outras aplicações de IoT (do inglês, *Internet of Things*) (G1, 2022).

Em um sistema de comunicação, a informação é enviada de um transmissor para um receptor que estão a uma certa distância, portanto existe um canal entre esses dois elementos que degrada a mensagem em maior ou menor grau. Essa degradação pode ser a adição de ruído, a distorção do canal ou a interferência entre símbolos dos sinais. Em sistemas sem fio, nos quais o canal normalmente é ruidoso e possui diversos fatores de interferência, esse problema é mais presente, tornando necessário o uso de técnicas mais robustas. Além de técnicas de modulação com símbolos que representam mais bits, aumentando a sua complexidade e a sua eficiência espectral, é possível adicionar técnicas de codificação para detecção e correção de erros. Essas técnicas de codificação são um processo que precede a modulação e que adicionam redundâncias à mensagem a ser enviada, aumentando a tolerância ao ruído.

Como exemplo de recente melhoria na área, um grupo de pesquisadores do MIT desenvolveu um decodificador com um algoritmo universal e com uma eficiência energética superior aos *hardwares* atuais (RIAZ *et al.*, 2021). Os pesquisadores seguem buscando formas de melhorar ainda mais o desempenho de decodificadores, e recentemente foi publicado um artigo que mostra que a reciclagem do ruído é capaz de melhorar a eficiência energética e a latência do receptor (RIAZ *et al.*, 2023).

Este projeto visa realizar uma análise de códigos convolucionais e códigos LDPC em relação a capacidade de correção de erros. Os códigos utilizados são voltados para redes móveis modernas. Essa avaliação é realizada variando a relação entre a potência do sinal e do ruído, SNR (do inglês, *Signal-to-Noise Ratio*), variando a taxa do código e o tamanho das mensagens e avaliando a taxa de erro de bit, BER (do inglês, *Bit error rate*). Um dos aspectos importantes a ser verificado é a técnica de decodificação a ser utilizada, que pode ser, por exemplo, Viterbi (VITERBI, 1967) no caso dos códigos convolucionais, sendo essa uma técnica baseada na máxima verossimilhança. A modulação do sinal é outro fator a ser considerado, serão aplicadas a modulação QPSK (do inglês, *Quadrature Phase Shift Keying*) e variações de QAM (do inglês, *Quadrature Amplitude Modulation*).

Esse tipo de análise é importante para fazer uma avaliação da capacidade de correção dessas técnicas e saber se elas são adequadas para a aplicação a qual estão sendo destinadas, para isso se avalia o custo computacional comparado à capacidade de correção. Se a aplicação da conexão sem fio se encontra em um meio urbano, com prédios e outras antenas emitindo sinais, ou se se trata de uma comunicação por satélite, onde o sinal deve atravessar a atmosfera, deve-se realizar uma modelagem do meio para avaliar a aplicabilidade do método escolhido. Porém, quando se trata de um meio com pouca interferência, como as fibras ópticas por exemplo, essa adição de complexidade à mensagem pode ser desnecessária e diminuir a eficácia espectral. Cada aplicação deve ser avaliada para definir a técnica de modulação e de codificação que melhor se adapte. As aplicações a serem analisadas neste trabalho são as redes móveis modernas, onde as altas taxas de dados, a mobilidade e a baixa latência são os principais fatores envolvidos.

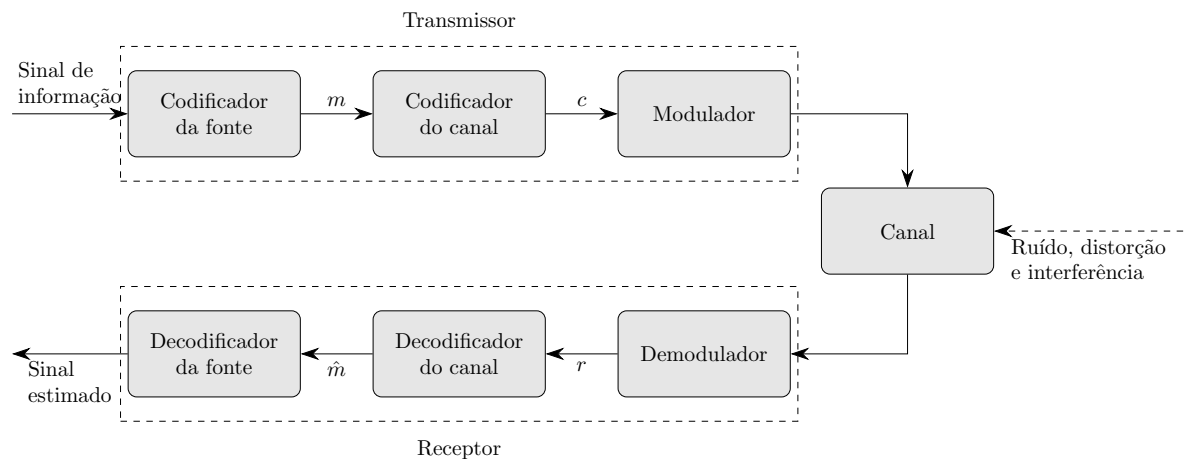
Inicialmente os códigos são construídos no software MATLAB, com seu desempenho avaliado variando-se o SNR e obtendo o BER. O sistema de comunicação deve incluir o transmissor, responsável pela modulação e codificação do sinal, o canal, que pode acrescentar ruído e introduzir interferências, e um receptor, que realiza a demodulação e decodificação do sinal. Em seguida, o sistema é analisado utilizando o software livre GNU Radio, que possibilita a implementação do sistema em um dispositivo USRP (do inglês, *Universal Software Radio Peripheral*), um SDR (do inglês, *Software Defined Radio*).

2 Fundamentação Teórica e Revisão Bibliográfica

2.1 Sistemas de comunicação digital

Um sistema de comunicação digital pode ser representado por blocos, como na Figura 1. O transmissor recebe um sinal de informação, que é inicialmente codificado, produzindo uma mensagem binária m que passa por uma segunda codificação, dessa vez uma codificação de canal, produzindo uma *codeword* c . Essa *codeword* em seguida é modulada, podendo então ser transmitida. O caminho entre o transmissor e o receptor é chamado de canal e, dependendo do meio, ele pode adicionar ruídos e distorções ao sinal. Esse meio pode ser um cabo coaxial, o espaço livre ou uma fibra óptica, por exemplo. No receptor esse sinal é inicialmente demodulado, para em seguida ser decodificado. O decodificador do canal produz \hat{m} , uma estimativa da mensagem binária, e o decodificador da fonte produz a estimativa do sinal de informação oriundo da entrada do transmissor (CARLSON; CRILLY, 2010).

Figura 1 – Sistema de comunicação digital.



Fonte: Elaborado pela autora.

O processo de digitalização das comunicações pode ser atribuído ao advento da Teoria da Informação, que nasceu com o artigo *A Mathematical Theory of Communication* (SHANNON, 1948), onde foi proposto que a informação fosse representada por bits. Além disso, também foi definido por Shannon um limite para capacidade de transmissão de um canal a partir da sua largura de banda e do seu SNR, chamado de Limite de Shannon. Esse limite é definido pela Equação (1), onde C é a capacidade em bits por segundo, B é a largura de banda do canal, S a potência média do sinal e N a potência média do ruído.

$$C_{m\acute{a}x} = B \cdot \log_2 \left(1 + \frac{S}{N} \right) \quad (1)$$

Ao codificar um canal, adicionando redundâncias para tornar possível que o receptor seja capaz de identificar e corrigir erros, a capacidade de transmissão do canal é aumentada. Essa melhora de desempenho do sistema é de baixo custo quando comparada a outras formas de redução de erros nas mensagens, como aumento do ganho ou da potência de antenas (SKLAR; HARRIS, 2004).

Até hoje não foi encontrado um meio de codificar mensagens que alcance a capacidade máxima do canal, definida pela Equação (1), de forma prática. Contudo, já foi comprovado que os códigos polares, propostos por Arikan (2009), podem atingir esse limite quando a mensagem possui um tamanho infinito, mas na prática isso não é possível. Diversas outras técnicas foram, e seguem sendo, criadas para tal e estão cada vez mais próximas dessa definição teórica.

2.2 Técnicas de detecção e correção de erros

Uma das técnicas amplamente utilizadas nos sistemas de comunicação sem fio atuais são os códigos convolucionais, capazes de corrigir erros nas mensagens de forma efetiva. Essa categoria de código foi introduzida por Peter Elias (1955), visando uma alternativa com maior capacidade de correção para os códigos de bloco, nos quais uma das principais diferenças é que os convolucionais possuem uma memória - eles armazenam bits de informação que estão transmitindo enquanto os de bloco não. Outra técnica bastante difundida são os códigos LDPC, propostos por Gallager (1962). Estes são códigos de bloco que possuem uma decodificação menos complexa computacionalmente quando comparados aos códigos convolucionais.

Para avaliação dessas técnicas é fundamental conhecer os conceitos de BER (do inglês, *bit error rate*) e SNR (do inglês, *signal-to-noise ratio*). O BER é a taxa de bits recebidos incorretamente e é definido pela Equação (2). Quanto menor for esse valor, menor será a quantidade de erros presentes e, portanto, de maior qualidade será o código. O SNR trata da relação entre a potência do sinal e a potência do ruído, relação essa comumente expressa em decibéis, e fornece uma indicação do nível de contaminação do sinal desejado pelo ruído adicionado no canal de comunicação (CARLSON; CRILLY, 2010).

$$BER = \frac{\text{n}^\circ \text{ de bits incorretos}}{\text{n}^\circ \text{ total de bits}} \quad (2)$$

2.2.1 Códigos LDPC

Os códigos LDPC, criados na década de 60, ganharam visibilidade somente décadas depois, com o trabalho de MacKay (1999), que redescobriu a técnica. Em Tanner (1981), foi criada uma representação gráfica para os códigos LDPC, atualmente chamada de gráfico de Tanner.

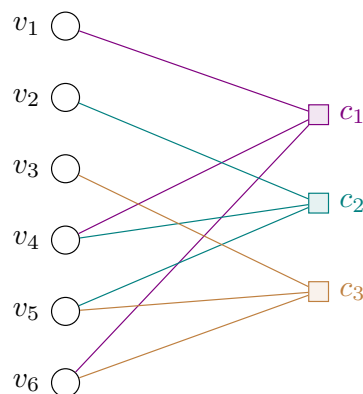
Tais códigos são definidos por uma matriz verificadora de paridade H que deve ser esparsa, ou seja, somente alguns componentes são iguais a 1, enquanto o resto é nulo. Isso leva a uma decodificação de baixa complexidade. Essa matriz possui dimensão $(n - k, k)$ onde n é o número de bits da mensagem codificada e k o número de bits de informação, logo $n - k$ é o número de bits de paridade. A taxa de codificação é dada pela razão entre k e n , conforme a Equação (3).

$$R = k/n \quad (3)$$

Como exemplo, a matriz H da Equação (4) foi construída para mensagens de três bits aos quais serão adicionados mais três bits de paridade, formando uma *codeword* de seis bits (SKLAR; HARRIS, 2004). Na Figura 2 o gráfico de Tanner ilustra as conexões entre os bits de verificação (v) e os bits de checagem (c) representadas pelos valores 1 da matriz H .

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (4) \quad G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Figura 2 – Gráfico de Tanner.

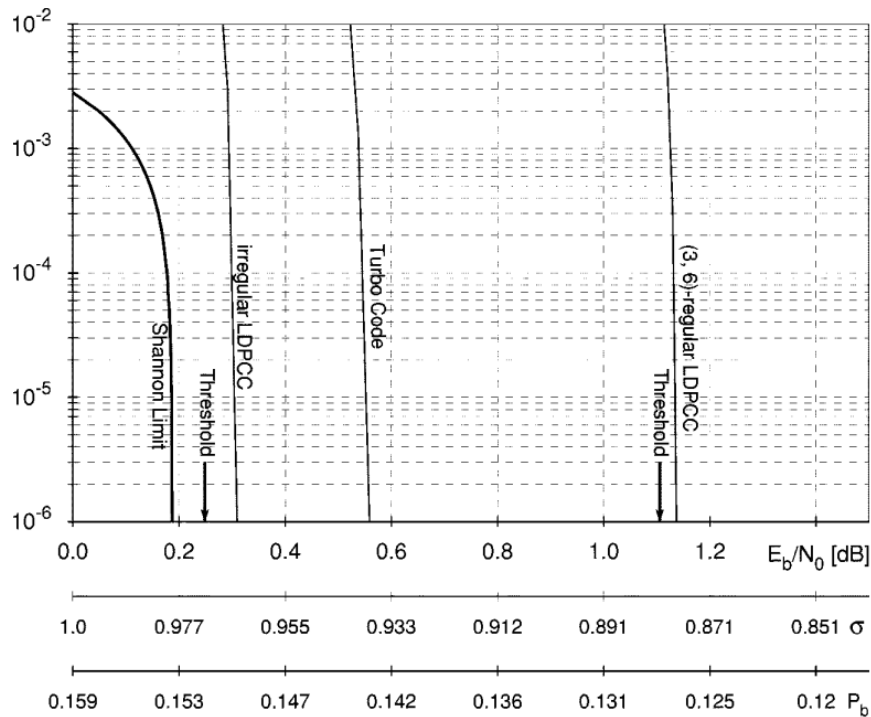


Fonte: Elaborado pela autora.

O código da matriz da Equação (4) é um código LDPC regular, pois possui o mesmo número de elementos não nulos em cada uma das suas linhas. Existem também os códigos LDPC irregulares e já foi comprovado que a sua performance pode ser superior

aos regulares e até a de códigos Turbo (uma outra classe de códigos de bloco) quando otimizado (BERROU; GLAVIEUX; THITIMAJSHIMA, 1993), como mostrado na Figura 3 (RICHARDSON; SHOKROLLAHI; URBANKE, 2001). Nesse gráfico o eixo vertical representa a taxa de erro (BER) e os eixos horizontais representam a relação entre a energia do sinal e do ruído (E_b/N_0), o desvio padrão (σ) e a probabilidade de erros nos bits (P_b).

Figura 3 – Comparação da taxa de erro (BER, eixo vertical) obtidas por código LDPC regular ((3,6)-regular LDPC), Turbo (*Turbo Code*) e LDPC irregular (*Irregular LPDC*) otimizado em função da relação entre a energia do sinal e do ruído (E_b/N_0), do desvio padrão (σ) e da probabilidade de erros nos bits (P_b).



Fonte: Richardson, Shokrollahi e Urbanke (2001).

A codificação da informação pode ser feita conforme a Equação (6), usando a mesma notação da Figura 1, onde c é a *codeword*, m é a mensagem binária a ser codificada e G é a matriz geradora. A matriz geradora G é construída a partir da matriz verificadora de paridade H . A transformação ocorre pelo método de escalonamento, para chegar no formato $H = [I_{n-k}, P^T]$, e com isso $G = [P, I_k]$. A matriz da Equação (5) foi obtida dessa forma.

$$c = mG \quad (6)$$

A matriz geradora da Equação (5) gera 8 possíveis *codewords* ($2^3 = 8$) a partir da Equação (6), visto que a mensagem é dividida em submensagens de 3 bits. As 8 possíveis mensagens e suas respectivas *codewords* estão listadas na Tabela 1.

Tabela 1 – Lista de possíveis *codewords*.

Mensagem	Codeword
000	000000
100	110100
010	011010
110	101110
001	101001
101	011101
011	110011
111	000111

Fonte: Adaptado de Sklar e Harris (2004).

Considerando o espaço vetorial \mathbb{F}_2^n (sendo \mathbb{F}_2 o espaço finito $\{0, 1\}$ e n o tamanho dos vetores binários que compõem esse espaço), um código linear (n, k) (designado por \mathcal{C}) é um subespaço linear de \mathbb{F}_2^n de dimensão 2^k (SKLAR; HARRIS, 2004). Logo as mensagens da primeira coluna da Tabela 1 formam o espaço vetorial binário de tamanho $k = 3$ ($m \in \mathbb{F}_2^3$) e as *codewords* da segunda coluna formam o subespaço vetorial \mathcal{C} ($c \in \mathcal{C}$) que pertence ao espaço vetorial $\mathbb{F}_2^{n=6}$.

Durante o processo de comunicação, a mensagem codificada c é transmitida, passa pelo canal e chega no receptor. Nesse percurso, bits podem ser invertidos, sendo modelados como um vetor de erro e (cujos valores não nulos correspondem aos locais dos bits invertidos) na mensagem recebida r , conforme a Equação (7). Nessa Equação, a operação \oplus equivale a operação lógica XOR (do inglês, *exclusive or*).

$$r = c \oplus e \quad (7)$$

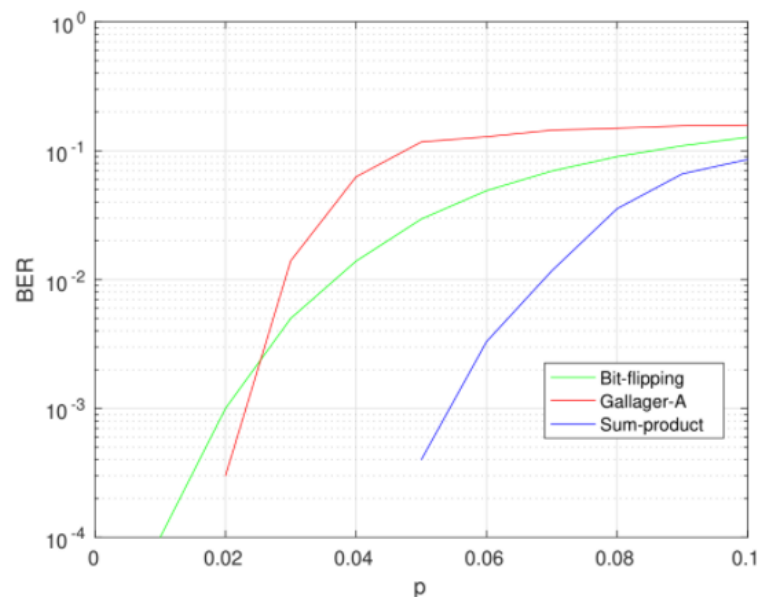
Qualquer mensagem na entrada no receptor que difira da lista da Tabela 1 indica um erro, o que não significa que se a mensagem no receptor estiver na lista ela estará correta. Por exemplo, se o receptor receber a sequência 110100, existe a possibilidade da mensagem transmitida ser 000000 e 3 bits (o primeiro, o segundo e o quarto) terem sido invertidos, porém nesse caso o receptor não será capaz de identificar o erro, visto que a *codeword* 110100 está na sua lista. O número de bits que duas *codewords* diferem (3 bits no exemplo anterior) se trata da distância de Hamming. Esse conceito é importante para correção de erros, pois, quanto maior for a distância de Hamming mínima de uma lista de *codewords*, maior será capacidade de correção do código (SKLAR; HARRIS, 2004). Daqui em diante quando citada a distância de mensagens, estará se referindo à distância de Hamming.

