



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DA UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA

Luíza Machado de Freitas

Estudo de códigos LDPC e polares

Florianópolis
2024

Luíza Machado de Freitas

Estudo de códigos LDPC e polares

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia Eletrônica do Centro Tecnológico da Universidade Federal de Santa Catarina da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia Eletrônica. Orientador: Prof. Bartolomeu Ferreira Uchôa-Filho, Dr.

Florianópolis

2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Freitas, Luíza Machado de
Estudo de códigos LDPC e polares / Luíza Machado de
Freitas ; orientador, Bartolomeu Ferreira Uchôa-Filho, 2024.
31 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Eletrônica, Florianópolis, 2024.

Inclui referências.

1. Engenharia Eletrônica. 2. Códigos LDPC. 3. Códigos Polares. 4. Concatenação de Códigos. 5. Correção de Erros. I. Uchôa-Filho, Bartolomeu Ferreira. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Eletrônica. III. Título.

Luíza Machado de Freitas

Estudo de códigos LDPC e polares

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Eletrônica” e aprovado em sua forma final pelo Curso de Graduação em Engenharia Eletrônica.

Florianópolis, 20 de Dezembro de 2024.

Profa. Dra. Daniela Ota Hisayasu Suzuki,
Dra.
Coordenadora do Curso

Banca Examinadora:

Prof. Bartolomeu Ferreira Uchôa-Filho, Dr.
Orientador

Profa. Fernanda Regina Smith Neves Corrêa,
Dra.
Avaliadora
Universidade Federal do Amapá

Prof. Richard Demo Souza, Dr.
Avaliador
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus queridos amigos e à minha amada família, cuja presença, apoio e amor foram fundamentais para a realização deste projeto

AGRADECIMENTOS

Gostaria de agradecer, em primeiro lugar, à minha família, pelo apoio incondicional e pelo amor que sempre me deram. A paciência, a compreensão e a confiança de todos foram fundamentais para minha jornada.

Gostaria de expressar minha gratidão ao meu orientador, Prof. Dr. Bartolomeu Ferreira Uchôa-Filho, pela paciência e dedicação ao longo de todo o processo de desenvolvimento deste trabalho. Seu conhecimento e experiência certamente enriqueceram minha formação acadêmica.

Aos meus amigos dentro do curso, agradeço pelo companheirismo ao longo desses seis anos. A parceria e apoio de todos durante as semanas de estudo e noites sem dormir foram essenciais para conseguir chegar até aqui.

Aos meus amigos fora do curso, muito obrigada pelo apoio, pelas palavras de incentivo e pela paciência durante os períodos de estudos intensos, sempre me ajudando a manter o equilíbrio (ou fazendo o possível).

Por fim, agradeço à instituição e aos professores. O aprendizado que adquiri aqui será sempre uma base para minha vida profissional.

Qualquer tecnologia suficientemente avançada é indistinguível da magia (Arthur C. Clarke, 1962)

RESUMO

A comunicação digital moderna depende de técnicas eficazes de codificação para corrigir erros causados por ruídos e interferências nos canais de transmissão. Dentre essas técnicas, os códigos polares e os códigos LDPC têm se destacado, especialmente com a evolução das redes 5G e o desenvolvimento das redes 6G, que exigem códigos de alta confiabilidade e eficiência. Os códigos LDPC, com matrizes de paridade esparsas, proporcionam uma decodificação eficiente e são amplamente utilizados em sistemas de comunicação modernos. Já os códigos polares, com base na polarização de canais, oferecem excelente desempenho, mas apresentam limitações em palavras-código curtas devido à distribuição intermediária de confiabilidades. Para superar essa limitação, a concatenação dos códigos polares com os códigos LDPC tem mostrado melhorar significativamente o desempenho, aproveitando as propriedades de polarização para otimizar a utilização de canais com confiabilidade média. Este trabalho explorou os códigos LDPC (incluindo a criação da matriz de paridade H) e polares de forma isolada, realizando simulações para avaliar seu desempenho e estudando a teoria da concatenação com o objetivo de analisar os ganhos de desempenho que essa abordagem pode oferecer.

Palavras-chave: Códigos LDPC, Códigos Polares, Concatenação de Códigos, Correção de Erros, Redes 6G, Desempenho de Decodificação.

ABSTRACT

Modern digital communication relies on effective coding techniques to correct errors caused by noise and interference in transmission channels. Among these techniques, polar codes and LDPC codes have stood out, especially with the evolution of 5G networks and the development of 6G networks, which demand highly reliable and efficient codes. LDPC codes, with sparse parity-check matrices, provide efficient decoding and are widely used in modern communication systems. On the other hand, polar codes, based on channel polarization, offer excellent performance but have limitations with short codewords due to the intermediate distribution of reliabilities. To overcome this limitation, the concatenation of polar codes with LDPC codes has shown to significantly improve performance, leveraging the polarization properties to optimize the use of channels with intermediate reliability. This work explored LDPC codes (including the creation of the parity-check matrix H) and polar codes in isolation, conducting simulations to evaluate their performance and studying the theory of concatenation with the goal of analyzing the performance gains this approach may offer.

Keywords: LDPC Codes, Polar Codes, Code Concatenation, Error Correction, 6G Networks, Decoding Performance.

LISTA DE FIGURAS

Figura 1 – Combinação de dois canais do tipo BEC	14
Figura 2 – Codificador, com 10 bits de mensagem e 6 congelados	20
Figura 3 – Árvore binária para decodificação com $N = 8$	21
Figura 4 – Ações realizadas nos nós internos durante a decodificação.	22
Figura 5 – Simulações BER para os códigos LDPC e Polar puros	26
Figura 6 – Polarização de um canal BEC com $\varepsilon = 0.5$	27

SUMÁRIO

1	INTRODUÇÃO	11
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	CÓDIGOS POLARES	13
2.1.1	Informações Preliminares	13
2.1.1.1	Produto de Kronecker	13
2.1.2	Polarização de Canal	13
2.1.3	Decodificação	14
2.2	CÓDIGOS LDPC	16
2.2.1	Informações Preliminares	16
2.2.2	Matriz de Paridade	17
2.2.3	Algoritmo PEG	18
2.2.4	Decodificação	19
3	SIMULAÇÕES	20
3.1	PROJETO DOS CÓDIGOS EM PYTHON	20
3.1.1	Polar	20
3.1.1.1	Codificação polar	20
3.1.1.2	Decodificação polar	20
3.1.2	LDPC	22
3.1.2.1	Construção de H	22
3.1.2.2	Parâmetros de simulação	24
3.2	RESULTADOS	25
3.3	CONCATENAÇÃO DE CÓDIGOS POLARES E LDPC (PRÓXIMOS PASSOS)	25
4	CONCLUSÃO E TRABALHOS FUTUROS	28
	REFERÊNCIAS	29

1 INTRODUÇÃO

A comunicação digital atual depende pesadamente dos códigos corretores de erros para garantir que a informação recebida corresponda à transmitida. Em cenários com ruídos e interferências, erros no canal não só são detectados, mas também podem ser corrigidos, quando se utilizam técnicas adequadas de codificação. A evolução das redes de comunicação, especialmente com o desenvolvimento da rede 5G e, posteriormente, 6G, exige melhorias constantes nos códigos de correção de erros, a fim de atender aos crescentes requisitos de confiabilidade e velocidade. Além disso, há uma demanda crescente por códigos eficientes com comprimento reduzido, devido aos casos de uso das redes de comunicação modernas.

