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

MATHEUS GABRIEL DIAS BARROS

Estudo, Implementação e Análise de Métodos de Polarização de Canal para Códigos Polares em Cenários de 5G NR

São João da Boa Vista
2024

Matheus Gabriel Dias Barros

Estudo, Implementação e Análise de Métodos de Polarização de Canal para Códigos Polares em Cenários de 5G NR:

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

Orientadora: Profª. Dra. Cintya Wink de Oliveira Benedito

São João da Boa Vista
2024

B277e

Barros, Matheus Gabriel Dias

Estudo, implementação e análise de métodos de polarização de canal para códigos polares em cenários de 5G NR / Matheus Gabriel Dias Barros. -- São João da Boa Vista, 2024

68 p. : tabs.

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

Orientadora: Cintya Wink de Oliveira Benedito

1. Telecomunicação. 2. Sistemas de telecomunicação. 3. Sistemas de comunicação sem fio. 4. Sistemas de transmissão de dados. 5. Codificação. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

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

TRABALHO DE CONCLUSÃO DE CURSO

ESTUDO, IMPLEMENTAÇÃO E ANÁLISE DE MÉTODOS DE POLARIZAÇÃO DE CANAL PARA CÓDIGOS POLARES EM CENÁRIOS DE 5G NR

Aluno: Matheus Gabriel Dias Barros
Orientador: Prof.ª Dr.ª Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientadora)
- Ivan Aritz Aldaya Garde (Examinador)
- Marlon Rodrigues Garcia (Examinador)

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

São João da Boa Vista, 05 de dezembro de 2024

DADOS CURRICULARES

MATHEUS GABRIEL DIAS BARROS

NASCIMENTO	26/10/1999 - São João da Boa Vista/ SP
FILIAÇÃO	Marcelo Cazumba de Barros Cristina de Fátima Dias
2018 / 2024	Engenharia Eletrônica e de Telecomunicações Faculdade de Engenharia de São João - FESJ

A Deus, a minha mãe, Cristina de Fátima Dias, a minha avó, Sebastiana Barbosa Dias, e a minha irmã, Millena Gabriela Dias Barros, dedico.

AGRADECIMENTOS

A Deus, primeiramente, por ter me guiado, cuidado e conduzido pelo caminho até o presente.

A minha mãe, Cristina de Fátima Dias, eu sou grato por todo o amor e dedicação empregados na minha criação

A minha avó, Sebastiana Barbosa Dias, eu agradeço por sempre estar ao meu lado como uma segunda mãe e por nunca ter deixado faltar amor e carinho.

A minha irmã, Millena Gabriela Dias Barros, eu sou grato por dividir essa vida desde o primeiro dia de vida.

Ao meu amigo da vida, Miguel Marcondes Martins Araujo, eu agradeço todos esses anos apoio incondicional.

A minha amiga Isabella Silva Teixeira, eu sou grato pela alegria que ela espalhou nos dias da graduação e nos momentos fora da universidade.

A minha orientadora Prof° Cintya Wink de Oliveira Benedito, eu agradeço por todo o apoio durante a Iniciação Cientifica e o Trabalho de Conclusão de Curso e por me encorajar a sempre continuar estudando.

Agradeço aos membros da banca de examinadores, pela disponibilidade e atenção dedicadas a este trabalho, além das sugestões de melhorias sobre o mesmo.

Agradeço à Faculdade de Engenharia de São João da Boa Vista, e a todos que participaram ativamente ou de espirito para que este trabalho fosse realizado.

“Lembre-se de olhar para as estrelas e não para baixo, para os seus pés. Tente achar sentido no que você vê e pergunte sobre o que faz o Universo existir. Seja curioso.“

Stephen Hawking

RESUMO

A crescente demanda por sistemas de comunicação mais eficientes e confiáveis, especialmente no contexto das redes móveis *Fifth Generation New Radio* (5G NR), impulsionou o desenvolvimento de técnicas avançadas de codificação de canal, como os códigos polares. Este estudo tem como objetivo analisar os métodos de polarização de canal Bhattacharyya e *Density Evolution with Gaussian Approximation* (DEGA), avaliando sua eficiência na construção de códigos polares para cenários de transmissão sujeitos ao canal *Additive White Gaussian Noise* (AWGN), como é o caso das tecnologias 5G NR. A pesquisa realiza uma comparação detalhada entre os métodos, considerando diferentes configurações de construção, como códigos sistemáticos e não sistemáticos, destacando os pontos fortes de cada abordagem. Ao final, são apresentados os impactos dessas técnicas no desempenho dos sistemas de comunicação, enfatizando sua aplicabilidade em contextos de alta demanda, como o 5G NR, e oferecendo subsídios para escolhas estratégicas em projetos futuros de comunicação sem fio.

PALAVRAS-CHAVE: códigos polares, codificação de canal, 5G NR , canal AWGN, redes móveis.