Para decodificação da mensagem recebida o objetivo é definir a probabilidade de cada bit do vetor r ser um ou zero, sabendo que a *codeword* estimada deve respeitar

a relação da Equação (8). Existem diversas técnicas para decodificação da mensagem recebida, entre elas os decodificadores de inversão de bits, Gallager A/B e o de soma e produto. O desempenho desses três decodificadores é comparado na Figura 4, a partir da qual é possível concluir que o decodificador de soma e produto possui um desempenho superior. No entanto, esse se trata de um decodificador mais complexo, e, dependendo da aplicação, é vantajoso utilizar uma técnica de menor consumo computacional (YE, 2019).

$$rH^T = 0 \quad \text{se } r \text{ é uma } \textit{codeword} \text{ de } \mathcal{C} \quad (8)$$

Figura 4 – Desempenho de três decodificadores para o código de Gallager (252, 504), expresso pelo BER em função da probabilidade de erros (p).



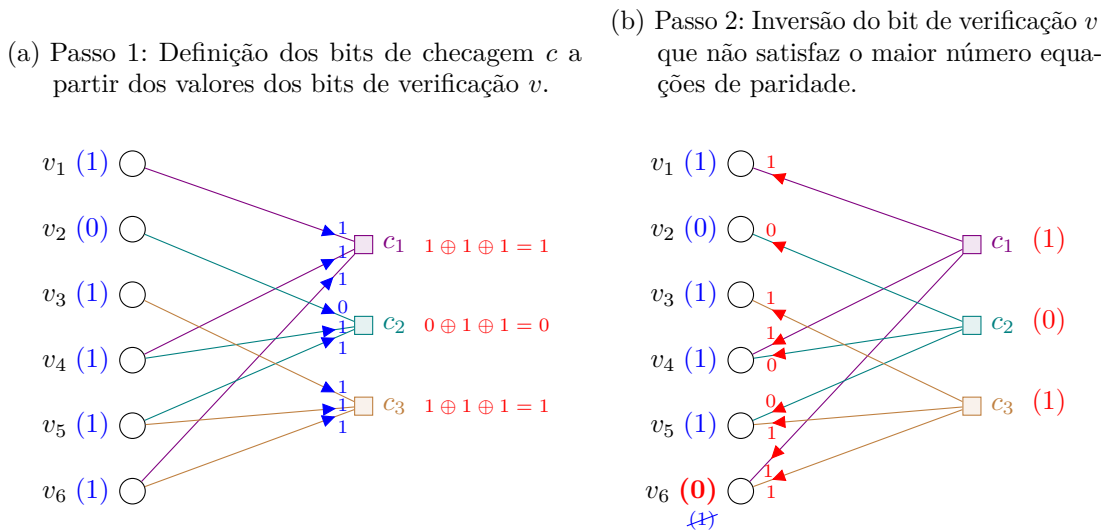
Fonte: Ye (2019).

A inversão de bits verifica (um bit por vez) se as equações de paridade estão sendo respeitadas e inverte os bits que correspondem a maior parte das equações não respeitadas (ZHANG; FOSSORIER, 2004). Já o decodificador de Gallager A/B também verifica as equações de paridade, porém ele opera nos bits de verificação, considerando apenas os bits de checagem que se conectam, ele utiliza a técnica de decisão do tipo *hard* (RICHARDSON; SHOKROLLAHI; URBANKE, 2001). Por fim, o decodificador de soma e produto difere do anterior por ser do tipo decisão *soft* (RICHARDSON; URBANKE, 2003). A decisão tipo *hard* e *soft* são duas categorias de decodificação. Decisão *hard* decodifica os bits como 1 ou 0 considerando um *threshold*, sem considerar o quão próximo ou distante está desse limite. Já a decisão do tipo *soft* considera a confiabilidade da decodificação, e não apenas o *threshold*.

Supondo que a mensagem recebida seja $r = 101111$ e utilizando o decodificador de inversão de bits, o processo está ilustrado na Figura 5. Para verificar onde as equações

de paridade são satisfeitas se considera que os bits de verificação (v) correspondem à mensagem recebida (r) e os bits os bits de checagem (c) correspondem à operação XOR dos bits de verificação aos quais são conectados, esse procedimento é demonstrado na Figura 5a. O segundo passo se trata da inversão do bit com o maior número de equações de paridade não satisfeitas, por exemplo, na Figura 5b o último bit que antes era 1 passa a ser 0 e todas as equações de paridade passam a ser satisfeitas com apenas uma iteração. Nota-se que a mensagem decodificada 101110 é uma das *codewords* presentes na Tabela 1. Esse é um processo iterativo, a cada iteração no máximo um bit é invertido.

Figura 5 – Exemplo de decodificação por inversão de bits com gráfico de Tanner.



Fonte: Elaborado pela autora.

2.2.2 Códigos convolucionais

Os códigos convolucionais, introduzidos por Peter Elias na década de 50, diferem das técnicas tradicionais por não ser uma técnica de código de bloco linear (ELIAS, 1955). Enquanto códigos de bloco enviam a mensagem original acrescida dos bits de paridade, os códigos convolucionais enviam apenas os bits de paridade.

Nessa técnica, a mensagem é codificada por uma janela de K bits, onde K é o comprimento de restrição. Quanto maior for esse comprimento, maior será a capacidade de correção de erros, mas o *trade-off* é que isso acarreta em uma decodificação mais lenta. Essa janela de K bits é transformada em r bits de paridade e a cada operação, a janela avança um bit por vez. A taxa de codificação nesse caso é $1/r$. Quanto maior o valor de r , maior a capacidade de correção de erros, porém isso também causa uma maior ocupação de banda, portanto, o objetivo é definir K e r tão pequenos quanto possível de acordo com os requisitos do sistema (MIT, 2012).

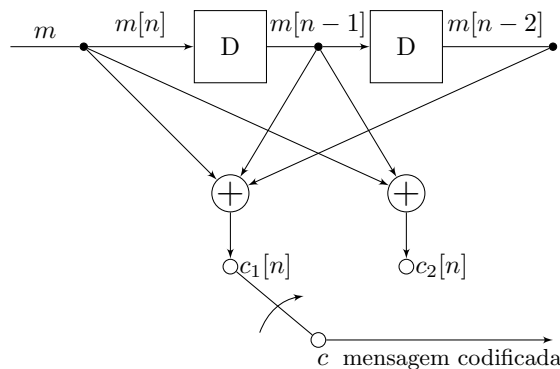
O transmissor para esse tipo de código deve contar com registradores de deslocamento, representados pelos blocos D (*delay*) na Figura 6. Outra notação possível é o z^{-1} , visto que se trata do processamento de um sinal digital. Os somadores geram os bits de paridade e são do tipo módulo 2, funcionando como uma porta lógica XOR (ou exclusivo). Ainda sobre esse exemplo, o codificador analisa uma sequência de 3 bits ($K = 3$) e entrega 2 bits de paridade ($r = 2$), logo, a taxa desse código é de $1/r = 1/2$.

$$\begin{aligned} c_1[n] &= m[n] \oplus m[n - 1] \oplus m[n - 2] \\ c_2[n] &= m[n] \oplus m[n - 1] \end{aligned} \tag{9}$$

$$c_i[n] = \sum_{j=0}^{K-1} g_i[j]m[n - j] \pmod{2} \tag{10}$$

As equações de paridade desse exemplo são as das Equações (9), e de forma generalizada podem ser representadas pela Equação (10), uma convolução de sinais discretos que dá origem ao nome desse tipo de código. A Equação (10) conta com o termo g , que é polinômio gerador do código. No caso do exemplo anterior os polinômios são $g_0 = 1, 1, 1$ e $g_1 = 1, 1, 0$.

Figura 6 – Diagrama de bloco de um codificador convolucional.

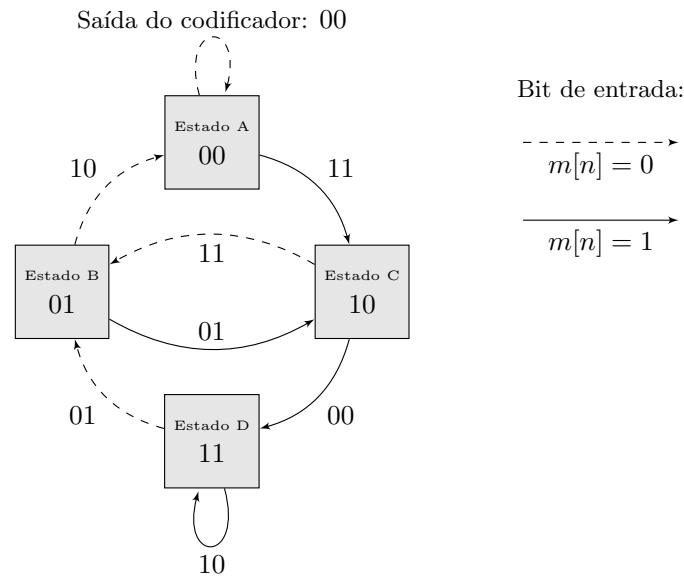


Fonte: Adaptado de MIT (2012).

Além do diagrama de bloco, outra forma de representar o codificador é através de um diagrama tipo máquina de estado. O número de estados deve ser 2^{K-1} e eles são identificados pelos bits armazenados nos registradores ($m[n - 1] m[n - 2] \dots m[n - K + 1]$), já os arcos identificam a saída do codificador ($c_1[n] c_2[n]$). A máquina de estados é uma forma visual para auxiliar a compreender o funcionamento do codificador com a passagem dos bits.

Depois de codificada, a mensagem é modulada e passa pelo canal, onde possíveis erros serão adicionados, e cabe ao receptor interpretar os bits de paridade recebidos e estimar a sequência de bits que poderia originá-los, fazendo a decodificação.

Figura 7 – Diagrama de estados de um codificador convolucional.



Fonte: Adaptado de MIT (2012).

Existem diversos métodos para decodificação de códigos convolucionais, entre eles se destacam três. O algoritmo de Viterbi (VITERBI, 1967), que se trata de uma técnica que busca a máxima verossimilhança e possui um desempenho elevado, mas possui um custo computacional maior em relação às outras. A decodificação de *feedback*, que se mostra como uma alternativa para quando o custo computacional é um fator crítico, sendo uma técnica mais simples e de desempenho inferior. Já a decodificação sequencial (WOZENCRAFT, 1957) é um meio termo entre essas duas técnicas, ela varia o seu número de iterações conforme o nível de ruído presente (CARLSON; CRILLY, 2010) (DHOLAKIA, 1994).

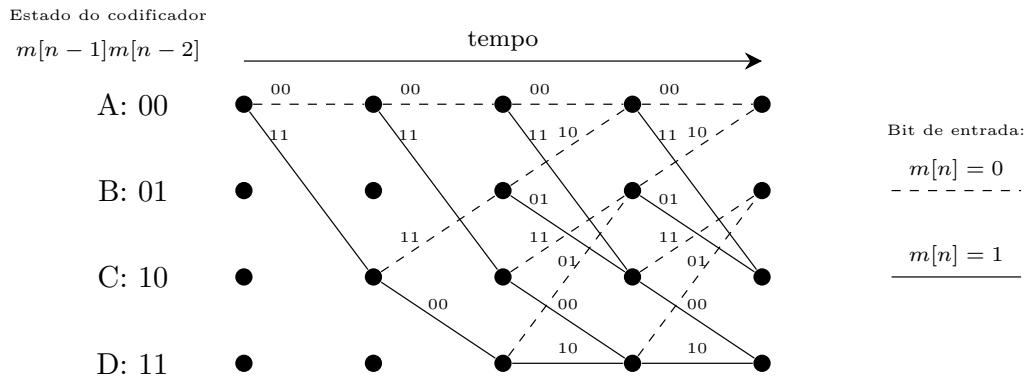
O método mais usual é o algoritmo de Viterbi, no qual o receptor deve determinar qual é a sequência de bits mais provável de ter sido enviada, sendo então um decodificador de máxima verossimilhança. Esse processo é feito a partir da comparação da mensagem recebida com as possíveis *codewords* e determinando qual é a mais verossímil. A distância de Hamming é o que determina essa semelhança entre mensagens, portanto a *codeword* que tiver a menor distância com a mensagem recebida é a mais semelhante (BEST *et al.*, 1995).

Sendo r a mensagem recebida, o objetivo é encontrar a *codeword* que maximize o valor de $P(r|\hat{c})$. Considerando uma *codeword* qualquer \hat{c} , se a distância de Hamming entre \hat{c} e r é d , a probabilidade condicional de \hat{c} ter sido a *codeword* enviada sendo que r foi a mensagem recebida, é definida pela Equação (11), onde a p é a probabilidade de receber bits incorretamente e N é o tamanho do vetor recebido r (MIT, 2012).

$$\log P(r|\hat{c}) = d \log(p) + (N - d) \log(1 - p) = d \log\left(\frac{p}{1 - p}\right) + N \log(1 - p) \quad (11)$$

O diagrama de treliça da Figura 8 é uma forma alternativa de visualizar o funcionamento do código, diferente do diagrama de estados da Figura 7, esse possui uma visualização da evolução no tempo. Iniciando pelo estado 00 (A), onde o codificador não está armazenando informações na sua memória, se o próximo bit de entrada for 0, o codificador segue no estado 00, se a entrada for 1, o codificador passa para o estado 10. A sequência de dois bits acima de cada ramo representa a saída do codificador.

Figura 8 – Diagrama de treliça com a evolução temporal do decodificador.



Fonte: Adaptado de MIT (2012).

O algoritmo de Viterbi percorre esse diagrama de treliça calculando as probabilidades e selecionando o caminho mais provável para realizar a decodificação. Sendo N o tamanho da mensagem binária, existem 2^N caminhos possíveis para se calcular a probabilidade, o que não é viável, visto que N costuma ser um valor alto. Utilizando o princípio das treliças, o número de caminhos que deve ter sua probabilidade calculada passa a ser $2^{k(K-1)}$, sendo k o número de bits na entrada e K o comprimento de restrição (CARLSON; CRILLY, 2010), não dependendo mais de N .

2.3 Modulação digital

Após o sinal digital ser codificado, ele deve ser modulado, passando essa informação digital para uma informação analógica e a tornando adequada para transmissão. A modulação pode ser feita em amplitude, frequência ou fase da portadora e a escolha da técnica de modulação deve levar em conta a eficiência espectral, a imunidade ao ruído desejada e o padrão do sistema de comunicação (CARLSON; CRILLY, 2010).

Um sinal modulado em banda passante pode ser representado pela Equação (12), onde a frequência ω_c , a amplitude A_c e a fase θ são constantes características da portadora.

As componentes em fase x_i e em quadratura x_q variam no tempo e caracterizam a mensagem.

$$s(t) = A_c[x_i(t) \cos(\omega_c(t) + \theta) - x_q(t) \sin(\omega_c(t) + \theta)] \quad (12)$$

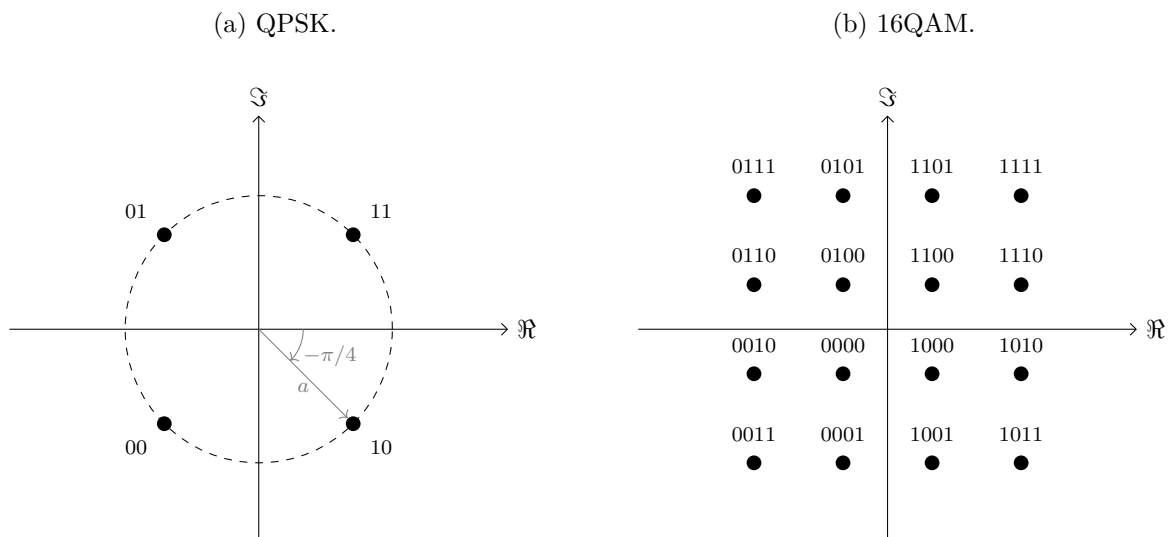
A modulação QPSK (do inglês, *Quadrature Phase-Shift Keying*) é comumente utilizada em redes móveis, e possui quatro símbolos que representam dois bits cada e o sinal é modulado em fase. Todos os símbolos possuem a mesma amplitude e uma defasagem de 90° entre eles. O alfabeto de símbolos dessa modulação está representado pela Equação (13). A Figura 9a representa uma constelação QPSK com $\alpha_0 = \pi/4$.

$$\mathcal{D}_{QPSK} = \{Ae^{i\alpha_0}, Ae^{i(\alpha_0+\pi/2)}, Ae^{i(\alpha_0+\pi)}, Ae^{i(\alpha_0+3\pi/4)}\} \quad (13)$$

A modulação QAM (do inglês, *Quadrature Amplitude Modulation*), realizada em amplitude e fase, é uma alternativa que carrega mais informação na portadora quando comparada a outras modulações. O alfabeto de símbolos da modulação M-QAM é definido pela Equação (14). A Figura 9b ilustra uma constelação 16-QAM, que possui 16 símbolos com diferentes fases e amplitudes e cada um com quatro bits.

$$\mathcal{D}_{QAM} = \{a + ib \mid (a, b) \in \mathcal{A} \times \mathcal{A}\}, \text{ onde } \mathcal{A} = \{\pm A, \pm 3A, \dots, \pm(\sqrt{M} - 1)A\} \quad (14)$$

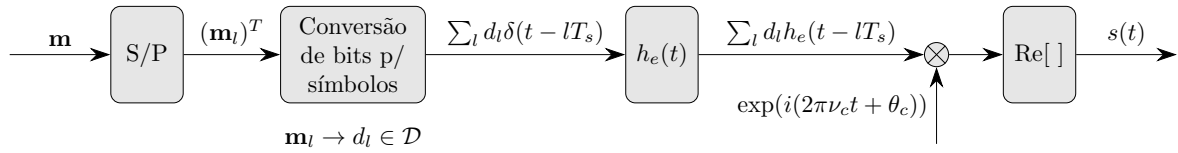
Figura 9 – Constelações de modulações digitais.



Fonte: Elaborado pela autora.