Nos sistemas de comunicação modernos utilizam-se os códigos polares e os códigos Low-Density Parity-Check (LDPC). Os códigos LDPC, desenvolvidos por Robert Gallager na década de 1960 (GALLAGER, 1962), são códigos de correção de erros lineares que empregam matrizes de paridade esparsas, proporcionando uma decodificação com complexidade relativamente baixa. Por outro lado, os códigos polares, introduzidos por Arikan em 2009 (ARIKAN, 2009), exploram a polarização de canal para concentrar a probabilidade de erro nos bits menos confiáveis, aumentando assim a eficiência do código. Ambos os tipos foram adotados na tecnologia 5G (MIAO *et al.*, 2024).

Recentemente, houve um avanço significativo na pesquisa sobre codificação para redes 6G, visando enfrentar os desafios impostos por aplicações futuras de alta demanda, como realidade virtual e aumentada, Internet das Coisas massiva e comunicações confiáveis e de baixa latência. Os códigos polares estão sendo estudados como possibilidade para atingir os requisitos da 6G, mas sua baixa eficiência em palavras código curtas os coloca em desvantagem, principalmente sem concatenação com algum outro código (MIAO *et al.*, 2024). Essa limitação em desempenho dos códigos polares quando as palavras-código são curtas ocorre porque a polarização acaba deixando canais intermediários com confiabilidade média, ao invés de confiabilidades altas e baixas apenas. Para mitigar esse problema, foi inicialmente proposta a concatenação com códigos Reed-Solomon (BAKSHI; JAGGI; EFFROS, 2010), e posteriormente exploradas outras formas de concatenação, incluindo com códigos CRC (TAL; VARDY, 2015; ZHU; JIANG; ZHAO, 2022) e LDPC (ZHANG *et al.*, 2018; ABBAS *et al.*, 2017), todas mostrando melhorias no desempenho do código polar.

No estudo de (ABBAS *et al.*, 2017), foi analisada uma concatenação de código LDPC-polar, onde um código LDPC foi utilizado para proteger os bits de dados originalmente destinados aos canais com confiabilidade média. Similarmente, (ZHANG *et al.*, 2018) utilizou um código LDPC para proteger os bits intermediários, empregando um método diferente para decidir quais canais possuíam essa confiabilidade intermediária. Mais recentemente, (SHEN *et al.*, 2024) introduziu os códigos GLDPC-PC, que combinam

o código LDPC generalizado com conceitos dos códigos polares, trazendo maior taxa de correção de erros e mais rapidez na convergência da decodificação.

Além disso, o artigo (SMITH; PELAES; UCHÔA-FILHO, 2017) explora uma abordagem para melhorar o desempenho de sistemas QAM codificados com LDPC através de técnicas otimizadas de mapeamento de bits. Os autores introduzem uma estratégia de mapeamento de bits com o objetivo de aprimorar a eficiência na correção de erros, permitindo que bits transmitidos por sub-canais mais confiáveis apoiem aqueles transmitidos por sub-canais menos confiáveis. A técnica apresenta ganhos significativos de desempenho, sendo adequada para cenários com desequilíbrios de confiabilidade elevados entre sub-canais.

Como já foi mencionado, a codificação polar resulta em canais de bit com múltiplas confiabilidades devido à polarização de canal. Propõe-se nesse trabalho explorar maneiras de implementar ambos os códigos e também estudar a concatenação desses códigos (LDPC do artigo (SMITH; PELAES; UCHÔA-FILHO, 2017) e polares) para alcançar um bom desempenho na codificação com palavras-código de comprimento reduzido, aproveitando as propriedades de polarização de canal dos códigos polares para identificar canais intermediários, com uma confiabilidade de transmissão que não é muito alta, mas ainda assim não baixa o suficiente para serem inutilizados, e melhorar seu desempenho com o auxílio de um código LDPC especialmente projetado.

Este trabalho de conclusão de curso se dedica ao estudo, simulação e análise de códigos polares puros e códigos LDPC puros, bem como à proposta de uma forma de concatenar esses dois tipos de códigos. A pesquisa bibliográfica realizada abordou os aspectos fundamentais dos códigos polares e LDPC, suas aplicações e os métodos de decodificação. Adicionalmente, este trabalho incluiu um estudo sobre a criação da matriz de paridade para os códigos LDPC, sendo utilizada uma adaptação do algoritmo PEG (LCRYPTO, 2024), junto com algumas restrições iniciais. Foram realizadas simulações em Python para gerar essa matriz de paridade criada, bem como para testar os desempenhos dos códigos.

2 FUNDAMENTAÇÃO TEÓRICA

A codificação de canal é uma técnica que visa minimizar os efeitos de ruídos durante a transmissão de dados na comunicação digital. Muitas aplicações exigem uma alta confiabilidade, trazendo a necessidade do uso de códigos para transformar a mensagem original em um formato mais robusto. Em termos simples, a codificação de canal envolve a adição de redundância aos dados transmitidos. Isso é feito por meio de códigos de correção de erros, como os códigos polares e códigos LDPC estudados. Essa redundância permite que, mesmo com a presença de erros, a mensagem original possa ser recuperada ou corrigida, o que é essencial para garantir a qualidade da comunicação em sistemas digitais. Neste capítulo, serão apresentados os conceitos fundamentais dos códigos polares e LDPC. A primeira sessão trata dos códigos polares, explicando suas propriedades, a técnica de polarização de canal e os métodos de decodificação. Em seguida, abordamos os códigos LDPC, detalhando sua estrutura de matriz de paridade proposta e os principais métodos de decodificação.

2.1 CÓDIGOS POLARES

2.1.1 Informações Preliminares

2.1.1.1 Produto de Kronecker

O produto Kronecker é uma operação entre duas matrizes de tamanho arbitrário. O resultado é a matriz de bloco mostrada a seguir.

$$A_{m \times n} \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \quad (1)$$

Outra definição importante é a potência de Kronecker, ou $A^{\otimes n}$:

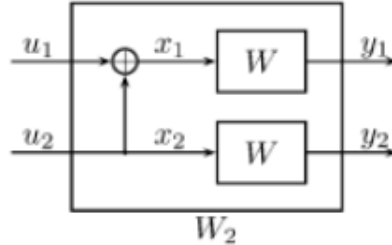
$$A^{\otimes n} = A \otimes A^{\otimes(n-1)} \quad (2)$$

para todo $n \geq 1$, em que $A^{\otimes 0}$ é a matriz identidade de dimensão 1x1. Esse conceito é utilizado recursivamente na criação da polarização de canal.