ABSTRACT

The growing demand for more efficient and reliable communication systems, especially in the context of Fifth Generation New Radio (5G NR) mobile networks, has driven the development of advanced channel coding techniques, such as polar codes. This work aims to analyze the Bhattacharyya and Density Evolution with Gaussian Approximation (DEGA) channel polarization methods, evaluating their efficiency in constructing polar codes for transmission scenarios under Additive White Gaussian Noise (AWGN) channels, as is the case with 5G NR technologies. The research provides a detailed comparison between the methods, considering different construction configurations, such as systematic and non-systematic codes, highlighting the strengths of each approach. Finally, the impacts of these techniques on communication system performance are presented, emphasizing their applicability in high-demand contexts, such as 5G, and offering insights for strategic choices in future wireless communication projects.

KEYWORDS: polar codes, channel coding, 5G NR, AWGN channel, mobile networks.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

5G	Quinta geração de redes móveis
AWGN	Additive White Gaussian Noise
NR	New Radio
LTE	Long Term Evolution
MIMO	Multiple Input Multiple Output
ETSI	European Telecommunications Standards Institute
ARIB	Association of Radio Industries and Businesses
TTC	Telecommunication Technology Committee
CCSA	China Communications Standards Association
ATIS	Alliance for Telecommunications Industry Solutions
TTA	Telecommunications Technology Association
3GPP	3rd Generation Partnership Project
eMBB	Enhanced Mobile Broadband
BER	Bit Error Ratio
DEGA	Density Evolution with Gaussian Approximation
SC	Successive Cancellation
SCL	Successive Cancellation List
DEGA	Density Evolution with Gaussian Approximation
PMF	Probability Mass Function
PDF	Probability Density Function
BEC	Binary Erasure Channel
B-DMC	Binary-Discrete Memoryless Channel.
BSC	Binary Symmetric Channel.
BPSK	Binary Phase Shift Keying
BER	Bit Error Ratio

LISTA DE SÍMBOLOS

X	Variável aleatória
χ	Alfabeto
p	Probabilidade
$\neg$	Negação
$H(X)$	Entropia
$E_p g(X)$	Esperança de uma variável aleatória
$H(X;Y)$	Entropia conjunta
$I(X;Y)$	Informação mútua
W	Canal discreto
C	Capacidade de informação
$C_{gaussiano}$	Capacidade do canal gaussiano
e	Bit apagamento
Y_i	Saída do canal de tempo discreto
X_i	Entrada do canal de tempo discreto
Z_i	Ruído do canal de tempo discreto
$\mathcal{N}$	Variância
P	Potência
B	Banda
n	Comprimento do código de entrada
N	Comprimento da palavra-código
m_k	Bits de mensagem
k	Comprimento do código de saída
b_k	Bits de paridade
c_n	Bits da palavra-código
p_{ij}	Coeficientes das equações de paridade

$\mathcal{C}$	Código polar
R	Taxa de código
w_h	Peso de Hamming
w_{min}	Peso mínimo de Hamming
G	Matriz geradora
I_k	Matriz identidade de ordem k
P	Matriz de ordem $(k \times (n-k))$
H	Matriz controle de paridade
$I(W)$	Capacidade simétrica
α	Constante
$Z(W)$	Parâmetro de Bhattacharyya
R_n	Bloco de permutação
F	Kernel de Arikan
$\oplus$	Xor
$\otimes$	Produto tensorial
$\mathbf{I}_{n_r}$	Matriz identidade de ordem n_r
$\det(\cdot)$	Determinante de uma matriz
$\log(\cdot)$	Logaritmo
$\|\| \cdot \|\|$	Norma de Frobenius
$(\cdot)^t$	Matriz transposta
$L_N^{(i)}$	Razão de verossimilhança
$\mathbf{V}$	Variância
μ	Valor médio
σ	Desvio padrão
ϕ	Função auxiliar
N_b	Número total de bits transmitidos
n_e	Número de bits que apresentaram erro durante a transmissão

SUMÁRIO

1 INTRODUÇÃO

A evolução contínua dos sistemas de comunicação tem impulsionado a demanda por técnicas cada vez mais eficientes na transmissão de dados, especialmente com a crescente necessidade por comunicação de alta velocidade e baixa latência (GLOBALDATA, 2022). No cenário dos sistemas de comunicação *Fifth Generation* (5G), a busca por soluções que melhorem a confiabilidade e a eficiência das transmissões de dados é uma prioridade (ROSLEE; TIANG; REHMAN, 2023). Entre as diversas abordagens emergentes, a utilização de códigos polares têm se destacado como uma alternativa promissora, proporcionando melhorias significativas na performance dos sistemas de comunicação através de técnicas de polarização de canal (ARIKAN, 2009; TAL; VARDY, 2011). Para representar situações práticas, o canal considerado é o *Additive White Gaussian Noise* (AWGN), que simula condições típicas de ruído em ambientes de comunicação sem fio (PROAKIS; SALEHI, 2008).