As constelações das Figuras 9a e 9b utilizam código Gray, uma regra de mapeamento que garante que símbolos “vizinhos” se diferenciem por apenas um bit. Tal regra torna a mensagem mais robusta e facilita a correção de erros.

Figura 10 – Sistema de modulação de uma portadora.



Fonte: Elaborado pela autora.

A Figura 10 ilustra a modulação de uma portadora. Uma mensagem \mathbf{m} é segmentada para em seguida ser convertida em símbolos pertencentes ao alfabeto \mathcal{D} . O bloco $h_e(t)$ é a resposta ao impulso de um filtro passa-baixas de largura de banda $B \ll \nu_c$, sendo ν_c a frequência da portadora. Com isso se chega na expressão da Equação (12).

2.4 Redes móveis modernas

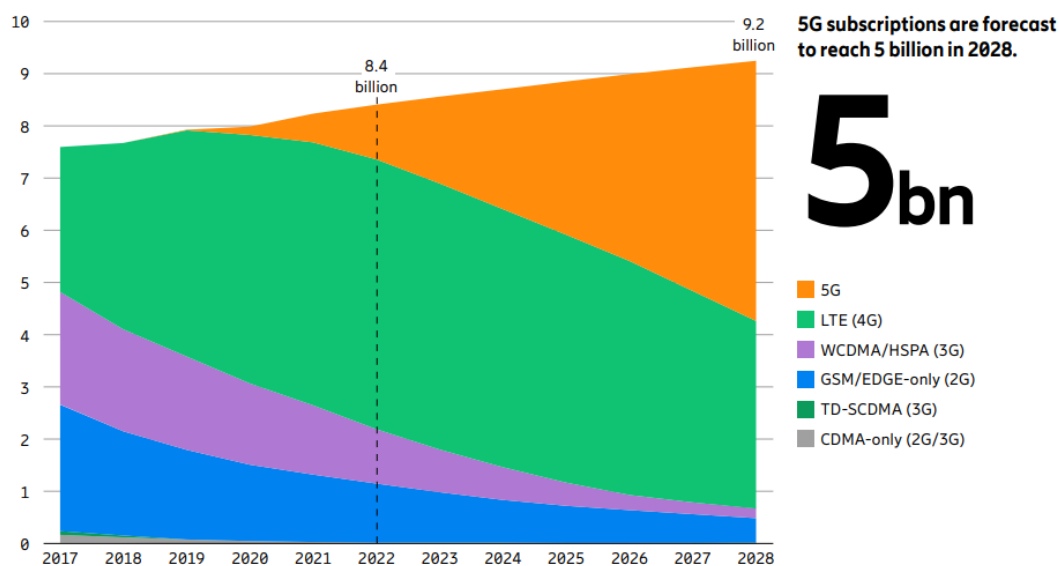
O leilão das faixas de frequência do 5G ocorreu em novembro de 2021 no Brasil e a partir disso as operadoras que adquiriram uma parcela da banda iniciaram a implementação da rede móvel de quinta geração no país (G1, 2022). Essa nova tecnologia veio para preencher requisitos do mercado atual, como conexão de baixíssima latência, alta taxa de dados e confiabilidade, e isso deve permitir o surgimento de outras tecnologias de áreas diversas que dependiam de uma rede com esses requisitos para serem concretizadas.

As faixas leiloadas pela Anatel são das frequências 700 MHz, 2,3 GHz, 3,5 GHz e 26 GHz, sendo que até o momento só foi implementado o 5G no Brasil na faixa de 3,5 GHz, a principal faixa para esse tecnologia (ANATEL, 2021). A frequência de 26 GHz, por ser muito elevada, apresenta desafios que ainda não foram solucionados. Já as outras faixas já são utilizadas por tecnologias de outras gerações.

Apesar de o 5G já ter sido implementado, a maior parte das redes móveis ainda são 4G e essa tecnologia segue recebendo investimentos das operadoras e da indústria. A Figura 11 ilustra a evolução do número de usuários por tecnologia, mostrando que o LTE (4G) ainda é a tecnologia dominante, mas que nos próximos 5 anos deve ter o seu número de usuários ultrapassado pelo 5G.

De acordo com o 3GPP TS 38.212 as técnicas de codificação para 5G NR são os códigos polares e LDPC (3GPP, 2020). Os códigos polares foram definidos para canais de controle e os LDPC para canais de dados. Na quarta geração (4G) LTE (do inglês, *Long Term Evolution*) as técnicas eram outras, códigos convolucionais para canais de controle e códigos Turbo para canais de dados (HUI *et al.*, 2018).

Figura 11 – Evolução do número de usuários por tecnologia (em bilhões).



Fonte: Ericsson (2022).

2.5 Trabalhos relacionados

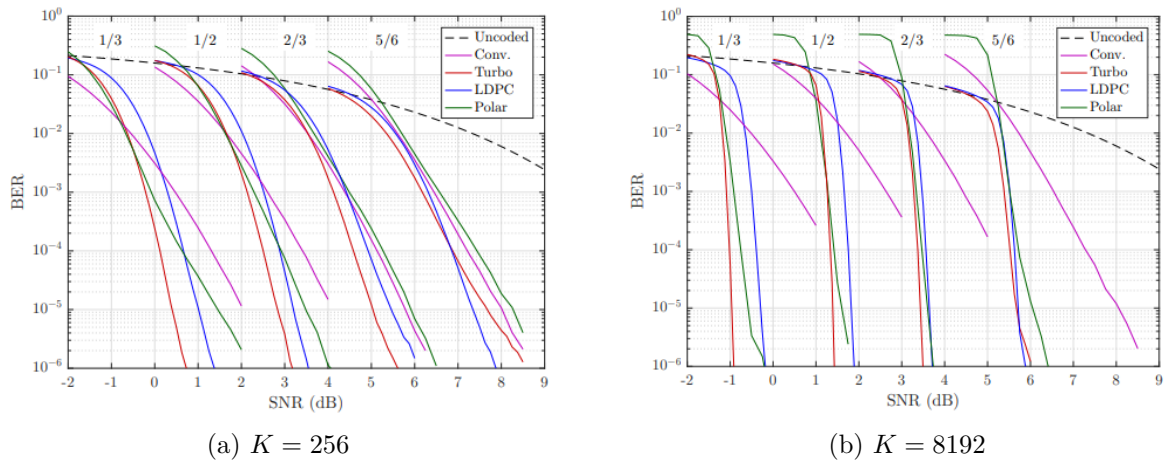
Patil, Pawar e Saquib (2020) escreveram uma revisão das técnicas de codificação do canal definidas para o 5G. Como dito anteriormente, essas técnicas foram alteradas do 4G para o 5G devido aos novos requisitos da rede e aos avanços na área. Sobre essa mudança de técnicas, Hui *et al.* (2018) fizeram um estudo bastante detalhado comparando as técnicas definidas para 4G LTE e 5G e justificando as alterações. Foi demonstrado que o desempenho dos códigos LDPC (5G) é similar a dos códigos Turbo (4G LTE), sendo a maior diferença entre as duas técnicas a capacidade de códigos LDPC atingirem uma latência inferior devido o seu processo de decodificação poder ocorrer de forma paralela. Já a diferença de performance entre códigos polares (5G) e códigos convolucionais do tipo *tail-biting* é mais evidente, sendo clara a razão para escolha de códigos polares para o plano de controle.

Riyadi, Anwar e Adriansyah (2022) estudaram a aplicação de codificação do canal para redes 5G *New Radio* em ferrovias, um ambiente complexo onde a velocidade do equipamento móvel chega a centenas de km/h, por isso o efeito Doppler é um dos fatores mais relevantes para esse estudo. Os resultados mostraram que códigos *Quasi-Cyclic* LDPC do tipo Raptor melhoram a decodificação iterativa, especialmente em casos onde o SNR é inferior a 15 dB.

Rachinger, Huber e Müller (2015) realizaram a comparação de códigos convolucionais e códigos de bloco em relação ao *delay*. Para os códigos convolucionais foram analisadas diferentes taxas, inclusive um com padrão de puncionamento, e Viterbi foi a técnica utilizada para decodificação. Também foram comparadas duas classes de códigos

de bloco, os LDPC e os código Turbo. Os LDPC analisados foram os irregulares, os do padrão WiMAX e não binários. Os códigos Turbo analisados são os do padrão 3GPP2. Foi concluído que códigos convolucionais apresentam um desempenho superior para *delays* menores, nessas regiões os códigos de bloco não foram capazes de superar o desempenho. Esse bom desempenho diminui com o aumento da taxa de código. Para faixas de *delays* não tão pequenas os códigos LDPC apresentam um desempenho superior.

Figura 12 – Comparação do BER para diferentes taxas de códigos, tamanho de mensagens e métodos.



Fonte: Tahir, Schwarz e Rupp (2017).

Tahir, Schwarz e Rupp (2017) investigaram a performance de códigos convolucionais, códigos LDPC, códigos Turbo e códigos polares em termos do BER. Utilizando modulação BPSK (do inglês, *Binary Phase Shift Keying*) em um canal AWGN (do inglês, *Additive White Gaussian Noise*), foram analisadas diferentes taxas de código e tamanhos de mensagens. Como resultado obteve-se os gráficos da Figura 12. Os esquemas obtiveram um desempenho semelhante, com a exceção dos códigos convolucionais, e essa semelhança tende a aumentar com o tamanho do bloco de mensagem.

Verma e Shrestha (2023) propuseram um algoritmo do tipo *offset min-sum* simplificado de decodificação para códigos *Quasi-Cyclic* LDPC com baixa complexidade computacional. Os cenários de teste utilizados foram baseados nas especificações do padrão 5G *New Radio* e com alterações na arquitetura clássica, como adição de processamento paralelo. Os resultados foram semelhantes a outros algoritmos do estado da arte, formando uma proposta para futuras arquiteturas.

3 Materiais e métodos

Este capítulo visa descrever a metodologia adotada no projeto de implementação de um sistema de comunicação digital com codificação do canal. O objetivo é analisar e comparar o desempenho de duas técnicas de correção de erros, os códigos LDPC e os códigos convolucionais. As duas técnicas foram estudadas inicialmente em simulações e para em seguida serem avaliadas experimentalmente.

Os materiais utilizados durante o desenvolvimento serão detalhados, assim como os métodos definidos para simulações e experimentos.

3.1 Materiais

As principais ferramentas utilizadas para realização desse estudo foram as plataformas de desenvolvimento MATLAB e GNU Radio e o rádio definido por *software* USRP.

3.1.1 *Software*

O *software* MATLAB, versão R2022b, desenvolvido pela empresa MathWorks é uma plataforma de cálculo numérico com análise interativa. Os cálculos realizados são baseados em matrizes e a partir deles o software permite o desenvolvimento de algoritmos, gráficos, análise de dados e diversas outras aplicações. Além disso, a plataforma possui diversos pacotes voltados para aplicações específicas, como sistemas de controle, aprendizado de máquina e processamento de imagens. Nesse trabalho foram utilizadas funções dos pacotes de comunicações sem fio, 5G e LTE.

O GNU Radio, versão 3.10, é uma plataforma de desenvolvimento livre que possibilita o estudo de sistemas de processamento de sinais e permite a implementação de SDR através de diagramas de blocos programados em C++ ou Python. A interface gráfica do programa é chamada de GRC (GNU Radio Companion). A plataforma disponibiliza diversos blocos já programados, mas também é possível criar blocos. Entre os blocos disponibilizados pela plataforma estão os responsáveis pela comunicação com o dispositivo USRP.

Os dois softwares são utilizados de forma paralela no sistema operacional Windows. A integração dos dois softwares foi realizada através das bibliotecas “*gnuradio*” e “*MATLABengine*” da linguagem Python.

3.1.2 Hardware

Para parte experimental o código construído no MATLAB foi implementado utilizando dois transceptores USRP. O modelo utilizado foi o USRP-2932, produzido pela National Instruments, que apresenta as características técnicas detalhadas na Tabela 2.

Tabela 2 – Características do rádio USRP.

Característica	Valor
Canais	2 (1 Tx/Rx e 1 Rx)
Banda passante	400 MHz a 4,4 GHz
Largura de banda máxima	20 MHz (16 bits) / 40 MHz (8 bits)
Potência de saída	50 a 100 mW (17 a 20 dBm)
Ganho	0 a 31 dB

Fonte: Adaptado de NI.(2023).

O *software* do rádio é executado no GNU Radio e portanto foi necessária a comunicação entre os rádios e o computador que processa o programa. Para isso foi utilizada uma rede Ethernet e um *switch* para conectar todos os elementos. O *switch* utilizado foi o SG 800 Q da marca Intelbras, ele possui 8 portas Ethernet que operam com uma taxa de até 1 Gbps.

O experimento foi realizado utilizando duas antenas omnidirecionais de 900 MHz. Como alternativa às antenas, foi utilizado um cabo coaxial de RF para interligação direta entre transmissor e receptor. O cabo de modelo LMR-195¹ possui uma velocidade de propagação de 80% da velocidade da luz no vácuo. Esse cabo deve ser conectado a um atenuador para não danificar o *front-end* de RF do SDR, e o modelo utilizado foi o VAT-30+², que possui atenuação de 30 dB.

Para analisar o funcionamento dos rádios e a configuração do sinal, foi utilizado um analisador de espectro, um instrumento que mede a potência do sinal em função da frequência, permitindo observar a ocupação do espectro de frequências. O equipamento utilizado foi o modelo N9912A da marca Agilent Technologies³. No modo SA (do inglês, *Spectrum Analyser*) o equipamento opera na faixa de 100 kHz a 6 GHz.

3.2 Métodos

O sistema de comunicação simulado segue o modelo da Figura 1. Inicialmente foi gerado um sinal discreto m do tipo PRBS (do inglês, *Pseudorandom Binary Sequence*). Esse

¹ Disponível em: <https://timesmicrowave.com/wp-content/uploads/2022/06/lmr-195-datasheet.pdf>. Acesso em: 25 de março de 2023.

² Disponível em: <https://www.minicircuits.com/pdfs/VAT-30+.pdf>. Acesso em: 25 de março de 2023.

³ Disponível em: <https://www.keysight.com/br/pt/assets/9018-02280/user-manuals/9018-02280.pdf>. Acesso em 24 de abril de 2023.

signal m passa por uma codificação de canal, que é o processo que possibilita a identificação e correção de erros na recepção. Para esse processo, foram aplicados diferentes códigos.

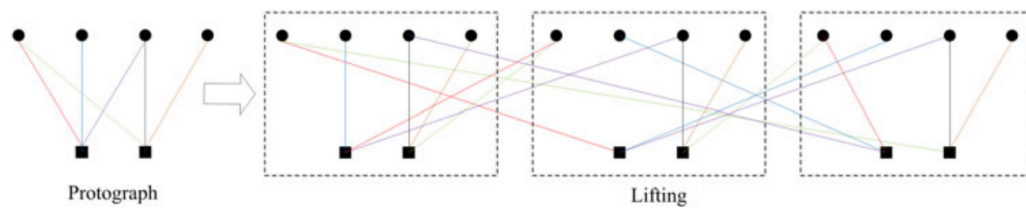
O padrão definido pelo 3GPP para o 5G NR designa códigos LDPC para canais de dados e códigos polares para canais de controle. Essa análise se baseia nos parâmetros definidos na especificação para compreensão de seu funcionamento e adequação às suas aplicações. Os códigos convolucionais foram designados para redes 4G LTE e foram adicionados a esse estudo para compreensão de suas características e comparação com as técnicas definidas para o padrão da quinta geração.

3.2.1 Código LDPC NR

O código LDPC definido pelo 3GPP para redes 5G é do tipo quasi-cíclico, sendo chamado de QC-LDPC, e pertence a família de códigos fotográficos (BAE *et al.*, 2019). A estrutura desse código será explicada na sequência. Para simulação do sistema de comunicação digital foi seguido o algoritmo do Apêndice A, utilizando o código LDPC definido pelo 3GPP para redes 5G.

A Figura 13 ilustra o conceito de códigos fotográficos através do gráfico de Tanner. São criadas cópias do código e elas são conectadas através dos bits de verificação e checagem pelas bordas. Esse processo é chamado de expansão e o número de cópias é o fator de expansão Z . A especificação 3GPP 38.212 chama essas fotografias de gráficos base (3GPP, 2020).

Figura 13 – Conceito de um código fotográfico.



Fonte: Bae *et al.* (2019).

A matriz geradora G é definida a partir da matriz de paridade, conforme a Equação (5). O padrão 3GPP TS 38.212 define duas bases para matriz de paridade para códigos LDPC, a base 1 possui 46 linhas e 68 colunas, enquanto que a base 2 possui 42 linhas e 52 colunas. São 51 valores possíveis para o fator de expansão Z e esses valores são divididos em 8 conjuntos, conforme a Tabela 3. Cada conjunto possui a sua matriz de paridade fundamentada por um gráfico base, portanto são 16 matrizes de paridade possíveis.

A base 1 é capaz de codificar um bloco de até 8448 bits e possui uma taxa de codificação de $1/3$. Já a base 2 é capaz de codificar até 3840 bits, tendo uma taxa de codificação de $1/5$. O fator de expansão Z é definido na Equação (15), sendo n_{col} o número de colunas, n_{rows} o número de linhas e K o número de bits de informação.

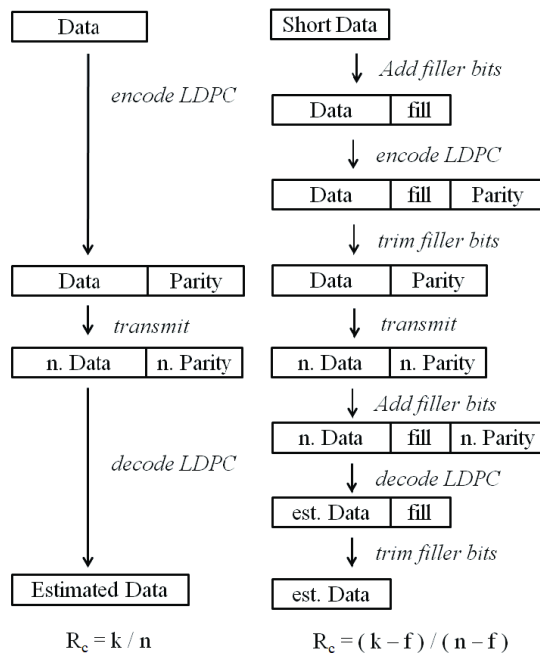
Tabela 3 – Conjuntos de fatores de expansão Z .

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

Fonte: Adaptado de 3GPP (2020).

$$Z = \frac{K}{(n_{col} - n_{rows})} \tag{15}$$

Figura 14 – Princípio de encurtamento do código LDPC com *filler* bits (direita) em comparação com o LDPC padrão (esquerda).



Fonte: PALENIK *et al.* (2015).