2.1.2 Polarização de Canal

Um “Binary Erasure Channel” (BEC) é um modelo de canal de comunicação em que ou o receptor recebe o bit transmitido corretamente ou, com uma certa probabilidade, recebe uma mensagem indicando que o bit foi apagado (“erased”). A operação fundamental da polarização consiste em combinar dois canais ($N=2$) com entrada binária (inicialmente

feito com um canal do tipo BEC) do modo mostrado na Figura 1. Ocorre que, com esta combinação, a taxa de erro do canal 1 aumenta, enquanto que a do canal 2 diminui (NIU *et al.*, 2014).



Fonte:ARIKAN, E., "Channel polarization: A method for constructing capacity-achieving codes,"

Figura 1 – Combinação de dois canais do tipo BEC

Repetindo-se recursivamente esta combinação para valores de N maiores (N canais), obtém-se uma distribuição, indo desde canais com valores de taxa de erro muito baixos (muito confiáveis) até canais pouco confiáveis. Esta combinação recursiva pode ser representada pela utilização sucessiva do produto Kronecker. Naturalmente, o número de canais obtidos desta forma será uma potência de 2 ($N = 2^n$).

Assim, dado um código de comprimento N e uma mensagem de comprimento $K < N$ bits, um bloco de código vai consistir dos K bits de informação, a serem transmitidos pelos canais mais confiáveis, e de mais $(N-K)$ bits “congelados”, a serem transmitidos pelos canais mais ruidosos. A sequência de confiabilidade (quais dentre os N canais do código são os mais confiáveis) é obtida através de cálculos de probabilidade. Existem várias sequências possíveis, mas no escopo desse trabalho foi utilizada a que foi definida como padrão da 5G ($N = 1024$).

Como exemplo, se a sequência de confiabilidade para $N = 8$ for dada por

$$Q = [1 \ 2 \ 3 \ 5 \ 4 \ 6 \ 7 \ 8]$$

uma mensagem m de tamanho $K = 4$ será transmitida pelos canais de maior confiabilidade, com bits congelados nos demais canais:

$$c = [0 \ 0 \ 0 \ m_1 \ 0 \ m_2 \ m_3 \ m_4]$$

2.1.3 Decodificação

A decodificação de códigos polares pode ser realizada por métodos distintos, sendo os dois principais o SC Decoding (Successive Cancellation Decoding) e o SC List Decoding (List Successive Cancellation Decoding). Ambos os métodos são baseados na técnica de successive cancellation, mas diferem na forma como processam as possíveis estimativas de cada bit transmitido.

O SC Decoding é o método clássico para a decodificação de códigos polares, no qual a decisão sobre o valor de cada bit transmitido é tomada de forma sequencial. Começa-se pelo bit mais significativo e, a cada passo, o algoritmo usa as informações dos bits anteriores para fazer uma previsão sobre o bit atual. A cada iteração, o algoritmo tenta cancelar as inconsistências, aproximando-se da solução correta.

No receptor, de posse da sequência recebida $\mathbf{y} = [y_0, y_1, \dots, y_{N/m-1}]$ e das decisões de bit de informação anteriores ao instante i , \hat{u}_0^{i-1} , o gerador de métrica de símbolo para bit obtém a razão de verossimilhança logarítmica (LLR) do bit de informação u_i (DONG-MIN SHIN; SEUNG-CHAN LIM; KYEONGCHEOL YANG, 2012)

$$L_N^{(i)}(\mathbf{y}, \hat{u}_0^{i-1}) = \log \frac{W_N^{(i)}(\mathbf{y}, \hat{u}_0^{i-1} | u_i = 0)}{W_N^{(i)}(\mathbf{y}, \hat{u}_0^{i-1} | u_i = 1)}, \quad (3)$$

em que $W_N^{(i)}(\mathbf{y}, u_0^{i-1} | u_i)$ é a densidade de probabilidades (PDF) que descreve o i -ésimo canal de bit virtual definido por Arikan (ARIKAN, 2009). O ponto de partida do algoritmo de decodificação SC é a obtenção das LLR's de canal dos bits codificados c_i , que no caso de transmissão binária (BPSK) é proporcional ao símbolo recebido y_i . Uma explicação do algoritmo de decodificação a partir desse ponto pode ser vista no capítulo de Simulações.

Embora eficiente em termos computacionais, o SC Decoding pode não ser o mais robusto, especialmente em canais com alto ruído, pois toma decisões com base em uma única sequência de cálculos, sem explorar outras possibilidades.

O SC List Decoding é uma versão aprimorada do SC Decoding, que, ao invés de fazer uma única decisão para cada bit transmitido, mantém uma lista de várias hipóteses para as sequências de bits possíveis a cada passo. Dessa forma, ao final do processo, o decodificador gera um conjunto de soluções possíveis e escolhe a mais provável. Essa escolha é feita normalmente com base em uma codificação CRC feita junto com a polar.

A Cyclic Redundancy Check (CRC) é uma técnica de verificação de erros que utiliza um polinômio gerador para adicionar uma sequência de bits (redundância) ao final de um bloco de dados. Através de uma divisão binária (modulo-2), a codificação CRC calcula um valor de verificação (checksum) que é anexado aos dados antes de sua transmissão ou armazenamento. Esse valor é utilizado para detectar erros nos dados recebidos ou lidos. A verificação é realizada ao aplicar novamente o polinômio gerador aos dados recebidos; se o resultado da divisão for zero, os dados estão corretos, caso contrário, um erro foi detectado.

Por exemplo, para dados 11011101 e um Polinômio Gerador $1011\ 1011_2$ (correspondente a $x^3 + x + 1$), o processo de codificação segue os seguintes passos:

- Adicionar Zeros: Como o polinômio gerador é de grau 3, 3 zeros são adicionados aos dados originais. Os dados 11011101 se tornam:

$$11011101 \rightarrow 11011101000$$

onde os 3 zeros são adicionados no final dos dados.

- Divisão Modulo-2: Realizando a divisão dos dados 11011101000 pelo polinômio gerador $1011\ 1011_2$ utilizando a operação XOR. O processo de divisão seria o seguinte:
- Divida os primeiros 4 bits de 11011101 por 1011. O resultado da operação XOR nos dá:

$$1101 \text{ XOR } 1011 = 0110$$

- Baixamos o próximo bit da sequência de dados, formando 11001100, e aplicamos a operação X

O valor final calculado após a divisão será o checksum, que será anexado aos dados originais antes de sua transmissão. A sequência de dados transmitida será:

$$\text{Dados Transmitidos} = 11011101 \text{ checksum}$$

onde o checksum é a sequência final obtida da operação de divisão.

Este processo garante a detecção de erros nos dados, pois qualquer erro durante a transmissão alterará o valor do checksum e será detectado ao realizar a verificação no receptor. Na recepção, o receptor aplica o mesmo polinômio gerador aos dados recebidos, incluindo o checksum. Se o resto da divisão for zero, os dados estão corretos; caso contrário, um erro de transmissão foi detectado.

Antes de a mensagem ser codificada com polarização, será adicionada a ela sua codificação CRC. Depois da transmissão da mensagem, será feita a decodificação como descrito acima. Sabendo que os últimos bits são os de CRC, é possível checar se a mensagem bate com seus bits de redundância, utilizando o polinômio gerador novamente. Assim, é possível obter uma melhor performance do que apenas com a decodificação simples.

Esse método oferece uma maior robustez em canais ruidosos, já que explora múltiplas possibilidades para os bits transmitidos, aumentando as chances de encontrar a sequência correta. Contudo, o SC List Decoding é mais complexo e exige maior poder computacional, pois requer a manutenção e análise de várias hipóteses simultaneamente.

2.2 CÓDIGOS LDPC

2.2.1 Informações Preliminares

Os códigos Low-Density Parity-Check (LDPC) são uma classe de códigos de correção de erros linearmente cíclicos, que possuem uma matriz de paridade esparsa, ou seja, uma matriz com um número relativamente pequeno de elementos diferentes de zero. Esses códigos têm sido amplamente estudados e aplicados devido ao seu desempenho excepcionalmente bom em termos de taxa de erro, especialmente em canais ruidosos. Os códigos

LDPC têm sido utilizados com sucesso em diversas tecnologias de comunicação, incluindo a norma Wi-Fi, DVB-S2, e, mais recentemente, na codificação de dados para as redes 5G.

A principal vantagem dos códigos LDPC é que sua estrutura esparsa permite a utilização de algoritmos de decodificação eficientes, como o algoritmo de decodificação de mensagem de passagem (BP) ou decodificação de soma e verificação (sum-product algorithm), que são capazes de aproximar o limite de Shannon para a capacidade de canal. Essa característica torna os códigos LDPC altamente eficientes em termos de capacidade de correção de erros, especialmente em longas distâncias de transmissão e em canais ruidosos.

2.2.2 Matriz de Paridade

A matriz de paridade é um elemento central na definição de um código LDPC. Esta matriz, normalmente denotada como H , é uma matriz esparsa e de grandes dimensões que define as equações de paridade que as palavras-código devem satisfazer. Cada linha da matriz corresponde a uma equação de paridade e cada coluna corresponde a um bit do código. O número de colunas da matriz H é igual ao número de bits na palavra-código, enquanto o número de linhas é igual ao número de equações de paridade. A matriz H deve ser construída de forma que o número de bits de paridade (linhas) seja suficientemente grande para garantir a correção de erros.

Uma característica importante dos códigos LDPC é que a matriz H é esparsa, ou seja, a maioria de seus elementos são zeros. Isso garante que a decodificação seja computacionalmente eficiente, uma vez que não é necessário manipular todos os elementos da matriz em cada passo do algoritmo de decodificação. A esparsidade da matriz H também significa que cada bit da palavra-código está envolvido em um número reduzido de equações de paridade, o que facilita a construção de algoritmos eficientes de decodificação.

A matriz de paridade H pode ser representada de várias formas, dependendo do tipo de código LDPC. Uma das formas mais comuns é a forma de blocos ou forma regular, em que o número de 1's em cada linha e cada coluna é constante. Em contrapartida, existem também códigos LDPC irregulares, onde o número de 1's por linha e por coluna pode variar. A escolha entre códigos LDPC regulares e irregulares depende dos requisitos específicos da aplicação, sendo que os códigos irregulares tendem a proporcionar um desempenho superior, mas com maior complexidade na construção da matriz H .

A matriz H de um código LDPC regular é estruturada de tal forma que se pode facilmente verificar se um conjunto de bits forma uma palavra-código válida. Para isso, basta multiplicar a palavra-código \mathbf{c} pela matriz H e verificar se o resultado é um vetor de zeros:

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

onde \mathbf{c} é a palavra-código e \mathbf{c}^T é a sua transposta. Caso a equação seja satisfeita,

a palavra-código é válida; caso contrário, erros foram detectados e a correção pode ser realizada durante o processo de decodificação.

2.2.3 Algoritmo PEG

Uma versão adaptada do algoritmo PEG (Progressive Edge-Growth) foi a técnica usada para gerar matrizes de paridade para códigos LDPC (Low-Density Parity-Check) de maneira eficiente. O algoritmo é útil na construção de códigos LDPC de grande dimensão, garantindo boas propriedades de desempenho, como a baixa densidade de vértices conectados em sua representação gráfica bipartida. Garante também que não haverá girth-4 no gráfico (algo que impossibilita a decodificação correta).

O objetivo principal do PEG é gerar uma matriz de paridade H , a partir de uma configuração inicial contendo um número fixo de linhas e colunas. Esta matriz deve ser esparsa e satisfazer a condição de paridade $Hx = 0$ para cada código de verificação.

1. Inicialização: A matriz de paridade H começa com uma configuração inicial onde cada vértice da parte de variáveis (colunas) tem conexões com uma linha da parte de verificação (linhas). Inicialmente, todas as linhas e colunas estão desconectadas.

2. Construção das Arestas: O algoritmo acrescenta progressivamente as arestas entre as variáveis e as verificações, uma por vez. Durante este processo, as conexões são feitas de forma a manter a matriz esparsa e garantir boas propriedades de codificação. O algoritmo escolhe as variáveis que estão mais "isoladas" ou "menos conectadas" à parte de verificação, evitando criar ciclos curtos.

3. Escolha da Melhor Aresta: A cada iteração, o algoritmo conecta um vértice de variável a um vértice de verificação, escolhendo a aresta com base em uma medida de grau. A conexão é feita entre o vértice de variável com o menor grau e o vértice de verificação com o menor grau.

4. Crescimento Progressivo: Este processo é repetido até que a matriz tenha o número desejado de conexões, garantindo que as arestas sejam adicionadas de maneira controlada para evitar ciclos pequenos e permitir uma boa correção de erros.

5. Propriedades Finais: O resultado é uma matriz H esparsa, onde as linhas representam as equações de paridade e as colunas representam as variáveis. A matriz gerada pelo PEG deve ter uma boa distância mínima, o que é importante para uma decodificação eficiente.

A matriz de paridade gerada pelo algoritmo PEG possui as seguintes características:

- Baixa densidade: A matriz é esparsa, com um número reduzido de conexões entre as variáveis e as verificações.
- Boa distribuição de grau: O algoritmo procura distribuir as conexões de forma balanceada entre as colunas e linhas, o que é desejável para uma boa performance na decodificação.

- Evita ciclos curtos: Embora o algoritmo não consiga eliminar completamente os ciclos curtos, ele minimiza sua ocorrência, o que melhora a eficiência da decodificação.