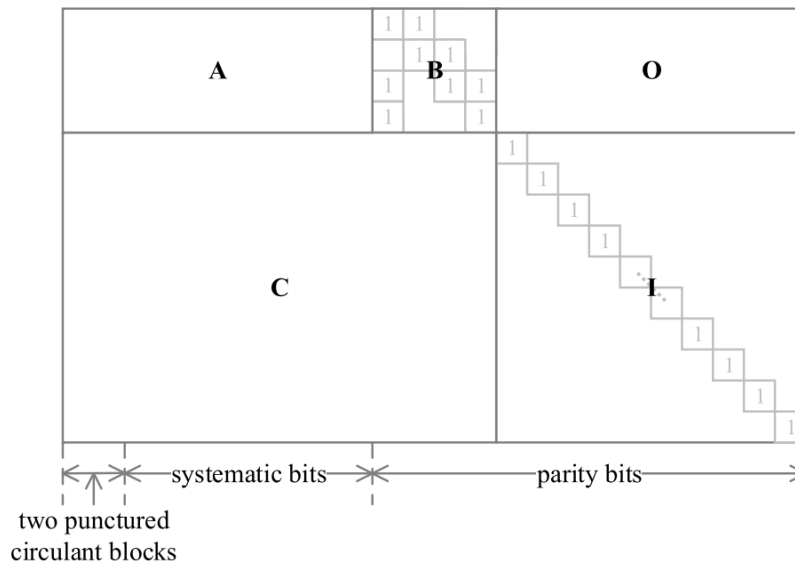
Caso o tamanho da mensagem não cumpra os pré requisitos relacionados ao fator de expansão e à base, é possível utilizar *filler* bits, ou bits de preenchimento. O objetivo desses bits é preencher a mensagem para que ela atenda aos requisitos. O uso de bits de preenchimento está ilustrado na Figura 14. Precedentemente à codificação no transmissor, é adicionado um número de bits fixo à mensagem, esses bits devem ser conhecidos pelo receptor e não transmitem informação, por isso são definidos como zeros. Após a codificação, os *filler* bits são removidos, seguindo então para o processo de modulação e transmissão.

Esse procedimento faz com que a taxa de codificação (definida na Equação (3)) seja reduzida e passe a ser definida pela Equação (16).

$$R' = \frac{k - f}{n - f} < R \quad (16)$$

Quanto aos detalhes e a estrutura das bases, as colunas são divididas em três partes, colunas de informação, colunas de paridade do núcleo e colunas de extensão de paridade. Já as linhas são divididas em dois, linhas de checagem do núcleo e linhas de checagem de paridade. Essas divisões compõem cinco sub matrizes. O formato dessas bases está ilustrado na Figura 15.

Figura 15 – Estrutura da matriz de paridade dos códigos LDPC para 5G NR.



Fonte: Wu e Wang (2019).

As primeiras duas colunas das bases não são transmitidas e correspondem ao punçionamento. As sub matrizes A e B são o núcleo do código, já C, I e O são partes de expansão (WU; WANG, 2019). A sub matriz matriz A corresponde aos bits sistemáticos, a B ao primeiro set de bits de paridade e é uma matriz quadrada de diagonal dupla, a O é uma matriz nula, a C corresponde a uma matriz de SPC (do inglês, *Single Parity-Check*) e I é uma matriz identidade (NGUYEN; TAN; LEE, 2019). A estrutura dessas matrizes é bastante complexa e esse trabalho não tem como objetivo analisar profundamente a arquitetura definida para cada base. Além disso partes das matrizes constituem a correspondência de taxas e a intercalação, processos aplicados à codificação LDPC, mas que não foram analisados nesse trabalho.

Na sua versão R2018b o MATLAB introduziu funções de codificação e decodificação conforme a especificação do 3GPP para a tecnologia New Radio. As funções “*nrLDPCen-*

code” e *nrLDPCDecode*” fazem parte da *toolbox* 5G e possuem as matrizes definidas pelo padrão, elas são utilizadas para o desenvolvimento da simulação e do experimento.

O 5G NR suporta quatro tipos de modulação, QPSK, 16-QAM, 64-QAM e 256-QAM. Foram simuladas todas essas modulações, sendo então esse um dos parâmetros variáveis. Outro parâmetro variado foi o SNR (do inglês, *Signal-to-Noise Ratio*), que corresponde a relação entre a potência do sinal e do ruído adicionado pelo canal. O canal considerado foi o AWGN (do inglês, *Additive White Gaussian Noise*), um dos mais utilizados em teoria da informação para simular ruídos adicionados à mensagem pelo meio. Os valores definidos para os parâmetros da simulação estão indicados na Tabela 4. Nota-se que como o fator de expansão é 256, a matriz utilizada corresponde ao conjunto 0, conforme a Tabela 3.

Tabela 4 – Parâmetros da simulação do código LDPC.

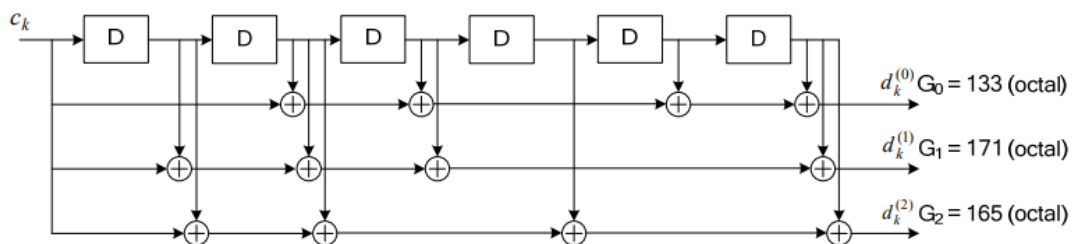
Parâmetro	Valor
Tamanho do bloco (k)	2560
Base (bgn)	2
Número de <i>filler bits</i> (f)	0
Taxa de codificação (R)	1/5
Fator de expansão (Z)	256
Quantidade de blocos	10^3
Relação sinal/ruído (E_b/N_o)	de -2 a 18 dB

Fonte: Elaborado pela autora.

3.2.2 Código Convolutacional LTE

O código convolutacional definido pelo 3GPP para redes LTE é do tipo *tail biting* com comprimento de restrição $K = 7$ bits e taxa de codificação $r = 1/3$ (3GPP, 2018). O codificador está ilustrado na Figura 16. Os polinômios geradores desse código estão definidos na Equação (17) no sistema octal e binário.

Figura 16 – Codificador convolutacional *tail biting* definido pelo 3GPP para redes LTE.

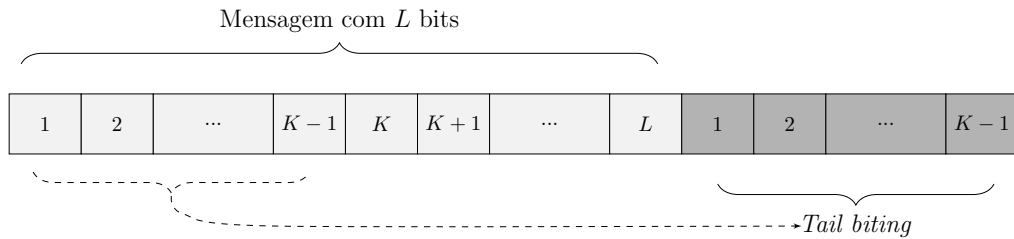


Fonte: 3GPP (2018).

$$\begin{aligned}
 g_0 &= (133)_8 = (1\ 0\ 1\ 1\ 0\ 1\ 1)_2 \\
 g_1 &= (171)_8 = (1\ 1\ 1\ 1\ 0\ 0\ 1)_2 \\
 g_2 &= (165)_8 = (1\ 1\ 1\ 0\ 1\ 0\ 1)_2
 \end{aligned}
 \tag{17}$$

Em um código convolucional tradicional, zeros são adicionados ao final dos dados para possibilitar a finalização da codificação. Em um código do tipo *tail biting* ao invés se inserir esses zeros, são inseridos os bits iniciais da mensagem, fazendo com que os dados iniciem e terminem da mesma forma e que o codificador inicie e termine no mesmo estado (OMAR *et al.*, 2013). A Figura 17 representa o funcionamento dessa variante dos códigos convolucionais.

Figura 17 – Adição de bits finais do tipo *tail biting*.



Fonte: Elaborado pela autora.

A decodificação da mensagem recebida é feita utilizando o algoritmo Viterbi, que teve a sua lógica explicada e ilustrada anteriormente, nas Figuras 7 e 8. No caso do codificador e decodificador para redes LTE existem $2^{K-1} = 64$ estados.

Tabela 5 – Parâmetros da simulação do código convolucional.

Parâmetro	Valor
Tamanho do bloco	2400
Comprimento de restrição	7
Taxa de codificação (R)	1/3
Polinômios geradores (G_1 , G_2 e G_3)	$(133)_8$, $(171)_8$ e $(165)_8$
Relação sinal/ruído (E_b/N_o)	de -2 a 18 dB

Fonte: Elaborado pela autora.

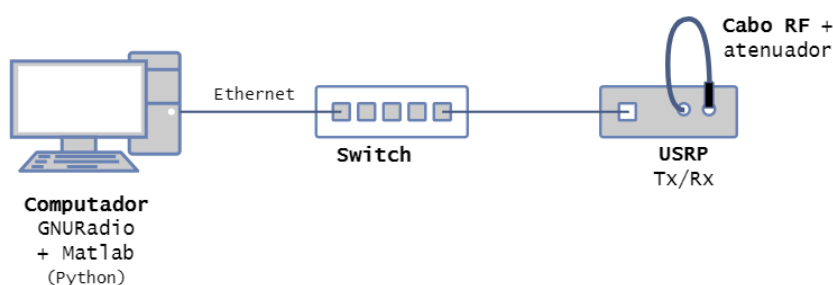
Para simulação do sistema de comunicação foi seguido o algoritmo do Apêndice B, utilizando o código Convolucional definido pelo 3GPP para redes LTE. Assim como para os códigos NR LDPC, o MATLAB disponibiliza funções de codificação e decodificação para os códigos padrões da rede LTE. Essas funções são “*lteConvolutionalEncode*” e “*lteConvolutionalDecode*”. A simulação foi feita utilizando as mesmas modulações e parâmetros do canal definidos para simulação dos códigos LDPC, porém o tamanho do bloco foi reduzido de 2560 para 2400 bits para se adequar aos pré requisitos das etapas de modulação e

demodulação. Os valores definidos para os parâmetros da simulação estão indicados na Tabela 5.

3.2.3 Experimento

Para validação dos resultados da simulação foi realizada a implementação dos códigos no USRP com auxílio do GNU Radio, visto que ele permite a conversão do sistema simulado para um SDR. O experimento foi realizado em duas partes, a primeira utilizando um cabo e a segunda utilizando antenas.

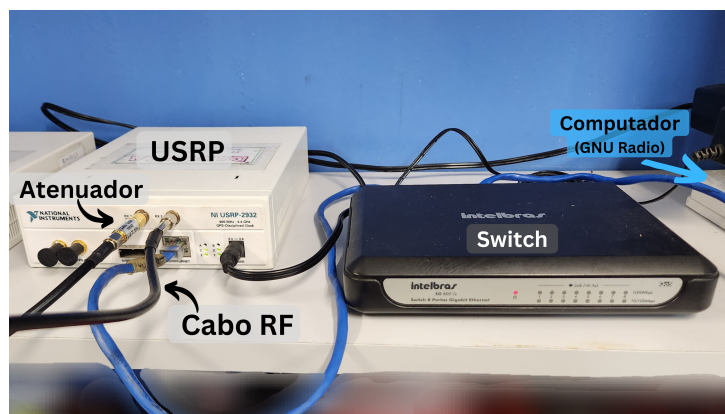
Figura 18 – Diagrama do experimento utilizando um cabo e um atenuador.



Fonte: Elaborado pela autora.

Nesse primeiro momento o experimento foi realizado utilizando um cabo coaxial para aplicações de RF junto a um atenuador, para restringir as interferências do ambiente. Apenas um rádio foi utilizado como transmissor e receptor para evitar problemas de sincronização entre os *hardwares*, conforme a estrutura do diagrama da Figura 18. A fotografia da Figura 19 mostra o *setup* experimental de acordo o diagrama. Um computador com os programas GRC e MATLAB associados foi conectado via uma rede Ethernet ao rádio USRP. A conexão via Ethernet dos equipamentos foi feita com o auxílio de um switch. O computador não está presente na fotografia, mas foi conectado via Ethernet com os outros componentes.

Figura 19 – Fotografia da montagem do sistema, mostrando os componentes da Figura 18.



Fonte: Elaborado pela autora.

O GNU Radio permite a criação de blocos utilizando a linguagem Python. Python, por sua vez, possui uma biblioteca que se integra com o MATLAB. Dessa forma o código criado no MATLAB para simulação foi transferido para o GNU Radio e ele é responsável pela configuração dos rádios USRP. Apesar de não ser uma forma prática e comum utilizar o GNU Radio associado ao MATLAB, isso foi feito por conta da complexidade das matrizes associadas aos códigos LDPC, visto que essas matrizes já estão definidas por funções do MATLAB. Para realizar a integração do MATLAB com o GNU Radio através de Python é importante estar atento às versões utilizadas. O MATLAB 2022b é compatível com Python 3.8, 3.9 e 3.10⁴, sendo 3.10 a versão do Python utilizada.

Para o experimento foram testados os códigos LDPC e Convolutacional simulados utilizando uma modulação mais simples, a modulação QPSK. Os parâmetros definidos para codificação e modulação no experimento estão na Tabela 6.

Tabela 6 – Parâmetros para codificação e modulação.

Parâmetro	Valor
Tamanho da mensagem	2560 bits
Gráfico base (p/ LDPC)	2
Número de <i>filler</i> bits (p/ LDPC)	0 bits
Número máx. de iterações no decodificador (p/ LDPC)	6
Número de bits por símbolo	2 (mod. QPSK)

Fonte: Elaborado pela autora.

O diagrama de blocos da Figura 20 foi utilizado para validar a integração entre o GNU Radio e o MATLAB. O GRC indica com cores o tipo das entradas e saídas dos seus blocos. Para esse trabalho são utilizados os tipos byte, ou int8 (representado pela cor rosa), *complex float 32* (em azul), e *float 32* (em laranja). Em resumo, rosa indica as portas com vetores de bits e azul as portas com vetores de símbolos. Na imagem está sendo utilizado o código LDPC, mas basta alterar o código Python dos blocos “*Channel Coding (MATLAB)*” e “*Channel Decoding (MATLAB)*” para utilizar os códigos convolucionais.

Tabela 7 – Parâmetros para rádio USRP.

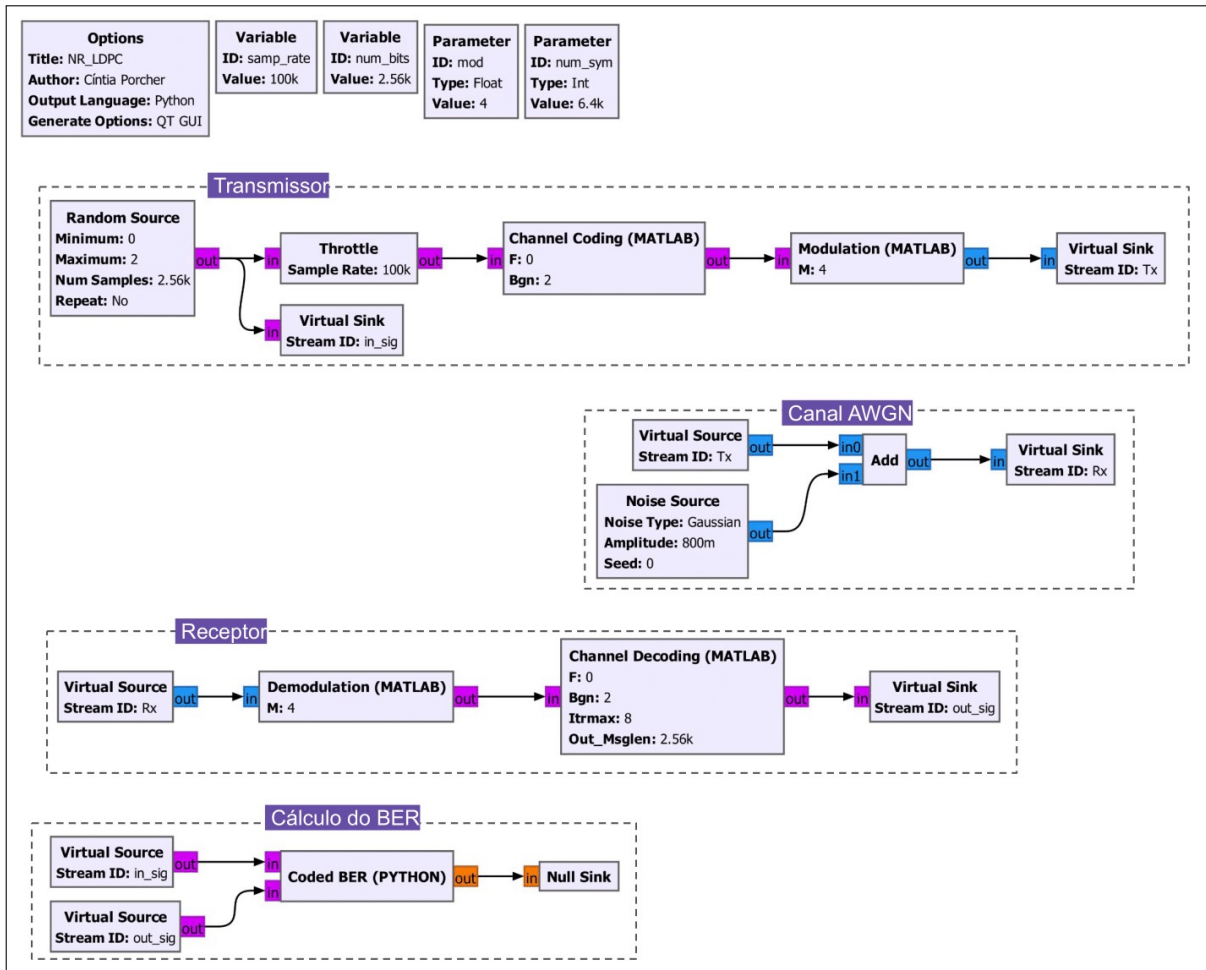
Parâmetro	Valor
Ganho	de 0 a 10 dB
Frequência	900 MHz
Largura de banda	1 MHz
Taxa de amostragem	200 kSps

Fonte: Elaborado pela autora.

Com a integração entre os softwares validadas, se substitui o bloco final do transmissor (o bloco “*Virtual Sink*” de *Stream ID Tx*) e o inicial do receptor (o bloco “*Virtual*

⁴ Disponível em: <https://www.mathworks.com/support/requirements/python-compatibility.html>. Acesso em: 12 de fevereiro de 2023.

Figura 20 – Diagrama de blocos do GRC para validação.