A explicação de como o algoritmo foi adaptado para o uso nesse caso pode ser vista na sessão de simulações.

2.2.4 Decodificação

A decodificação LDPC pode ser realizada de várias formas e uma das mais avançadas é o algoritmo Turbo-decoding Message-Passing (TDMP). Este algoritmo combina a decodificação de códigos LDPC com a ideia de códigos turbo, utilizando um processo iterativo de passagem de mensagens entre os nós do gráfico de Tanner para corrigir erros no código recebido.

O núcleo da decodificação é o algoritmo Turbo-decoding Message-Passing (TDMP). Este algoritmo é baseado na passagem iterativa de mensagens entre os checknodes e os nós de variável. A cada iteração, são realizadas as seguintes etapas:

1. Conversão Inicial: Os bits recebidos são convertidos de $0 \rightarrow +1$ e $1 \rightarrow -1$, para trabalhar com valores probabilísticos.
2. Passagem de Mensagens: Durante cada iteração, as mensagens L_{NM} (mensagens de nós de variável para checknodes) e L_{MN} (mensagens de checknodes para nós de variável) são atualizadas. As mensagens de cada checknode são calculadas considerando as mensagens anteriores dos outros checknodes e nós de variável.
3. Decisão Final (Quantização): Após as iterações, as mensagens finais são convertidas em valores binários. A regra de decisão é que os valores negativos se tornam 1 e os valores positivos se tornam 0.

3 SIMULAÇÕES

3.1 PROJETO DOS CÓDIGOS EM PYTHON

3.1.1 Polar

3.1.1.1 Codificação polar

A codificação polar utilizou uma árvore binária. Inicialmente, a mensagem a ser codificada é posicionada nas folhas, com a mensagem original nos canais mais confiáveis e os canais menos confiáveis preenchidos com zeros. A árvore representa o processo de polarização, onde os canais de comunicação são sucessivamente combinados em pares, depois em grupos de quatro, e assim por diante. Esse processo de polarização permite que a codificação aloque os bits de mensagem nos canais mais confiáveis, enquanto os bits nos canais menos confiáveis são fixados, geralmente com valor zero. À medida que o processo sobe na árvore, os canais vão se combinando e o vetor de mensagem é codificado, formando a sequência final que será transmitida.

A Figura 2 ilustra esse processo com dez bits de mensagem e seis bits de redundância, demonstrando a divisão recursiva dos canais e a alocação dos bits de forma eficiente. O algoritmo garante que, ao final do processo, a mensagem original seja transmitida de forma robusta, utilizando os canais mais confiáveis e mantendo a complexidade da codificação em $O(N \log N)$.

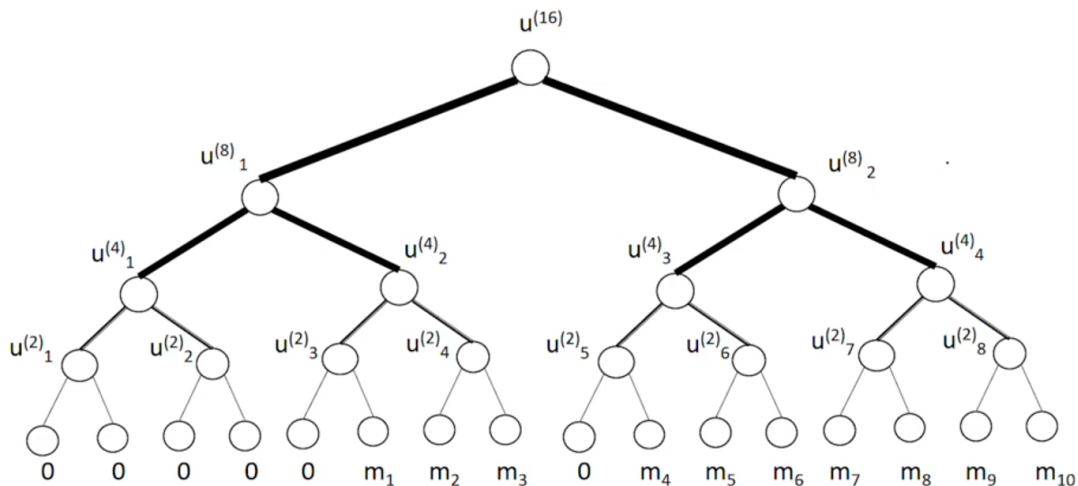


Figura 2 – Codificador, com 10 bits de mensagem e 6 congelados

Fonte: <https://www.youtube.com/playlist?list=PLyqSpQzTE6M81HJ26ZaNV0V3R0Brcv-Kc>

3.1.1.2 Decodificação polar

O algoritmo para decodificação por cancelamento sucessivo pode ser implementado usando uma árvore binária, com os nós percorridos por profundidade (depth first). Essa árvore é tal que os dois filhos de cada nó representam vetores com metade do tamanho

do vetor representado pelo nó de origem. A complexidade dessa decodificação é dada por $O(N \log N)$. A Figura 3 mostra como é montada a função de decodificação, para $N = 8$, ao receber um vetor R . Observa-se que R (na raiz da árvore) representa um vetor que tem tamanho 8 e depois é dividido em duas metades, que também são divididas em duas até chegar ao último nível, com oito folhas representando vetores de tamanho 1.

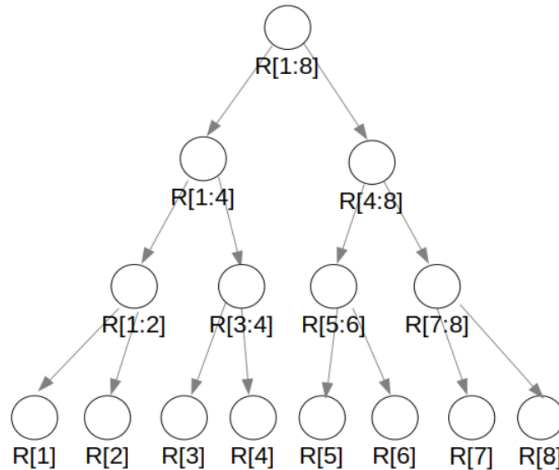


Figura 3 – Árvore binária para decodificação com $N = 8$.

O algoritmo de decodificação segue recursivamente pela árvore, com cada nó pedindo informações para os nós que estão imediatamente abaixo, e termina quando retorna à raiz. O ponto de partida é a raiz. As ações realizadas em cada ponto podem ser divididas em duas partes: ações realizadas nas folhas (nós da base da árvore) e ações realizadas no resto dos nós.