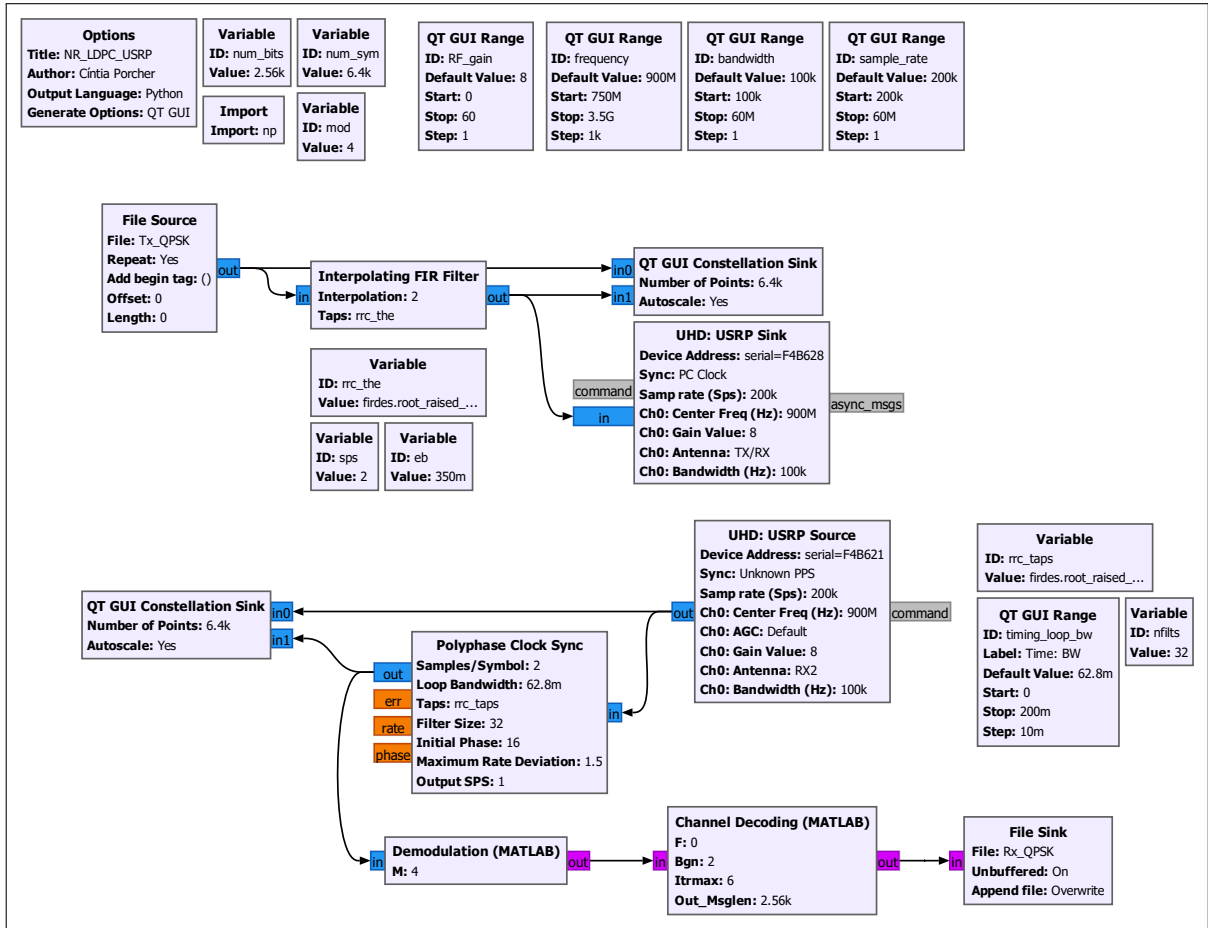
Fonte: Elaborado pela autora.

Source” de *Stream ID* Rx) por blocos “UHD: USRP Sink” e “UHD: USRP Source”. Para os blocos do USRP, devem ser definidos os parâmetros físicos a serem utilizados, como ganho, frequência e largura de banda, mostrados na Tabela 7. Os blocos representando o canal são removidos, visto que, com o experimento, se utiliza um canal real. O diagrama de blocos resultante está ilustrado na Figura 21.

Algumas alterações foram feitas da Figura 20 para a 21 para possibilitar o funcionamento no sistema na prática: a mensagem codificada foi adicionada como um arquivo e foram adicionados filtros no transmissor e no receptor. Os blocos que geravam a mensagem codificada e modulada foram substituídos pelo bloco “*File Sink*”, dessa forma se possibilita o funcionamento com repetição da mensagem. O arquivo utilizado como a mensagem “*Tx_QPSK*” é gerado pelo *flowgraph* da Figura 22.

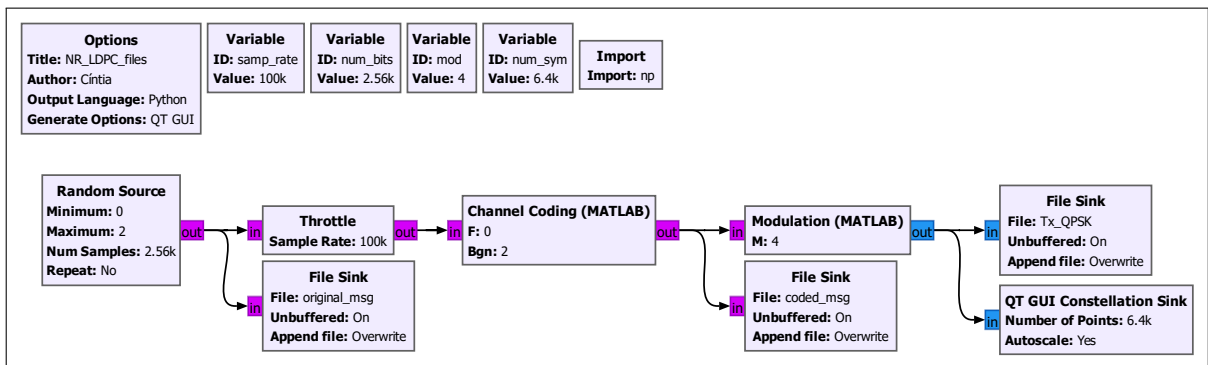
Um filtro FIR (do inglês, *Finite Impulse Response*) foi adicionado na saída do transmissor para definir a resposta ao impulso do sinal, o fator de interpolação foi definido como 2 e o filtro é do tipo *root-raised cosine* com um ganho unitário, uma frequência de amostragem sendo o dobro da largura de banda e um fator de *roll-off* de 0,35.

Figura 21 – Diagrama de blocos do GRC para realização do experimento utilizando o rádio USRP com a adição do bloco “Polyphase Clock Sync”.



Fonte: Elaborado pela autora.

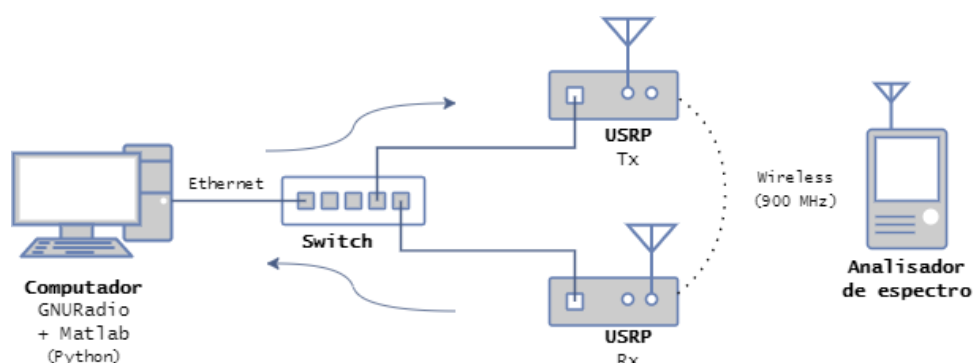
Figura 22 – Diagrama de blocos do GRC para gerar arquivos com a mensagem codificada e modulada.



Fonte: Elaborado pela autora.

Já na recepção foi utilizado um bloco sincronizador de tempo chamado “*Polyphase Clock Sync*” para evitar a interferência intersimbólica, fenômeno que pode acontecer em um canal real e que não foi considerado na simulação. A construção do bloco é baseada no artigo de Harris e Rice (2001), onde foi desenvolvido um banco de filtros polifásicos para sincronização em SDR. Esse bloco é sugerido por um dos tutoriais do GNU Radio para evitar problemas de sincronização e os parâmetros do bloco foram definidos conforme esse guia (GNURADIO, 2021).

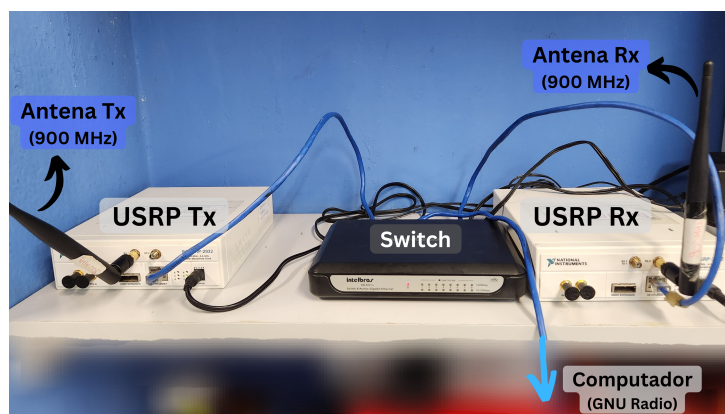
Figura 23 – Diagrama do experimento utilizando antenas.



Fonte: Elaborado pela autora.

Com o experimento realizado em um ambiente restrito, se passa para a segunda parte, onde são utilizadas antenas. A estrutura utilizada no experimento está ilustrada na Figura 23. Diferentemente do diagrama da Figura 18, além das antenas, nessa parte são utilizados dois rádios, um como transmissor e outro como receptor. Além disso, foi utilizado um analisador de espectro para verificar a emissão dos sinais e a ocupação da banda. A fotografia da Figura 24 mostra o *setup* experimental montado com alguns dos principais componentes.

Figura 24 – Fotografia da montagem do sistema, mostrando os componentes da Figura 23.



Fonte: Elaborado pela autora.

Para possibilitar o funcionamento com os dois USRP e as antenas, foi adicionado um bloco de processamento do sinal no receptor, conforme o diagrama da Figura 25. O

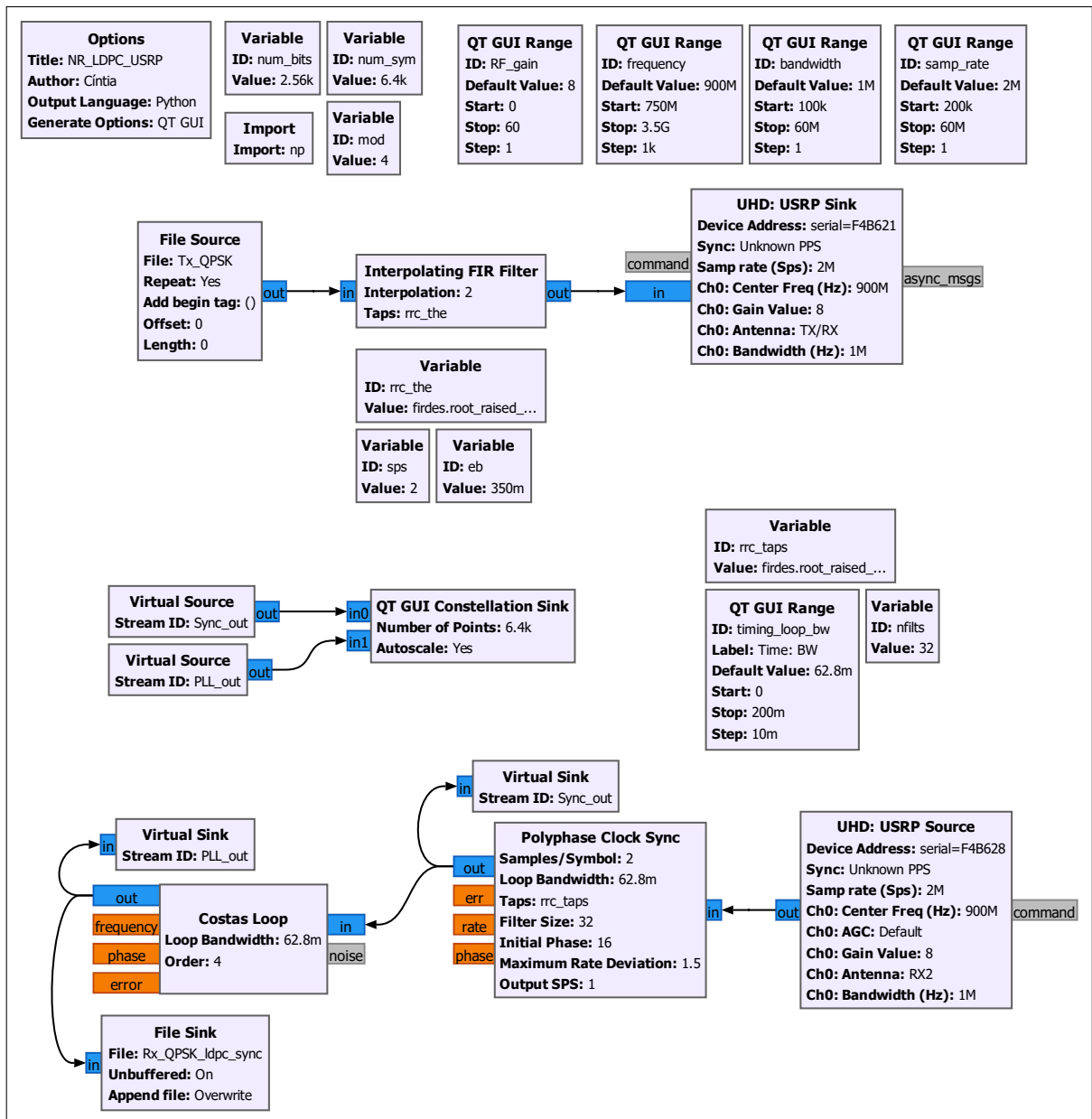
bloco Costas Loop é um tipo de PLL (do inglês, *Phase-Locked Loop*) e ele foi utilizado para alinhar a fase dos símbolos I e Q. O PLL possui dois parâmetros a serem definidos no GNU Radio, a largura de banda do *loop*, definida como $2\pi/100$, e a ordem do loop, definida conforme a modulação. Para QPSK a ordem do loop é quatro e, como Costas Loop foi desenvolvido para modulações do tipo PSK, ele não prevê modulações do tipo QAM.

Os blocos de demodulação e decodificação foram retirados nessa parte para que esses processos sejam realizados posteriormente, evitando problemas gerados pela repetição da mensagem do transmissor. Para isso, foi salvo um arquivo (“Rx_QPSK_ldpc_sync” ou “Rx_QPSK_conv_sync”) com o vetor da saída do bloco Costas Loop e em seguida esse vetor salvo foi lido e demodulado em um código em Python utilizando a API do MATLAB.

A mensagem no receptor é uma repetição da mensagem do arquivo “Tx_QPSK”, conforme o diagrama da Figura 25, e para poder decodificar essa mensagem é preciso identificar os pontos onde ela começa. Para isso foi feita uma correlação entre a mensagem codificada enviada (salva no arquivo “coded_msg” do diagrama da Figura 22) e a mensagem recebida demodulada. Com os pontos de repetição da mensagem identificados, foi realizado o processo de decodificação para em seguida ser calculado o BER.

Os códigos desenvolvidos no MATLAB para simulação e em Python para os blocos do GNU Radio estão disponibilizados no GitHub, o link se encontra no Apêndice C.

Figura 25 – Diagrama de blocos do GRC para realização do experimento com a adição do bloco Costas Loop.



Fonte: Elaborado pela autora.

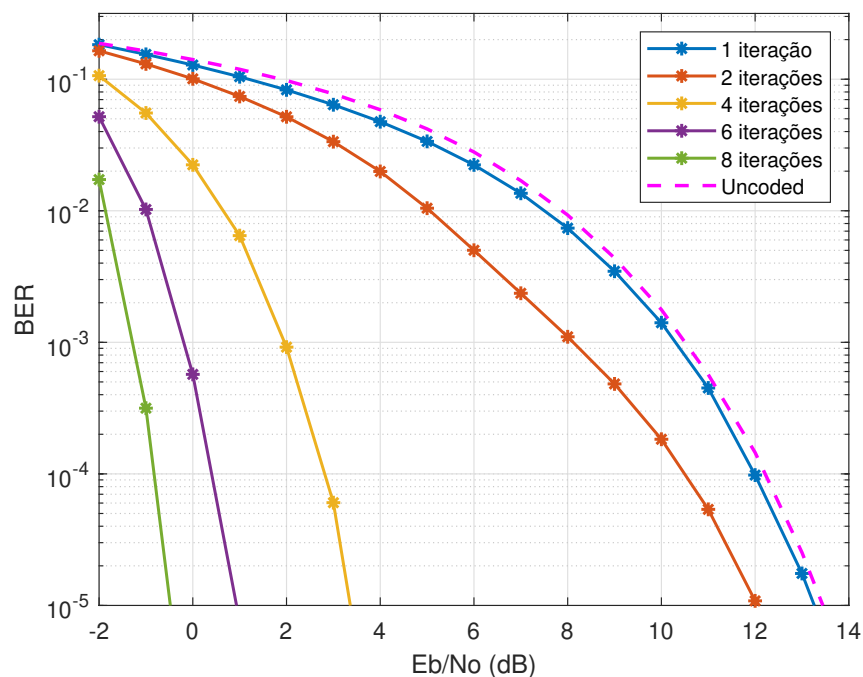
4 Resultados

Os resultados da simulação e do experimento são apresentados e discutidos neste capítulo.

4.1 Simulação

A simulação do sistema de comunicação foi inicialmente realizada no MATLAB utilizando o código NR-LDPC de acordo com o TS 38.212. Em um primeiro momento foi realizado um estudo do impacto do número máximo de iterações do decodificador, para em seguida realizar a simulação com outros tipos de modulação.

Figura 26 – Evolução do BER com as iterações do código LDPC com modulação 16-QAM.



Fonte: Elaborado pela autora.

A Figura 26 mostra aumento do desempenho do decodificador do receptor com o aumento do número de iterações. A simulação foi feita com os parâmetros da Tabela 4, com modulação 16-QAM, uma constelação intermediária, e diversos valores de iterações máximas do decodificador (1, 2, 4, 6 e 8), também foi calculada a taxa de erros sem codificação. Com apenas uma iteração o desempenho é semelhante ao caso sem codificação, não foram notadas melhorias significativas. Com o avanço do número de iterações o desempenho é cada vez melhor, mas se espera que o processo de decodificação também se torne mais lento. Para definir um número de iterações o ideal é realizar uma análise do

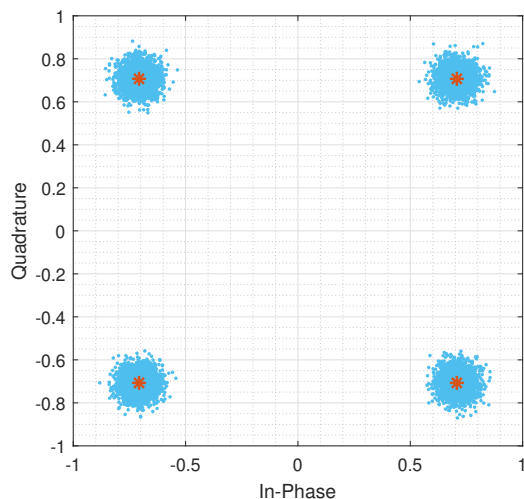
custo computacional do processo, porém, como esse estudo não foi realizado, foi definido, apenas com a interpretação dos resultados do gráfico da Figura 26, que o número máximo de iterações do decodificador será seis, visando uma boa relação entre a capacidade de correção de erros e o tempo despendido.

As constelações das modulações utilizadas estão ilustradas nas Figuras 27a, 27b, 27c e 27d, esses são os quatro tipos de modulação definidos para as redes móveis de última geração. A potência média dos sinais foi definida como sendo de 1 W, ou 30 dBm, e o SNR 24 dB. Essas constelações seguem o código de Gray, portanto os símbolos “vizinhos” se diferenciam por apenas um bit, tornando a mensagem mais robusta. Os asteriscos em laranja indicam os símbolos sem ruído na saída do transmissor, e os pontos em azul os símbolos com um ruído adicionado, na entrada no receptor. Com essas imagens é possível notar que uma modulação de maior grau carrega mais informação, mas é mais suscetível ao ruído, por isso a modulação é feita de forma adaptativa, variando a taxa conforme o nível de ruído.

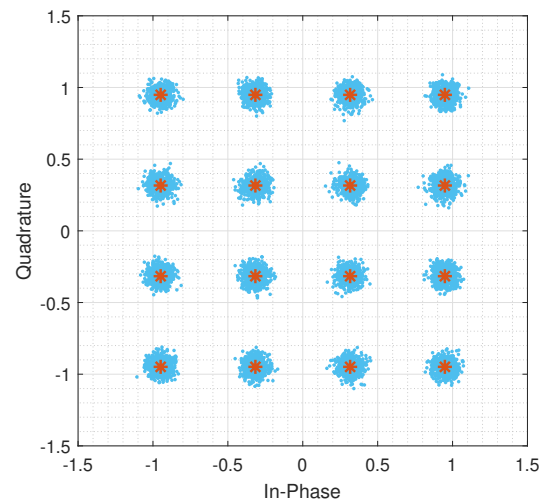
Com o número de iterações e as diferentes modulações analisadas, é avaliado o desempenho do sistema completo variando esses parâmetros em conjunto. O desempenho do sistema foi comparado com os valores teóricos de BER para sistemas sem codificação em um canal AWGN. Os parâmetros da simulação também estão definidos na Tabela 4 e o número máximo de iterações foi definido como sendo seis, visto que quanto maior o número de iterações, maior é o tempo de processamento dos dados. O resultado está ilustrado no gráfico da Figura 28.

A partir das curvas da Figura 28 nota-se que um sinal com uma taxa de modulação menor, como QPSK, é menos suscetível a erros. A mensagem quando quando codificada

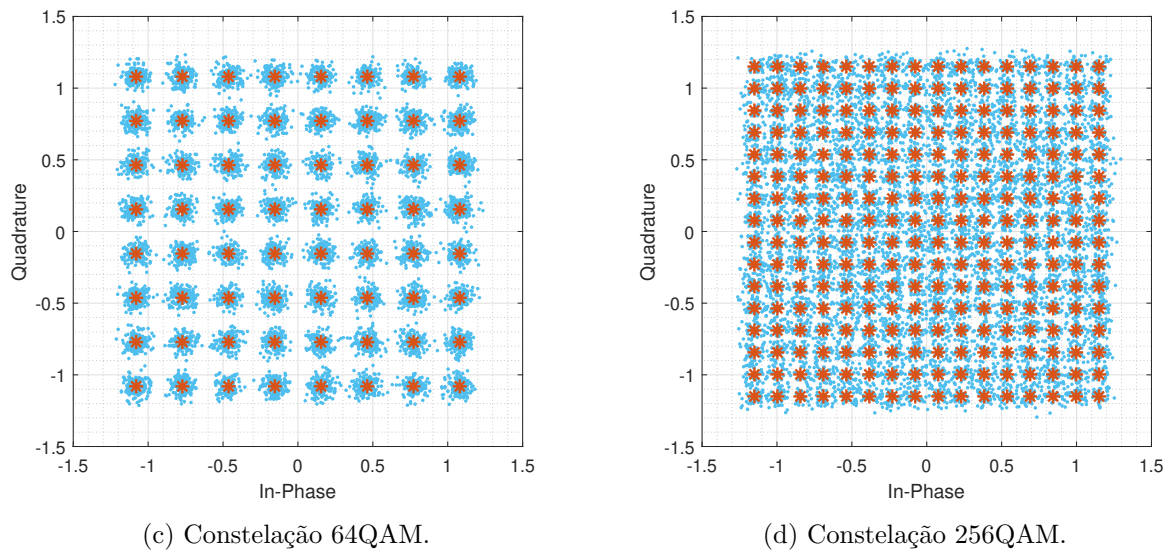
Figura 27 – Constelações dos sinais antes (em laranja) e após (em azul) um canal AWGN com um SNR de 24 dB.



(a) Constelação QPSK.



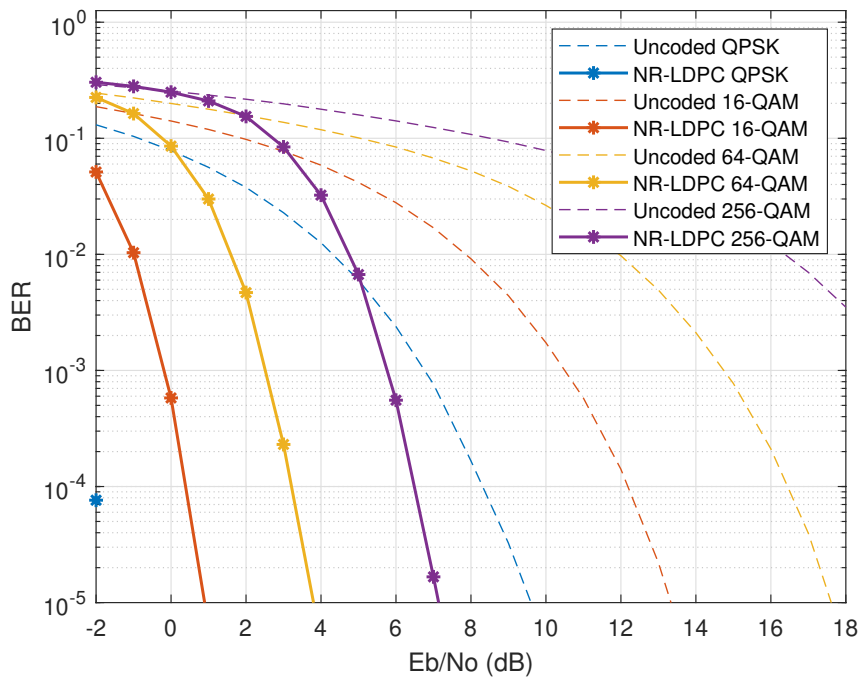
(b) Constelação 16QAM.



Fonte: Elaborado pela autora.

e modulada em QPSK só apresentou erros para relação sinal/ruído de -2 dB , com um BER de $7,625 \cdot 10^{-5}$, para os outros valores de E_b/N_0 o decodificador foi capaz de corrigir todos os erros. Comparando os desempenhos, para um BER de 10^{-2} a diferença entre caso codificado e sem codificação com modulação 16-QAM foi de aproximadamente 9 dB , para 64-QAM foi $\approx 10\text{ dB}$ e para 256-QAM foi $\approx 12\text{ dB}$.

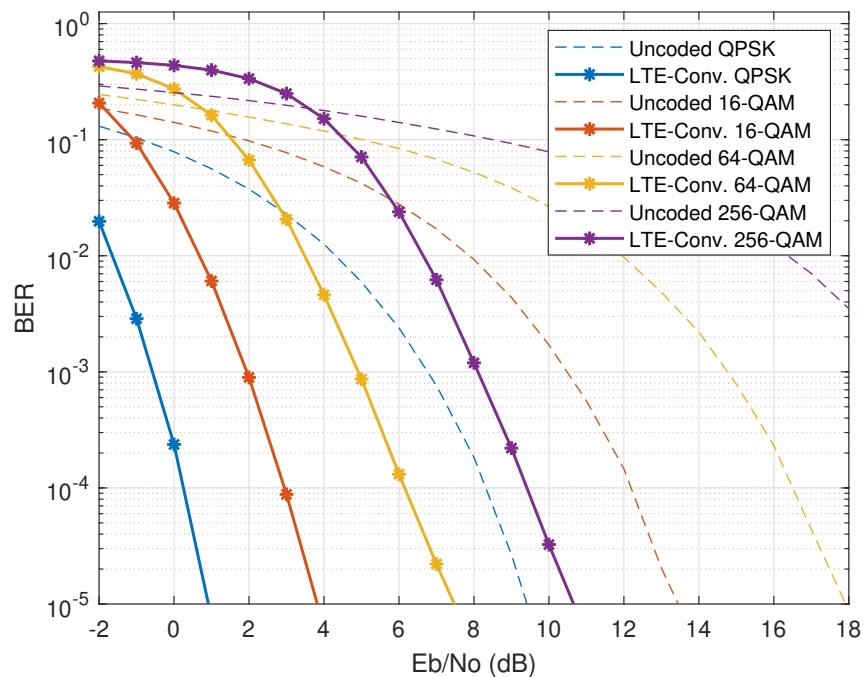
Figura 28 – Comparação do BER com codificação LDPC e sem codificação para diversos esquemas de modulação.



Fonte: Elaborado pela autora.

Para o código convolucional LTE de acordo com o TS 36.212, com os parâmetros da Tabela 5 e diversas modulações, o resultado está ilustrado na Figura 29. Comparando os desempenhos, para um BER de 10^{-2} a diferença entre caso codificado e sem codificação com modulação QPSK foi de aproximadamente 6 dB, para 16-QAM foi ≈ 8 dB, para 64-QAM foi ≈ 9 dB e para 256-QAM foi ≈ 10 dB. Por esses valores já é possível perceber que o desempenho dos códigos LDPC foi superior aos códigos convolucionais.

Figura 29 – Comparação do BER com codificação Convolucional e sem codificação para diversos esquemas de modulação.

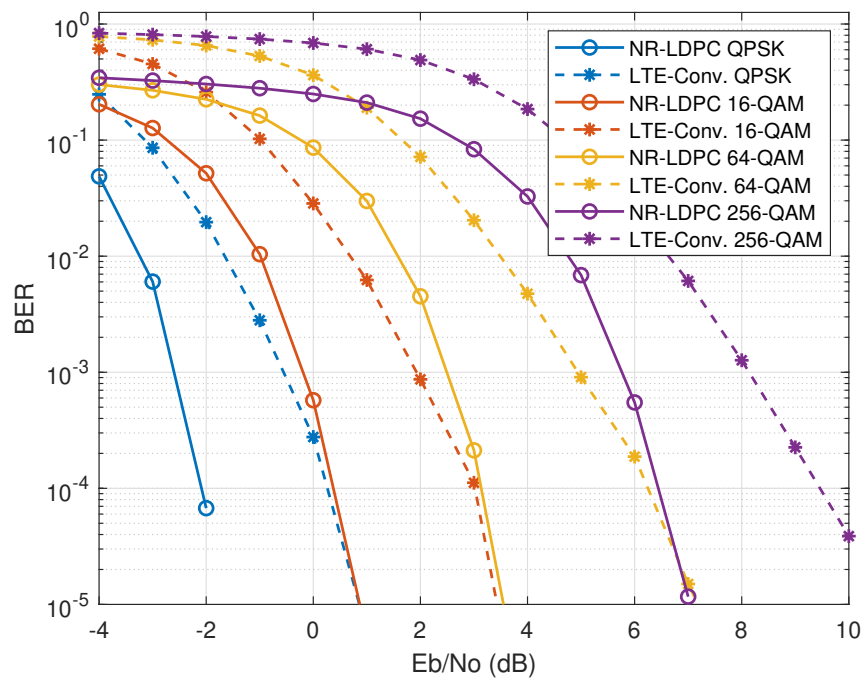


Fonte: Elaborado pela autora.

Para evidenciar a comparação dos códigos, as curvas foram plotadas juntas no gráfico da Figura 30. Ressalta-se que foram utilizadas seis iterações para decodificação dos códigos LDPC, com um valor menor o desempenho se aproxima dos códigos convolucionais, podendo ser até inferior. Já com mais iterações a diferença entre os desempenhos é ainda mais evidente.

Esses resultados parciais evidenciam a importância do uso de codificação do canal, visto que ela é capaz de aumentar significativamente a eficiência do sistema de comunicação. Os resultados também mostram a diferença entre as performances dos dois tipos de código, evidenciando a maior flexibilidade e, com os parâmetros utilizados, maior capacidade dos códigos LDPC.

Figura 30 – Comparação do BER com codificação LDPC e com codificação Convolutacional.



Fonte: Elaborado pela autora.

4.2 Experimento

Para validar os resultados da simulação foi realizada a implementação do sistema em SDR. Inicialmente foi testada a integração entre o GNU Radio e o MATLAB, utilizando o fluxograma da Figura 20. Para um ruído gaussiano complexo de desvio padrão 0,6 e uma modulação 16-QAM a constelação na saída do transmissor e na entrada do receptor é representada na Figura 31.

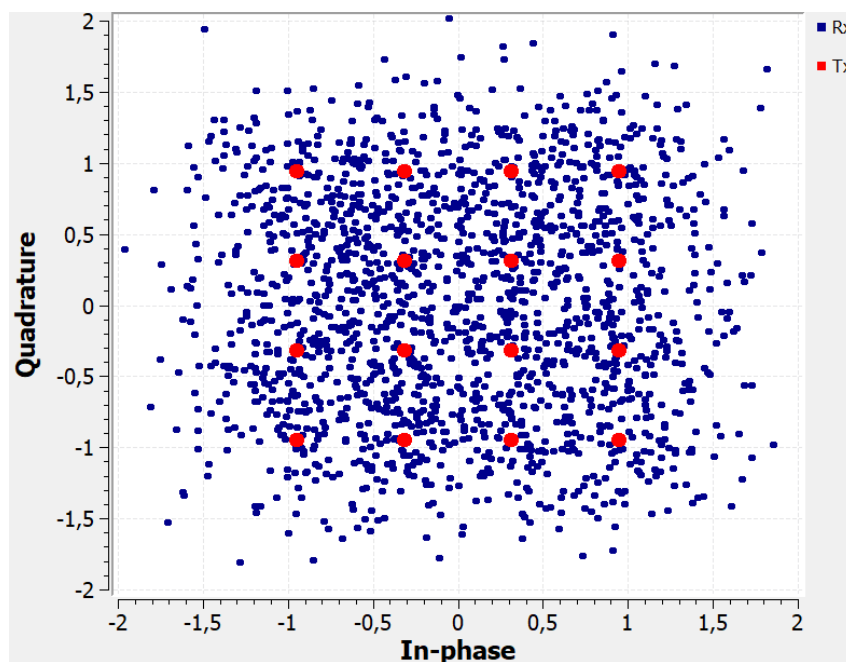
Com 10 repetições e até seis iterações no caso LDPC, a mensagem recebida, depois de ser demodulada e decodificada, apresentou uma média de 0,40 erros em 2560 bits, resultando em um BER de 0,016%. A codificação convolutacional, com o mesmo número de repetições e com 2400 bits, apresentou em média 42,70 erros em 2400 bits, resultando em um BER de 1,779%. Já a mensagem sem codificação apresentou em média 305,90 erros e um BER de 12,746%. Esses resultados estão detalhados na Tabela 8.

Tabela 8 – Taxas de erros da simulação no GNU Radio.

Codificação	Média de erros	Desvio padrão	BER
LDPC	0,400 (10 repetições)	0,490	0,016%
Convolutacional	42,700 (10 repetições)	10,179	1,779%
Sem codificação	305,900 (10 repetições)	7,134	12,746%

Fonte: Elaborado pela autora.

Figura 31 – Constelações produzidas pela simulação no GRC antes (em vermelho) e após (em azul) passar pelo canal AWGN com um desvio padrão de 0,6.



Fonte: Elaborado pela autora.

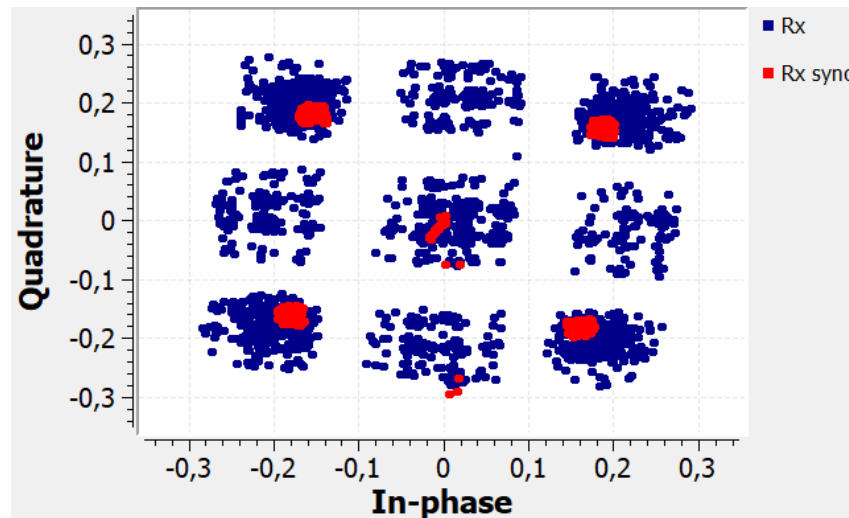
Com a integração do GNU Radio e do MATLAB validada, se utilizou os rádios USRP para transmitir a mensagem em um canal real, conforme o diagrama da Figura 23 e o fluxograma da Figura 21. Em um primeiro momento o experimento com o rádio foi feito com o auxílio de um cabo e de um atenuador, conforme o diagrama da Figura 18 e da fotografia da Figura 19.

A Figura 32 apresenta a constelação recebida antes e depois da sincronização. A constelação recebida (em azul) apresenta um padrão diferente do esperado para uma modulação QPSK. Era esperado receber os símbolos centralizados em quatro pontos, mas ao invés disso foram recebidos símbolos divididos em setores que se alteravam com o tempo e que apresentavam algum padrão, o que indica a possibilidade de *aliasing* ou interferência entre símbolos. Essas condições não foram consideradas nas simulações, onde foi utilizado um canal AWGN.