- **Folhas:** Quando na i -ésima folha, é feito o cálculo da razão de verossimilhança (LR). A partir da LR, é possível decodificar o bit \hat{u}_i :
 - Se a i -ésima posição não for de mensagem, então $\hat{u}_i = 0$.
 - Se for de mensagem e $LR < 1$, então $\hat{u}_i = 1$.
 - Para os demais casos, $\hat{u}_i = 0$.
- **Nós:** A Figura 4 mostra as ações para cada situação de chegada no nó vermelho. No final do processo, ter-se-á passado por cada nó três vezes:
 - Primeira vez: Ao chegar no nó, pede-se informação (qual o valor da mensagem) para sua metade esquerda.
 - Segunda vez: Obteve-se informação da metade esquerda. Agora pede-se a informação da metade direita.
 - Terceira vez: Sabe-se ambas as metades. Essas são então concatenadas e mandadas para o nó acima, que havia pedido a informação.

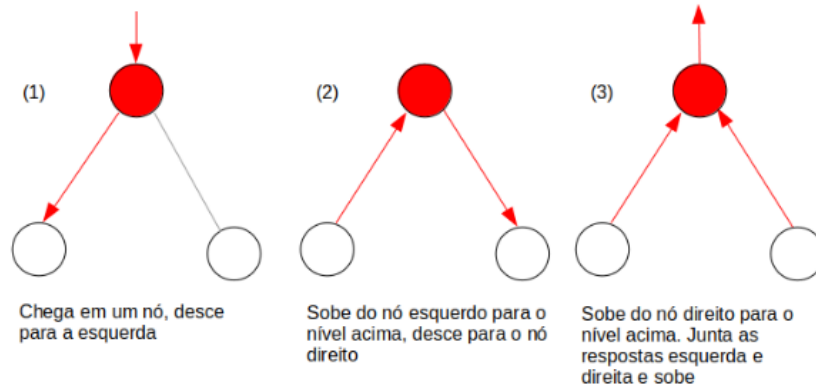


Figura 4 – Ações realizadas nos nós internos durante a decodificação.

3.1.2 LDPC

A simulação LDPC realizou uma transmissão de dados com modulação BPSK e ruído Gaussiano. O foco da simulação foi a construção de uma matriz de paridade H personalizada, que obedece a certas exigências estruturais, incluindo uma diagonal dupla no final e uma coluna de grau 3 antes da diagonal. Além disso, a construção da matriz H segue o algoritmo PEG (Progressive Edge Growth) (LCRYPTO, 2024), garantindo que a matriz resultante tenha as propriedades desejadas para a decodificação eficiente. A decodificação é realizada utilizando o algoritmo de Decodificação Turbo Message-Passing (TDMP), que é eficaz em ambientes ruidosos.

3.1.2.1 Construção de H

A construção da matriz H segue um procedimento baseado na especificação dada para o código LDPC. A seguir, um exemplo de construção da matriz de paridade H_{10} é apresentado. Inicialmente, algumas colunas da matriz H_{10} são definidas, resultando numa submatriz de paridade 8×8 , com um formato que visa a facilitar a codificação:

$$H_{10} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Esta matriz é o ponto de partida para a construção da matriz de paridade final, que será utilizada no processo de codificação. A partir de H_{10} acima, o algoritmo PEG foi utilizado para adicionar colunas à esquerda desta submatriz, em quantidade que depende do tamanho da palavra-código e da taxa do código. O algoritmo PEG é um método de

crescimento progressivo de arestas, no qual as colunas são adicionadas iterativamente, respeitando as propriedades de baixa densidade e a distribuição de grau das colunas.

Após a aplicação do algoritmo PEG neste exemplo, uma matriz de paridade binária 8×16 foi produzida, correspondendo a um código de taxa meio. No próximo passo, realizou-se uma troca de valores dos elementos da matriz binária conforme o tamanho de palavra-código desejado. Com a troca, uma matriz não binária é produzida cada elemento da qual representando uma submatriz $b \times b$, em que b é chamado de fator de expansão. Como exemplo, consideremos um fator de expansão $b = 64$, para obtermos uma matriz de paridade final binária 512×1024 . A troca de elementos neste caso é feita da seguinte forma:

- 1s da diagonal dupla são trocados por 0s
- 0s são trocados por -1s
- Demais 1s são trocados por qualquer valor inteiro entre 0 e 63

resultando na matriz não binária

$$H_{10nb} = \begin{bmatrix} 5 & -1 & -1 & -1 & 23 & -1 & 54 & -1 & 15 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 17 & 58 & -1 & -1 & -1 & -1 & 31 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\ 38 & -1 & -1 & 63 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 26 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\ -1 & 5 & -1 & -1 & 18 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\ 10 & -1 & -1 & -1 & -1 & 32 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & 62 & -1 & -1 & -1 & 33 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & -1 & 41 & -1 & -1 & -1 & 16 & 15 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 \end{bmatrix}$$

Com esses valores é possível expandir a matriz para o tamanho desejado (no caso da H10, espera-se palavras-código de tamanho 1024). Para isso considera-se

- Cada -1 representa uma matriz de 0s de tamanho 64×64
- Cada valor n entre 0 e 63 representa a matriz 64×64 produzida permutando-se ciclicamente as colunas de uma matriz identidade 64×64 em n deslocamentos.

Gerando a versão final de H , denotada como H_{final} , que obedece às exigências do código LDPC personalizado e tem dimensões 512×1024 .

Além da matriz H_{10} , uma matriz auxiliar H_{T2} também é utilizada para garantir a diversidade necessária no código. A matriz H_{T2} tem a seguinte forma:

As simulações realizadas foram baseadas na estrutura do codificador e decodificador do padrão WiMAX (MYERSW12, 2024), utilizando o código LDPC (Low-Density Parity-Check) adaptado para este sistema. O WiMAX (Worldwide Interoperability for Microwave Access) é um padrão de comunicação sem fio de alta capacidade que permite o acesso à internet e outros serviços de dados em áreas urbanas e rurais. Ele é projetado para oferecer conexões de banda larga de longa distância, podendo ser utilizado em sistemas fixos ou móveis, e é conhecido por sua capacidade de fornecer altas taxas de transmissão em canais ruidosos e com grande distorção.

No WiMAX, a camada física (PHY) é responsável por garantir a integridade dos dados transmitidos, utilizando codificação e modulação avançadas. O código LDPC é uma das principais técnicas utilizadas para correção de erros no WiMAX, sendo altamente eficaz em cenários ruidosos. As simulações aproveitaram a estrutura base de codificação e decodificação do WiMAX, utilizando uma matriz de paridade construída separadamente mas que segue o modelo do sistema, com a modulação BPSK e o algoritmo de decodificação Turbo Message-Passing (TDMP).

3.2 RESULTADOS

As simulações dos códigos polares foram feitas com 5000 blocos de mensagem, com $K = 512$ bits de mensagem e $N = 1024$ bits codificados. Já as dos códigos LDPC foram feitas para ambas a matriz H_{10} , com $K = 512/N = 1024$, como a matriz H_{T2} , com $K = 768/N = 1536$, com 300 blocos em cada simulação.