Para compreender o problema foram testadas diversas condições e a melhor solução encontrada foi o processamento do sinal pelo bloco “*Polyphase Clock Sync*”. O bloco se trata de sincronizador de tempo que usa um banco de filtros polifásicos. O resultado do sinal processado por esse bloco são os símbolos em vermelho da Figura 32. Os símbolos ficaram essencialmente divididos em quatro pontos, como era esperado de uma constelação QPSK. Com isso nota-se que se tratava de um problema de sincronização causando a interferência entre os símbolos.

O experimento realizado com o cabo e o atenuador foi repetido, agora com o uso de antenas, conforme o diagrama da Figura 23 e da fotografia da Figura 24. O atenuador

Figura 32 – Constelação recebida antes (em azul) e após (em vermelho) do processo de sincronização.

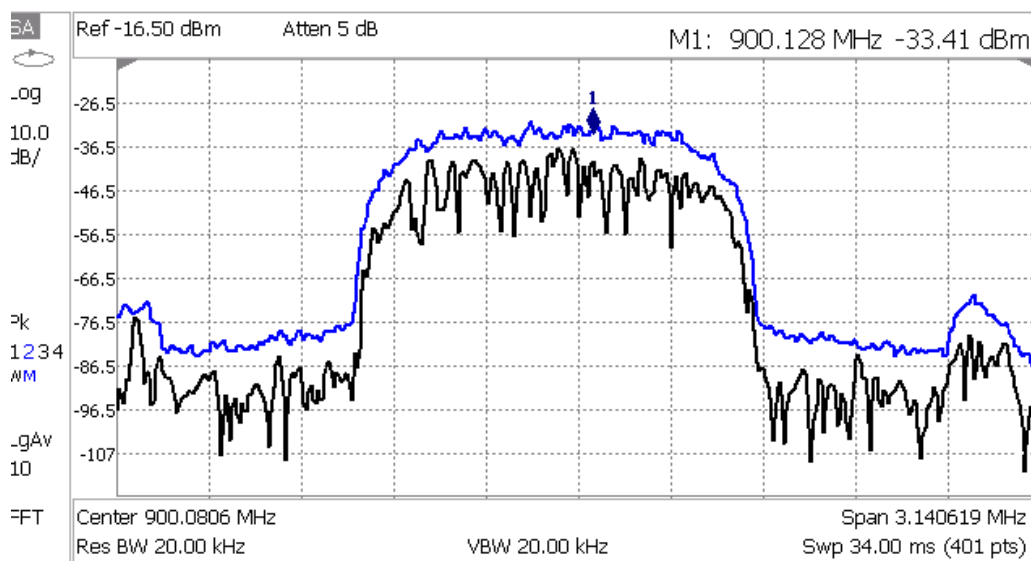


Fonte: Elaborado pela autora.

utilizado com apenas um USRP foi dispensado, já com as antenas passou-se a utilizar dois URSP, um como transmissor e outro como receptor.

O sinal foi medido através do analisador de espectro e foram obtidos os gráficos das Figuras 33 e 34. A Figura 33 apresenta o espectro de frequências centralizado em 900 MHz, com a curva em azul com os valores máximos ao longo do período de obtenção dos dados e a curva em preto com os valores atuais. A Figura 34 apresenta medidas do sinal, sendo elas uma largura de banda de 1,15 MHz, próxima ao valor de 1 MHz selecionado pelo software, e uma potência de -25,52 dBm.

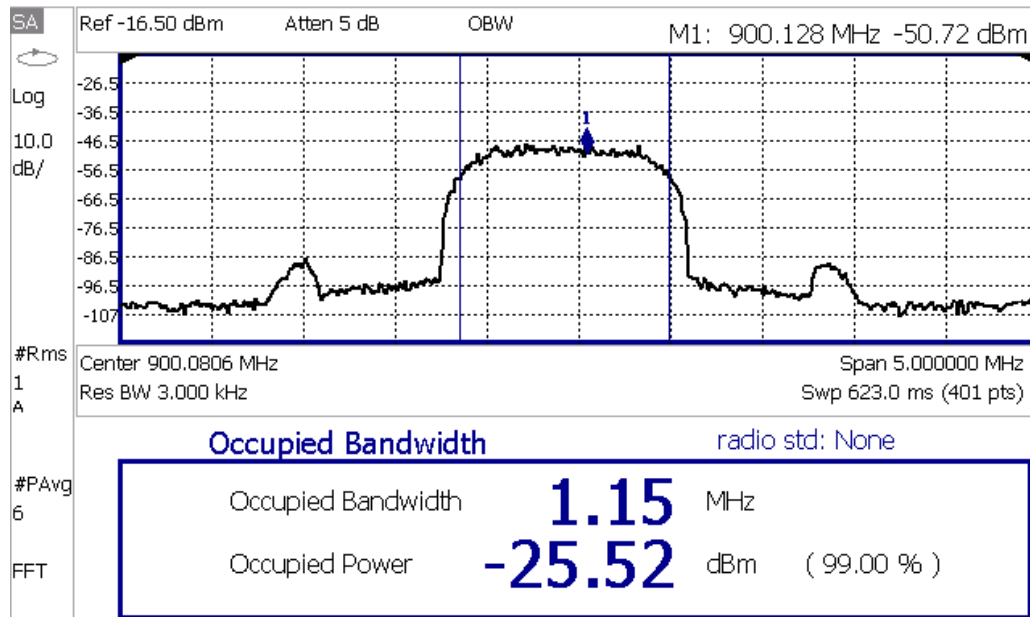
Figura 33 – Espectro do sinal utilizado para o experimento.



Ref Level: -16.5 dBm

Fonte: Elaborado pela autora.

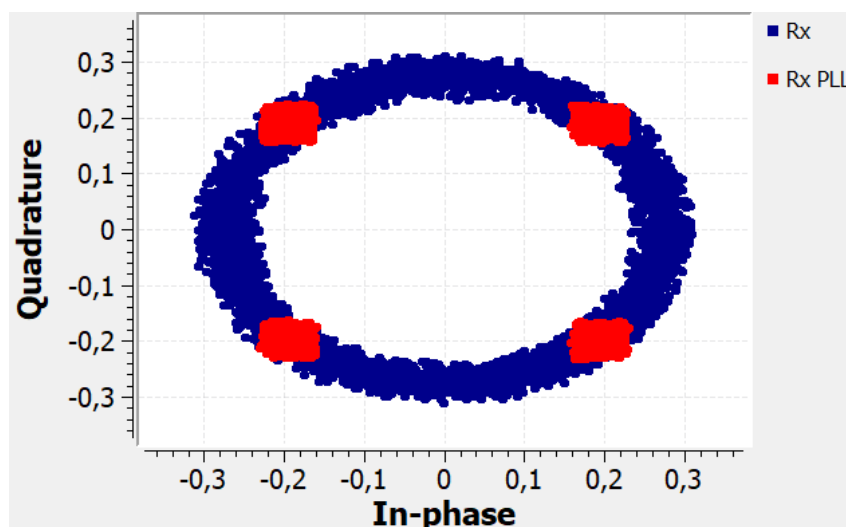
Figura 34 – Espectro do sinal com medidas de largura de banda e potência.



Fonte: Elaborado pela autora.

A constelação resultante do uso de antenas está na Figura 35. Em azul estão os símbolos na saída do bloco “*Polyphase Clock Sync*”, nota-se que esses símbolos estão rotacionando ao redor da origem e para ajustar esse comportamento é necessário adicionar mais um bloco de processamento do sinal. O bloco utilizado é um Costas Loop e o experimento foi feito conforme o fluxograma da Figura 25. Conforme mencionado anteriormente, esse bloco é um PLL que corrige a fase dos símbolos, resultando na constelação em vermelho da Figura 35.

Figura 35 – Constelação QPSK recebida antes (em azul) e após (em vermelho) o PLL.

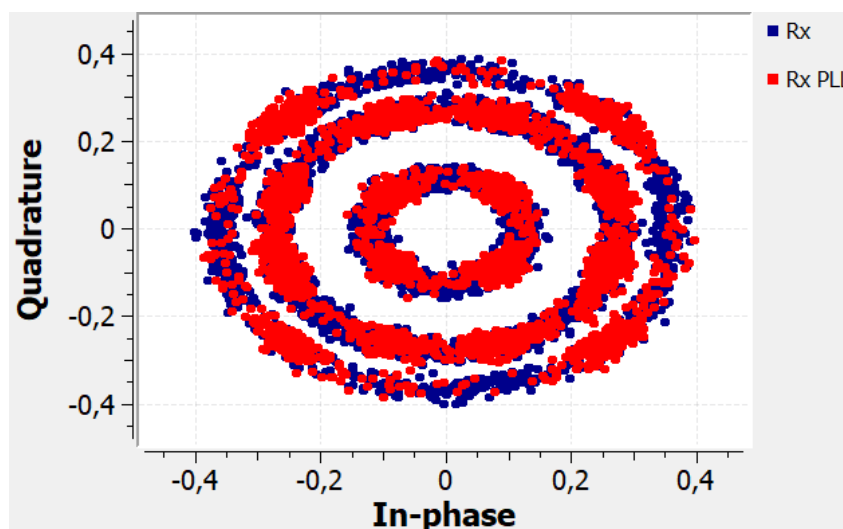


Fonte: Elaborado pela autora.

O resultado do experimento para modulações QAM está na Figura 36. O bloco “*Polyphase Clock Sync*” funcionou como esperado, porém o PLL não corrigiu a fase dos

símbolos de forma efetiva para esse esse tipo de modulação. O Costas Loop foi desenvolvido para modulações PSK, portanto ele não prevê modulação em amplitude. Para realizar o experimento de forma efetiva com modulações mais altas é necessário utilizar outro modelo do PLL.

Figura 36 – Constelação 16-QAM recebida antes (em azul) e após (em vermelho) o PLL.



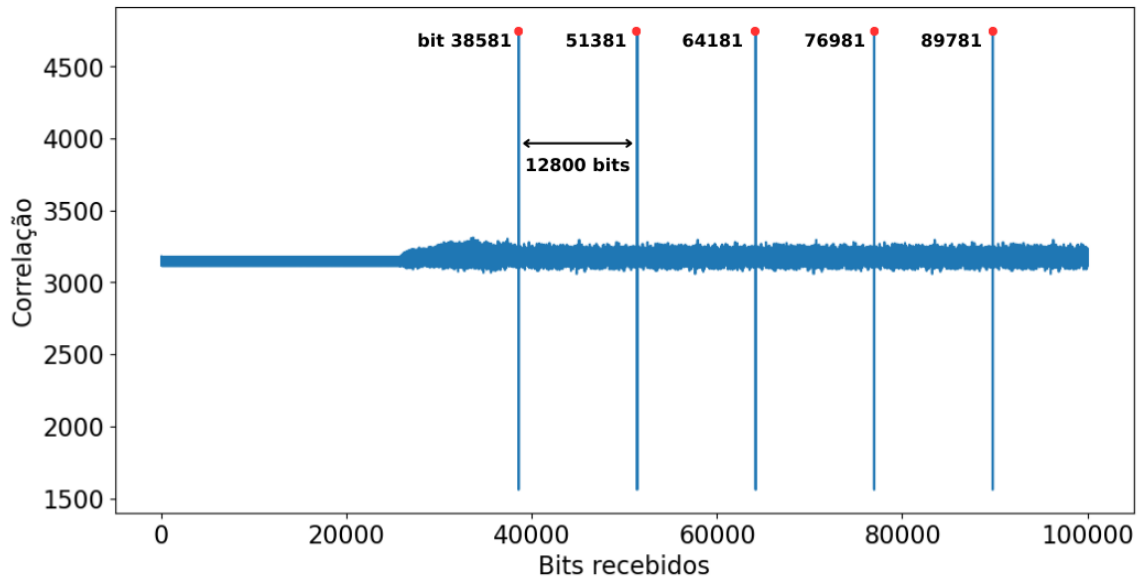
Fonte: Elaborado pela autora.

Para o estudo do desempenho dos códigos corretores é essencial o cálculo das taxas de erros, porém essa análise não foi bem sucedida na parte experimental utilizando o GNU Radio. O BER foi corretamente calculado utilizando o diagrama da Figura 20, onde a mensagem não era repetida, mas para os testes seguintes a mensagem era repetida e os blocos do receptor de demodulação e decodificação deixaram de funcionar. Para tornar possível o cálculo da taxa de erros foi salva a mensagem na saída do bloco Costas Loop e a mensagem foi demodulada e decodificada em um código em Python.

Os vetores de símbolos salvos em arquivos como “Rx_QPSK_ldpc_sync” e “Rx_QPSK_conv_sync” foram inteiramente demodulados, se transformando em vetores de bits ainda codificados. Para decodificação foi necessário identificar os pontos onde as repetições das mensagens iniciavam e para isso foi feita uma correlação entre a mensagem codificada enviada e a mensagem recebida demodulada. O resultado dessa operação está ilustrado no gráfico da Figura 37 para o código LDPC e da Figura 38 para o código convolucional.

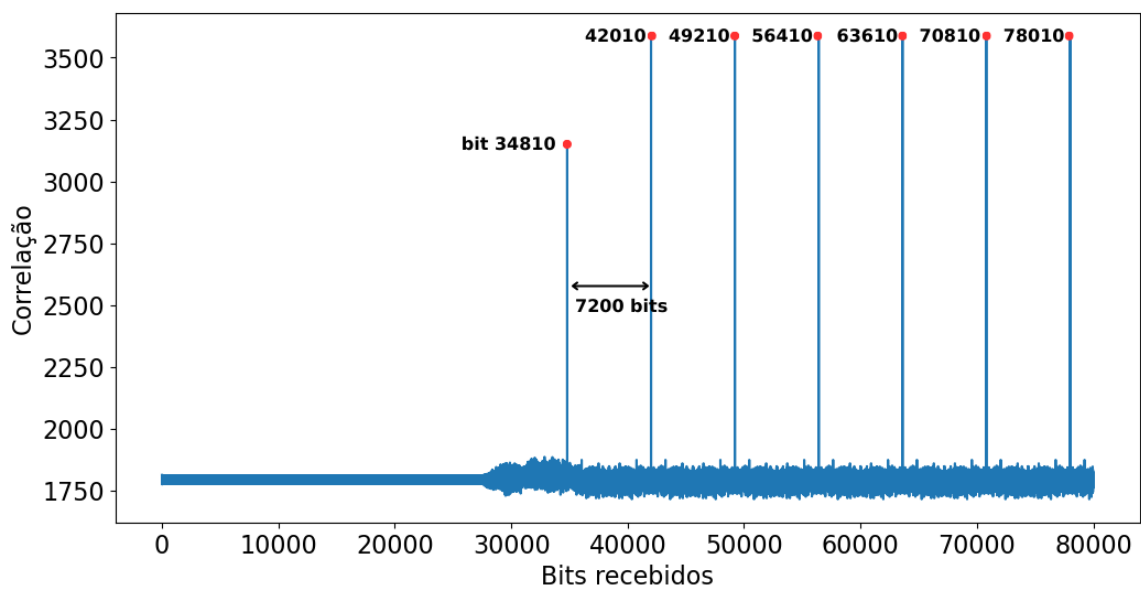
Os primeiros bits recebidos possuem uma baixa correlação quando comparados aos demais e analisando módulo dos símbolos recebidos, presente na Figura 39, fica evidente que esses primeiros símbolos são apenas ruído, visto que o módulo deles é inferior a 0,1 e após aproximadamente 17 mil símbolos o módulo passa a ter uma média de aproximadamente 0,4. Voltando a analisar o resultado da correlação entre as mensagens, após esses primeiros símbolos que representam ruídos, a distância entre os picos da correlação se torna constante,

Figura 37 – Correlação entre a mensagem com codificação LDPC enviada e a mensagem recebida demodulada.



Fonte: Elaborado pela autora.

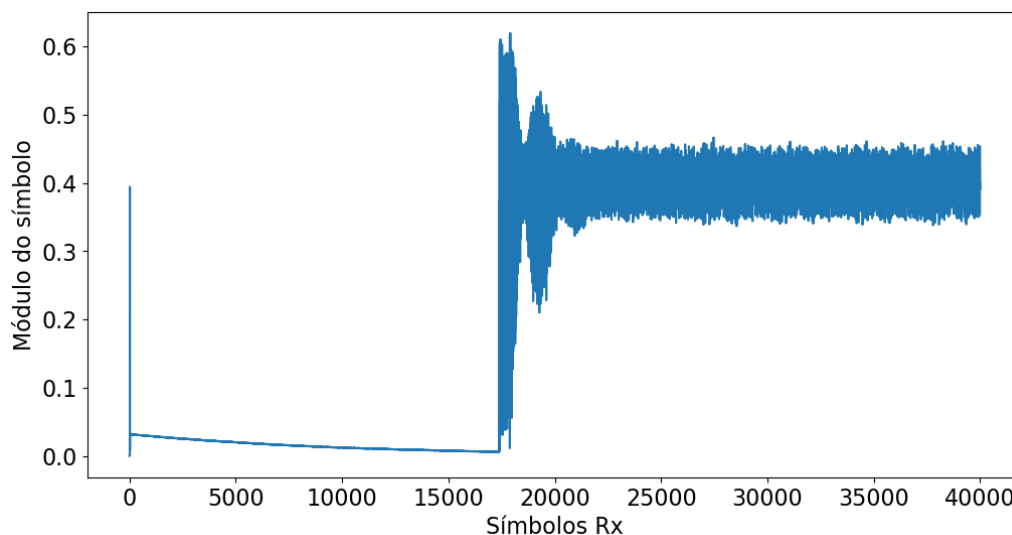
Figura 38 – Correlação entre a mensagem com codificação convolucional enviada e a mensagem recebida demodulada.



Fonte: Elaborado pela autora.

sendo de 12800 bits para codificação LDPC e 7200 para codificação convolucional. Esses valores estão de acordo com o esperado, visto que para LDPC foram codificados 2560 bits com uma taxa de 1/5, resultando em 12800 bits, e para convolucional foram codificados 2400 bits com uma taxa de 1/3, resultando em 7200 bits.

Figura 39 – Módulo dos símbolos recebidos.



Fonte: Elaborado pela autora.

Com os pontos de repetição da mensagem identificados, os vetores foram divididos em sub mensagens e foi feita a decodificação. As taxas de erros resultantes estão na Tabela 9. Para codificação LDPC se obteve um BER de 20,745% e para convolucional 0,018%.

Tabela 9 – Taxas de erros do experimento.

Codificação	Média de erros	BER
LDPC (com 97 repetições)	20,12	20,745%
Convolucional (com 94 repetições)	0,45	0,018%

Fonte: Elaborado pela autora.

Considerando que o ruído presente no experimento era baixo, como ficou evidente pela constelação das Figuras 32 e 35 e pela amplitude dos símbolos iniciais do gráfico da Figura 39, era esperado um BER praticamente nulo. Por conta do BER elevado para codificação LDPC, se conclui que ocorreu algum problema no processo de decodificação que não foi identificado. A diferença entre o comportamento das correlações das Figuras 37 e 38 indica alguma diferença no processo, visto que para o código LDPC há pontos com correlação mínima abaixo da média.