As taxas de erro BER para ambas as matrizes H construídas para o código LDPC, assim como para os códigos polares, podem ser vistas na Figura 5. Observa-se que de fato a Decodificação por Cancelamento Sucessivo em Lista tem um desempenho consideravelmente melhor do que a Decodificação por Cancelamento Sucessivo Simples. Além disso, para sinais com qualidade pior (menor razão de energia por bit sobre a densidade espectral de potência do ruído) os códigos LDPC foram mais eficientes na correção de erros, mas apresentaram desempenhos similares quando a E_b/N_0 chegou a $2.5dB$.

3.3 CONCATENAÇÃO DE CÓDIGOS POLARES E LDPC (PRÓXIMOS PASSOS)

A ideia central da codificação polar baseia-se na polarização dos canais de transmissão, onde, por meio de um processo iterativo de codificação, alguns canais se tornam altamente confiáveis, enquanto outros se tornam muito ruidosos. Assim, a codificação polar atribui os bits de informação aos canais mais confiáveis (próximos de 1), enquanto os bits congelados, que não carregam informação, são alocados nos canais com confiabilidade muito baixa (próximos de 0). Esse processo torna a comunicação eficiente, pois os bits importantes são transmitidos através dos canais mais confiáveis, enquanto os bits congelados, com pouca relevância para a transmissão, são alocados nos canais mais ruidosos.

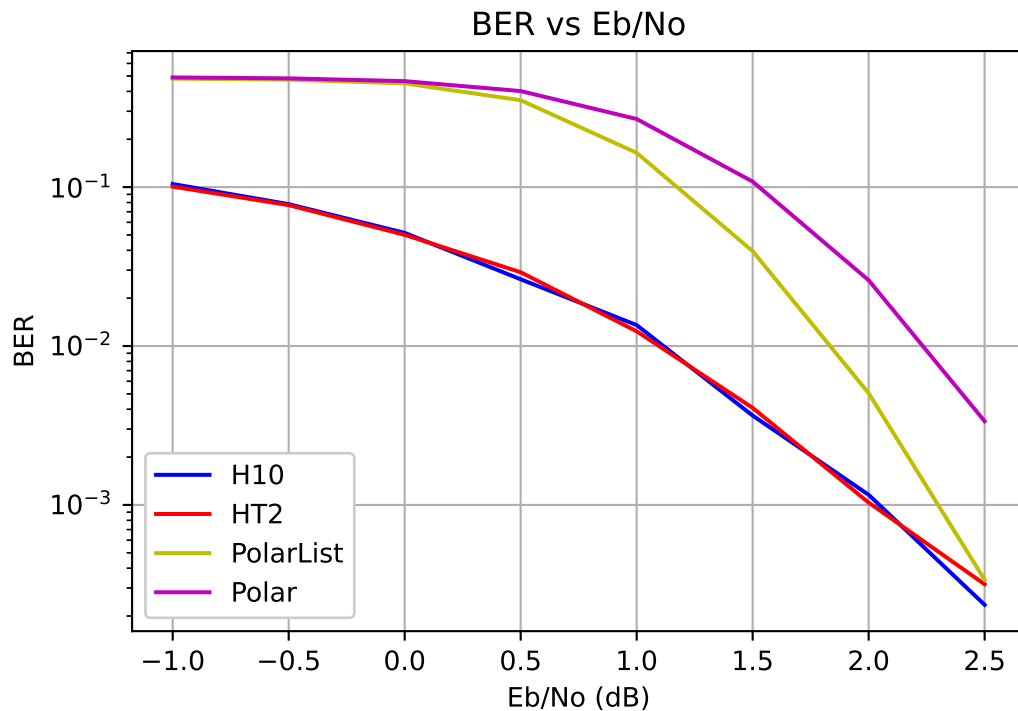


Figura 5 – Simulações BER para os códigos LDPC e Polar puros

No entanto, a eficácia da codificação polar diminui à medida que o tamanho da palavra-código diminui. A Figura 6 mostra as confiabilidades para $N = 1024$ em um canal BEC $\varepsilon = 0.5$. Nesse caso vemos uma grande quantidade de canais com alta e baixa confiabilidade (polarização feita de forma efetiva), mas em palavras-código mais curtas, a polarização dos canais não é tão acentuada, o que resulta em muitos canais com confiabilidade intermediária. Esse comportamento dificulta o uso eficiente dos canais para transmitir dados de forma confiável, pois muitos dos canais que deveriam ser de alta ou baixa confiabilidade apresentam um desempenho médio, o que os torna inadequados tanto para a transmissão de dados quanto para o congelamento dos bits.

Para contornar essa limitação, a solução proposta que está sendo estudada é a concatenação desses bits intermediários com códigos LDPC. O código LDPC é projetado para melhorar a confiabilidade desses canais de desempenho médio. Com isso, espera-se que a performance da codificação polar seja aumentada, permitindo que os canais de confiabilidade intermediária, que seriam descartados no esquema polar puro, sejam utilizados de forma mais eficiente.

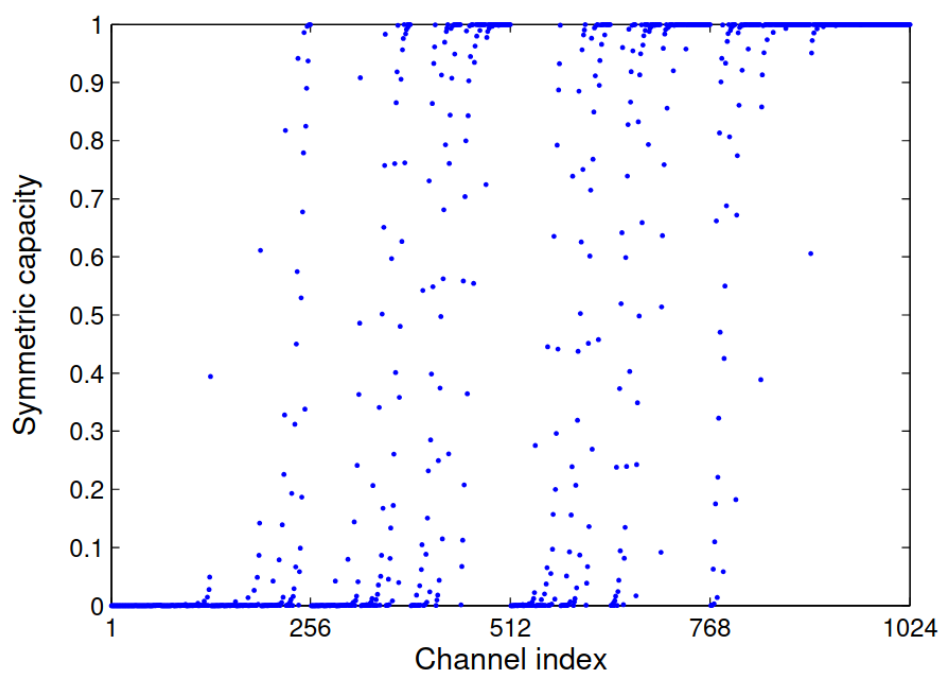


Figura 6 – Polarização de um canal BEC com $\varepsilon = 0.5$.

Fonte: (ARIKAN, 2009)

4 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como objetivo investigar o desempenho de sistemas de comunicação através de dois tipos de códigos de correção de erros: os códigos polares e os códigos LDPC. A concatenação desses dois códigos foi proposta como uma solução para aumentar a eficiência na transmissão de dados em canais ruidosos, aproveitando as vantagens específicas de cada código.

Foram feitas simulações e implementações com códigos LDPC e polares puros, além de um estudo teórico sobre a concatenação. A partir das simulações realizadas, foi possível observar que, individualmente, tanto os códigos polares quanto os códigos LDPC oferecem um bom desempenho em cenários com ruído AWGN, com o código LDPC demonstrando maior robustez em cenários com maior interferência.

Além disso, a implementação em Python dos algoritmos de codificação e decodificação foi bem-sucedida, proporcionando uma base sólida para a análise dos códigos em um ambiente controlado.

A construção personalizada das matrizes de paridade para o código LDPC, utilizando o algoritmo PEG, permitiu uma flexibilidade adicional na adaptação do código às necessidades específicas de desempenho e eficiência, contribuindo para a melhoria da correção de erros. Já os códigos polares, com sua estrutura de polarização de canais, provaram ser uma boa escolha para a codificação em sistemas de alta velocidade e baixa latência.

Os estudos efetuados neste trabalho sugerem que a combinação de códigos polares e LDPC oferece uma solução promissora para melhorar a eficiência das redes de comunicação em cenários com alta interferência e ruído. Uma pesquisa de mestrado será conduzida para aprofundar os estudos sobre a concatenação de códigos LDPC e polares, explorando de maneira mais detalhada os parâmetros envolvidos, como o projeto das matrizes de paridade para os códigos LDPC e a estrutura de polarização dos códigos polares. O objetivo será investigar as melhores formas de otimizar a concatenação e avaliar seu desempenho em uma variedade mais ampla de cenários de canais, buscando refinar as estratégias de codificação e decodificação.

REFERÊNCIAS

- ABBAS, Syed Mohsin *et al.* **Concatenated LDPC-Polar Codes Decoding Through Belief Propagation**. [*S.l.*]: arXiv, mar. 2017. arXiv:1703.05542 [cs, math]. Disponível em: <http://arxiv.org/abs/1703.05542>. Acesso em: 16 jul. 2024.
- ARIKAN, Erdal. 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, jul. 2009. ISSN 0018-9448, 1557-9654. DOI: 10.1109/TIT.2009.2021379. Disponível em: <https://ieeexplore.ieee.org/document/5075875/>. Acesso em: 15 jul. 2024.
- BAKSHI, Mayank; JAGGI, Sidharth; EFFROS, Michelle. **Concatenated Polar Codes**. [*S.l.*]: arXiv, jan. 2010. arXiv:1001.2545 [cs, math]. DOI: 10.48550/arXiv.1001.2545. Disponível em: <http://arxiv.org/abs/1001.2545>. Acesso em: 15 jul. 2024.
- DONG-MIN SHIN; SEUNG-CHAN LIM; KYEONGCHEOL YANG. Mapping Selection and Code Construction for 2^m -ary Polar-Coded Modulation. **IEEE Communications Letters**, v. 16, n. 6, p. 905–908, jun. 2012. ISSN 1089-7798. DOI: 10.1109/LCOMM.2012.040912.120070. Disponível em: <http://ieeexplore.ieee.org/document/6185517/>. Acesso em: 23 dez. 2024.
- GALLAGER, R. Low-density parity-check codes. en. **IEEE Transactions on Information Theory**, v. 8, n. 1, p. 21–28, jan. 1962. ISSN 0018-9448. DOI: 10.1109/TIT.1962.1057683. Disponível em: <http://ieeexplore.ieee.org/document/1057683/>. Acesso em: 15 jul. 2024.
- LCRYPTO. **Progressive Edge Growth (PEG) for LDPC and QC-LDPC codes construction using C++, Matlab and Python including ACE maximization and Avoiding Generating Small Cycles**. [*S.l.*: *s.n.*], 2024. <https://github.com/Lcrypto/classic-PEG-/tree/master>. Accessed: 2024-12-18.
- MIAO, Sisi *et al.* Trends in Channel Coding for 6G. **Proceedings of the IEEE**, p. 1–23, 2024. ISSN 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2024.3416050. Disponível em: <https://ieeexplore.ieee.org/document/10571997/>. Acesso em: 15 jul. 2024.
- MYERSW12. **LDPC (Low Density Parity Check) Library**. [*S.l.*: *s.n.*], 2024. https://github.com/myersw12/wimax_ldpc_lib. Accessed: 2024-12-18.
- NIU, Kai *et al.* Polar codes: Primary concepts and practical decoding algorithms. **IEEE Communications Magazine**, v. 52, n. 7, p. 192–203, jul. 2014. ISSN 0163-6804. DOI: 10.1109/MCOM.2014.6852102. Disponível em: <http://ieeexplore.ieee.org/document/6852102/>. Acesso em: 23 dez. 2024.

SHEN, Li *et al.* **GLDPC-PC Codes: Channel Coding Towards 6G Communications**. [S.l.]: arXiv, abr. 2024. arXiv:2404.14828 [cs, math]. Disponível em: <http://arxiv.org/abs/2404.14828>. Acesso em: 15 jul. 2024.

SMITH, Fernanda; PELAES, Evaldo; UCHÔA-FILHO, Bartolomeu F. EXIT charts analysis of a root-like bit mapping for LDPC-coded QAM systems. en. **Digital Signal Processing**, v. 70, p. 39–48, nov. 2017. ISSN 10512004. DOI: 10.1016/j.dsp.2017.07.019. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1051200417301690>. Acesso em: 15 jul. 2024.

TAL, Ido; VARDY, Alexander. List Decoding of Polar Codes. **IEEE Transactions on Information Theory**, v. 61, n. 5, p. 2213–2226, mai. 2015. ISSN 0018-9448, 1557-9654. DOI: 10.1109/TIT.2015.2410251. Disponível em: <http://ieeexplore.ieee.org/document/7055304/>. Acesso em: 15 jul. 2024.

ZHANG, Xueting *et al.* Interleaver Design for LDPC-Partial Polar Codes Based on EXIT Analysis. *In: 2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. [S.l.: s.n.], 2018. P. 1–5. DOI: 10.1109/ISWCS.2018.8491247.

ZHU, Mingyang; JIANG, Ming; ZHAO, Chunming. Adaptive Belief Propagation Decoding of CRC Concatenated NR LDPC and Polar Codes. **IEEE Transactions on Communications**, v. 70, n. 8, p. 4991–5003, ago. 2022. ISSN 0090-6778, 1558-0857. DOI: 10.1109/TCOMM.2022.3184359. Disponível em: <https://ieeexplore.ieee.org/document/9800962/>. Acesso em: 15 jul. 2024.