5 Conclusões

Este trabalho apresentou um estudo de técnicas de correção de erros voltadas para redes móveis modernas (4G LTE e 5G NR). Foram analisadas as capacidades de correção dos códigos LDPC e dos códigos convolucionais, implementados conforme parâmetros definidos pelo 3GPP, através de simulações e experimentos. A análise experimental foi modelada, mas não foi possível analisar os resultados da codificação efetivamente, deixando-se a finalização dessa etapa experimental do projeto como uma das sugestões de trabalhos futuros.

A partir dos resultados obtidos é possível concluir que as duas codificações estudadas geram uma significativa redução das taxas de erros. Também foi possível analisar o impacto de diferentes modulações digitais. Uma modulação com menos símbolos, como a QPSK, é menos suscetível à erros. Com o aumento do número de símbolos, o ruído do canal tende a gerar mais erros. Com isso se conclui que constelações com mais símbolos (como a 256-QAM que possui 256 símbolos e cada um carrega 8 bits) são adequadas para cenários menos suscetíveis ao ruído. Modulações mais elevadas transmitem mais bits, atingindo maiores taxas de dados, portanto a suscetibilidade ao ruído é um dos *trade-offs* de maiores velocidades de transferência de dados.

A codificação LDPC com matrizes definidas pelo padrão 3GPP TS 38.212 (desenvolvido para tecnologia 5G *New Radio*) e com seis iterações no decodificador, apresentou uma capacidade de correção de erros superior aos códigos convolucionais definidos pelo padrão 36.212 (desenvolvido para tecnologia LTE).

5.1 Trabalhos Futuros

Para trabalhos futuros sugere-se a complementação das análises experimentais. Neste trabalho foi utilizada correlação para identificar os pontos de repetição da mensagem enviada e assim possibilitar a decodificação. Os resultados para codificação convolucional foram conclusivos, mas para codificação LDPC apresentaram valores muito elevados e que não estavam de acordo com o baixo ruído presente no canal, indicando algum erro que não foi identificado.

Foram estudados os códigos LDPC designados para os canais de dados 5G NR e os códigos convolucionais para canais de controle LTE. Adicionalmente, é importante realizar o estudo dos códigos polares para canais de controle 5G e os códigos turbo para canais de dados LTE, abrindo a possibilidade de um estudo das diferenças entre canais de dados e canais de controle.

A latência é um dos pilares das redes móveis modernas e essa análise pode ser aprofundada em trabalhos futuros. Análises tanto de custo computacional como de tempo de decodificação são indicadas para diferenciar as técnicas estudadas.

Por fim, outros modelos de canais, mais complexos, que considerem caminhos múltiplos, devido aos ambientes urbanos, e o efeito Doppler, devido à mobilidade, se aproximariam mais dos cenários típicos de redes móveis. Realizar a simulação com canais que considerem esses efeitos pode aproximar ainda mais os resultados da simulação aos resultados experimentais.

Referências Bibliográficas

- 3GPP. *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding*. 2018. (TS 36.212 version 15.2.1 Release 15). Disponível em: <https://www.etsi.org/deliver/etsi_ts/136200_136299/136212-15.02.01_60/ts_136212v150201p.pdf>. Acesso em: 21 jan. 2023.
- 3GPP. *5G NR; Multiplexing and channel coding*. [S.l.], 2020. (TS 38.212 version 16.2.0 Release 16). Disponível em: <https://www.etsi.org/deliver/etsi_ts/138200_138299-138212/16.02.00_60/ts_138212v160200p.pdf>. Acesso em: 21 jan. 2023.
- ANATEL. Leilão de espectro 5g. Fev 2021. Disponível em: <<https://www.gov.br/anatel/pt-br/assuntos/5G/leilao-de-espectro-5g>>. Acesso em: 18 fev. 2023.
- ARIKAN, E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, v. 55, n. 7, p. 3051–3073, 2009. Disponível em: <[doi:10.1109/TIT.2009.2021379](https://doi.org/10.1109/TIT.2009.2021379)>.
- BAE, J. H.; ABOTABL, A.; LIN, H.-P.; SONG, K.-B.; LEE, J. An overview of channel coding for 5g nr cellular communications. *APSIPA Transactions on Signal and Information Processing*, Cambridge University Press, v. 8, p. e17, 2019. Disponível em: <[doi:10.1017/ATSIP.2019.10](https://doi.org/10.1017/ATSIP.2019.10)>.
- BERROU, C.; GLAVIEUX, A.; THITIMAJSHIMA, P. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In: *Proceedings of ICC '93 - IEEE International Conference on Communications*. [s.n.], 1993. v. 2, p. 1064–1070 vol.2. Disponível em: <[doi:10.1109/ICC.1993.397441](https://doi.org/10.1109/ICC.1993.397441)>.
- BEST, M.; BURNASHEV, M.; LEVY, Y.; RABINOVICH, A.; FISHBURN, P.; CALDERBANK, A.; COSTELLO, D. On a technique to calculate the exact performance of a convolutional code. *IEEE Transactions on Information Theory*, v. 41, n. 2, p. 441–447, 1995. Disponível em: <[doi:10.1109/18.370145](https://doi.org/10.1109/18.370145)>.
- CARLSON, A.; CRILLY, P. *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*. 5. ed. [S.l.]: McGraw-Hill Education, 2010. ISBN 978-0-07-338040-7.
- DHOLAKIA, A. *Introduction to Convolutional Codes with Applications*. 1. ed. [S.l.]: Springer Science+Business Media, 1994. ISBN 978-1-4615-2712-1.
- ELIAS, P. Coding for noisy channels. *IRE Conv. Rec.*, p. 37–46, 1955.
- ERICSSON. *Ericsson Mobility Report*. [S.l.], 2022. Disponível em: <<https://www.ericsson.com/4ae28d/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-november-2022.pdf>>. Acesso em: 18 jan. 2023.
- G1. 5G no Brasil: guia explica o que vai mudar com a nova tecnologia. *G1*, Jul 2022. Disponível em: <<https://g1.globo.com/tecnologia/noticia/2022/07/06/5g-chega-ao-brasil-nesta-quarta-guia-explica-o-que-vai-mudar-com-a-nova-tecnologia.ghtml>>. Acesso em: 3 mar. 2023.

- GALLAGER, R. Low-density parity-check codes. *IRE Transactions on Information Theory*, v. 8, n. 1, p. 21–28, 1962. Disponível em: <doi:10.1109/TIT.1962.1057683>.
- GNURADIO. Guided tutorial PSK demodulation. Mai 2021. Disponível em: <https://wiki.gnuradio.org/index.php?title=Guided_Tutorial_PSK_Demodulation>. Acesso em: 8 abr. 2023.
- HARRIS, F.; RICE, M. Multirate digital filters for symbol timing synchronization in software defined radios. *IEEE Journal on Selected Areas in Communications*, v. 19, n. 12, p. 2346–2357, 2001. Disponível em: <doi:10.1109/49.974601>.
- HUI, D.; SANDBERG, S.; BLANKENSHIP, Y.; ANDERSSON, M.; GROSJEAN, L. Channel coding in 5G New Radio: A tutorial overview and performance comparison with 4G LTE. *IEEE Vehicular Technology Magazine*, v. 13, n. 4, p. 60–69, 2018. Disponível em: <doi:10.1109/MVT.2018.2867640>.
- MACKAY, D. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, v. 45, n. 2, p. 399–431, 1999. Disponível em: <doi:10.1109/18.748992>.
- MIT. *Introduction to EECS II: Digital Communication Systems. Convolutional Coding*. 2012. Disponível em: <https://ocw.mit.edu/courses/6-02-introduction-to-eecs-ii-digital-communication-systems-fall-2012/resources/mit6_02f12_chap07/>. Acesso em: 25 fev. 2023.
- NGUYEN, T. T. B.; TAN, T. N.; LEE, H. Efficient QC-LDPC encoder for 5G new radio. *Electronics*, Multidisciplinary Digital Publishing Institute, v. 8, n. 6, p. 668, 2019. Disponível em: <doi.org/10.3390/electronics8060668>.
- NI. *USRP-2932 Specifications*. [S.l.], 2023. Disponível em: <https://www.ni.com/docs/en-US/bundle/usrp-2932-specs/page/specs.html>. Acesso em: 7 fev. 2023.
- OMAR, M. H.; EL-MAHMOUDY, A.; SEDDIK, K. G.; ELEZABI, A. On the tail-biting convolutional code decoder for the LTE and LTE-A standards'. In: *2013 Asilomar Conference on Signals, Systems and Computers*. [s.n.], 2013. p. 510–514. Disponível em: <doi:10.1109/ACSSC.2013.6810330>.
- PALENIK, T.; FARKAS, P.; RAKUS, M.; DOBOS, J. Analysis of minimal LDPC decoder system on a chip implementation. *Radioengineering*, v. 24, n. 3, p. 783, 2015. Disponível em: <doi:10.13164/re.2015.0783>.
- PATIL, M. V.; PAWAR, S.; SAQUIB, Z. Coding techniques for 5G networks: A review. In: IEEE. *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)*. 2020. p. 208–213. Disponível em: <doi:10.1109/CSCITA47329.2020.9137797>.
- RACHINGER, C.; HUBER, J. B.; MÜLLER, R. R. Comparison of convolutional and block codes for low structural delay. *IEEE Transactions on Communications*, v. 63, n. 12, p. 4629–4638, 2015. Disponível em: <doi:10.1109/TCOMM.2015.2488661>.
- RIAZ, A.; BANSAL, V.; SOLOMON, A.; AN, W.; LIU, Q.; GALLIGAN, K.; DUFFY, K. R.; MEDARD, M.; YAZICIGIL, R. T. Multi-code multi-rate universal maximum likelihood decoder using grand. In: *ESSCIRC 2021 - IEEE 47th European*

- Solid State Circuits Conference (ESSCIRC)*. [s.n.], 2021. p. 239–246. Disponível em: <doi:10.1109/ESSCIRC53450.2021.9567867>.
- RIAZ, A.; SOLOMON, A.; ERCAN, F.; MEDARD, M.; YAZICIGIL, R. T.; DUFFY, K. R. Noise recycling using GRAND for improving the decoding performance. In: *2023 15th International Conference on COMMunication Systems & NETWORKS (COMSNETS)*. [s.n.], 2023. p. 171–173. Disponível em: <doi:10.1109/COMSNETS56262.2023.10041405>.
- RICHARDSON, T.; SHOKROLLAHI, M.; URBANKE, R. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, v. 47, n. 2, p. 619–637, 2001. Disponível em: <doi:10.1109/18.910578>.
- RICHARDSON, T.; URBANKE, R. The renaissance of gallager’s low-density parity-check codes. *IEEE Communications Magazine*, v. 41, n. 8, p. 126–131, 2003. Disponível em: <doi:10.1109/MCOM.2003.1222728>.
- RIYADI, F. M.; ANWAR, K.; ADRIANSYAH, N. M. Study on channel coding for future railway mobile communication systems. In: *2022 IEEE Symposium on Future Telecommunication Technologies (SOFTT)*. [s.n.], 2022. p. 103–108. Disponível em: <doi:10.1109/SOFTT56880.2022.10009038>.
- SHANNON, C. E. A mathematical theory of communication. *The Bell system technical journal*, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948.
- SKLAR, B.; HARRIS, F. The ABCs of linear block codes. *IEEE Signal Processing Magazine*, v. 21, n. 4, p. 14–35, 2004. Disponível em: <doi:10.1109/MSP.2004.1311137>.
- TAHIR, B.; SCHWARZ, S.; RUPP, M. BER comparison between convolutional, turbo, LDPC, and polar codes. In: IEEE. *2017 24th international conference on telecommunications (ICT)*. 2017. p. 1–7. Disponível em: <doi:10.1109/ICT.2017.7998249>.
- TANNER, R. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, v. 27, n. 5, p. 533–547, 1981. Disponível em: <doi:10.1109/TIT.1981.1056404>.
- VERMA, A.; SHRESTHA, R. Low computational-complexity soms-algorithm and high-throughput decoder architecture for qc-ldpc codes. *IEEE Transactions on Vehicular Technology*, v. 72, n. 1, p. 66–80, 2023. Disponível em: <doi:10.1109/TVT.2022.3203802>.
- VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, v. 13, n. 2, p. 260–269, 1967. Disponível em: <doi:10.1109/TIT.1967.1054010>.
- WOZENCRAFT, J. M. Sequential decoding for reliable communication. Research Laboratory of Electronics, Massachusetts Institute of Technology, 1957.
- WU, H.; WANG, H. A high throughput implementation of QC-LDPC codes for 5G NR. *IEEE Access*, v. 7, p. 185373–185384, 2019. Disponível em: <doi:10.1109/ACCESS.2019.2960839>.
- YE, F. *Slepian-Wolf source coding using LDPC codes for free viewpoint television*. Tese (Doutorado) — IMT Atlantique, Brest, France, Dez 2019.

ZHANG, J.; FOSSORIER, M. A modified weighted bit-flipping decoding of low-density parity-check codes. *IEEE Communications Letters*, v. 8, n. 3, p. 165–167, 2004. Disponível em: <doi:10.1109/LCOMM.2004.825737>.

APÊNDICE A – Algoritmo LDPC

Algoritmo 1 Sistema de comunicação com codificação LDPC NR conforme 3GPP

Inputs: (\mathbf{m}, G, F, M)

\mathbf{m} : vetor binário pseudo-aleatório de tamanho L ;

G : matriz geradora (obtida através da matriz de paridade H , que é definida conforme o número da matriz base);

L : tamanho da mensagem;

F : número de bits de preenchimento;

M : número de símbolos da modulação (ex.: QPSK: $M=4$, 256QAM: $M=256$);

Output: $(\hat{\mathbf{m}})$

$\hat{\mathbf{m}}$: estimativa da mensagem \mathbf{m} .

» **CODIFICADOR DO CANAL** «

1: $\mathbf{m} \leftarrow [\mathbf{m} \ \mathbf{f}]$ ▷ Adicionar F bits de preenchimento à mensagem m .

2: $[\mathbf{c} \ \mathbf{f}] \leftarrow \mathbf{m} \cdot \mathbf{G}$ ▷ Codificar m através da matriz geradora G .

» **MODULADOR** «

▷ Segmentar a mensagem em sub mensagens de tamanho M , formando um vetor de tamanho $K = L/\log_2(M)$.

3: $\mathbf{c} \leftarrow [\mathbf{c}_0 \ \mathbf{c}_1 \ \dots \ \mathbf{c}_{K-1}]$, $\mathbf{c}_k \in \{0, 1\}^{\log_2(M)}$

▷ Modular a mensagem formando um símbolo (pertencente ao alfabeto \mathcal{D} definido pela modulação), a ser transmitido.

4: $\mathbf{s} = [s_0 \ s_1 \ \dots \ s_{K-1}]$, $s_k = f(\mathbf{c}_k)$

» **CANAL** «

5: $\mathbf{r}(t) = \mathbf{s}(t) + \mathbf{n}(t)$ ▷ Transmitir a mensagem por um canal AWGN adicionando ruído.

» **DEMODULADOR** «

6: **for** $k \leftarrow 0$ to $K - 1$ **do**

7: $\hat{c}_k = f(\mathbf{r}_k)$ ▷ Demodular o sinal estimando a *codeword* transmitida.

8: **end for**

9: $\hat{\mathbf{c}} = [\hat{c}_0 \ \hat{c}_1 \ \dots \ \hat{c}_{K-1}]$

» **DECODIFICADOR DO CANAL** «

10: $\hat{\mathbf{c}} \leftarrow [\hat{\mathbf{c}} \ \mathbf{f}]$ ▷ Adicionar F bits de preenchimento à mensagem demodulada.

11: **for** $k \leftarrow 0$ to $K - 1$ **do**

12: Decodificar a mensagem com ruído iterativamente conforme o algoritmo *belief propagation*.

13: **end for**

14: Retirar bits de preenchimento.

APÊNDICE B – Algoritmo Convolucional

Algoritmo 2 Sistema de comunicação com codificação Convolucional NR conforme 3GPP

Inputs: (\mathbf{m}, G, F, M)

\mathbf{m} : vetor binário pseudo-aleatório de tamanho L ;

L : tamanho da mensagem;

K : comprimento de restrição;

g : polinômio gerador;

M : número de símbolos da modulação (ex.: QPSK: $M=4$, 256QAM: $M=256$);

Output: ($\hat{\mathbf{m}}$)

$\hat{\mathbf{m}}$: estimativa da mensagem \mathbf{m} .

» **CODIFICADOR DO CANAL** «

for $n \leftarrow 0$ to $L - 1$ **do**

$$c_1[n] \leftarrow \sum_{j=0}^{K-1} [\mathbf{g}_1][j]m[n-j]$$

$$c_2[n] \leftarrow \sum_{j=0}^{K-1} [\mathbf{g}_2][j]m[n-j]$$

$$c_3[n] \leftarrow \sum_{j=0}^{K-1} [\mathbf{g}_3][j]m[n-j] \quad \triangleright \text{Codificar } m \text{ através dos polinômios geradores } \mathbf{g},$$

sendo $g_1 = [1011011]$, $g_2 = [1111001]$ e $g_3 = [1110101]$.

end for

» **MODULADOR** «

\triangleright Segmentar a mensagem em sub mensagens de tamanho M , formando um vetor de tamanho $L/\log_2(M)$.

$$\mathbf{c} \leftarrow [\mathbf{c}_0 \ \mathbf{c}_1 \ \dots \ \mathbf{c}_{L/\log_2(M)-1}], \quad \mathbf{c}_k \in \{0, 1\}^{\log_2(M)}$$

\triangleright Modular a mensagem formando um símbolo (pertencente ao alfabeto \mathcal{D} definido pela modulação), a ser transmitido.

$$\mathbf{s} = [s_0 \ s_1 \ \dots \ s_{L/\log_2(M)-1}], \quad s_k = f(\mathbf{c}_k)$$

» **CANAL** «

$\mathbf{r}(\mathbf{t}) = \mathbf{s}(\mathbf{t}) + \mathbf{n}(\mathbf{t})$ \triangleright Transmitir a mensagem por um canal AWGN adicionando ruído.

» **DEMODULADOR** «

for $k \leftarrow 0$ to $L/\log_2(M) - 1$ **do**

$$\hat{c}_k = f(\mathbf{r}_k) \quad \triangleright \text{Demodular o sinal estimando a } \textit{codeword} \text{ transmitida.}$$

end for

$$\hat{\mathbf{c}} = [\hat{c}_0 \ \hat{c}_1 \ \dots \ \hat{c}_{L/\log_2(M)-1}]$$

» **DECODIFICADOR DO CANAL** «

for $k \leftarrow 0$ to $L/\log_2(M) - 1$ **do**

Decodificar a mensagem com ruído através do algoritmo de Viterbi (Conforme a Equação 11).

end for

APÊNDICE C – Códigos desenvolvidos

Os códigos implementados para esse trabalho foram disponibilizados no GitHub a seguir: <https://github.com/cintiaporcher/gnuradio_MATLAB>.