A quinta geração das redes de comunicações móveis introduz a tecnologia de *New Radio* (NR, Novo Rádio), desenvolvida para superar os limites da geração anterior, a *Long Term Evolution* (LTE) (PARKVALL et al., 2017). O 5G NR foi desenvolvido para atingir altas taxas de transmissão de dados com baixa latência, suportando aplicações avançadas como veículos autônomos, monitoramento de gado, plantações, realização de cirurgias à distância, transmissão de vídeos em ultra definição e realidade virtual (OSSEIRAN; MONSERRAT; MARSCH, 2016). Esta tecnologia opera em diversas faixas do espectro, permitindo o uso de células de diferentes tamanhos e taxas, inclusive em ondas milimétricas (ANDREWS et al., 2014). A técnica de múltiplas entradas e múltiplas saídas (MIMO, *Multiple Input Multiple Output*) é empregada para aumentar a capacidade de transmissão, otimizando a taxa de dados (BEKO et al., 2021). No entanto, o uso de frequências elevadas limita a distância de cobertura, exigindo a adoção do *beamforming* para direcionar a transmissão ou recepção de dados (SHAFI et al., 2017).

A harmonização global das tecnologias de comunicação móvel requer a ação coordenada de organizações capazes de definir requisitos, padrões e regulamentações. Em 1998, cinco entidades — o *European Telecommunications Standards Institute* (ETSI, Instituto Europeu de Padrões de Telecomunicações), a *Association of Radio Industries and Businesses* (ARIB, Associação das Indústrias e Empresas de Rádio), junto com o *Telecommunication Technology Committee* (TTC, Comitê de Tecnologia de Telecomunicações) do Japão, a *China Communications Standards Association* (CCSA, Associação de Padrões de Comunicações da China), a *Alliance for Telecommunications Industry Solutions* (ATIS, Aliança para Soluções da Indústria de Telecomunicações) da América do Norte, e a *Telecommunications Technology Association* (TTA, Associação de Tecnologia de Telecomunicações) da Coreia do Sul — se uniram para regulamentar o desenvolvimento das gerações de comunicações móveis.

Esta congregação, conhecida como *3rd Generation Partnership Project* (3GPP), ficou encarregada de estabelecer as regras e definições para as gerações de comunicações móveis (3GPP, 2023). Este trabalho foi desenvolvido com foco nas exigências de *Enhanced Mobile Broadband* (eMBB, Transmissões de Alta Velocidade) estabelecidas pelo 3GPP, que especificam uma razão de erro de bit (*Bit Error*

Ratio, BER) entre 10^{-6} e 10^{-8}, visando garantir a qualidade e a confiabilidade das comunicações em cenários de alta demanda de dados (3GPP, 2018).

Para atender às rigorosas exigências estabelecidas pelo 3GPP, foi necessário o desenvolvimento de novas técnicas e métodos avançados de codificação e processamento de sinais. Entre essas inovações, destacam-se os códigos polares, uma abordagem revolucionária que aproveita propriedades matemáticas para maximizar a eficiência da comunicação. Essas técnicas permitem não apenas atender aos requisitos de alta confiabilidade e baixa taxa de erros, mas também explorar de maneira eficiente a capacidade de canais de comunicação complexos e variados, como os presentes nos sistemas modernos de telecomunicações.

A polarização de canal, proposta por Arikan (ARIKAN, 2009), colaborou com o avanço da teoria de codificação ao introduzir uma técnica que transforma canais de comunicação não confiáveis em canais congelados, ou seja, não carregam informação, o que proporcionou o aumento da qualidade do sistema. Este método é fundamental para a implementação dos códigos polares, que foram os primeiros a se aproximar da capacidade de máxima de canais de comunicação binários e simétricos. Conforme destacado por Tal e Vardy (TAL; VARDY, 2011), a aplicação de métodos de polarização em sistemas de comunicação modernos, como o 5G, demonstrou um impacto significativo na redução da razão de erro de bit e no aumento da capacidade de canal. No entanto, a aplicação prática desses métodos enfrenta desafios consideráveis relacionados à complexidade computacional e à flexibilidade em diversos cenários de comunicação.

Este trabalho tem como objetivo principal investigar e analisar os métodos de polarização de canal, com foco específico em duas abordagens: o parâmetro de Bhattacharyya, proposto por Arikan a partir dos estudos realizados por A. Bhattacharyya (BHATTACHARYYA, 1943), e a DEGA (*Density Evolution with Gaussian Approximation*), proposta por Trifonov (TRIFONOV, 2012). Para atingir esse objetivo, são definidos vários objetivos específicos, começando pela compreensão dos fundamentos dos métodos de polarização de canal e a exploração dos princípios teóricos subjacentes. Isso inclui uma análise detalhada das técnicas de codificação e decodificação para códigos polares, que são cruciais para a implementação eficaz desses métodos.

Este trabalho esta estruturado da seguinte forma. No Capítulo 2 temos a revisão dos conceitos de teoria da informação e os tipos de canais relevantes para o estudo; o Capítulo 3 aborda a definição de códigos polares e como realizar a polarização de canal, bem como as estrategias de codificação e decodificação para transmissão desses códigos; o Capítulo 4 é o ponto principal do estudo, pois nele é apresentado de maneira minuciosa os métodos de polarização de canal para códigos polares; o Capítulo 5 apresenta os resultados e discussões das simulações de transmissão de códigos polares utilizando os métodos de polarização para a construção de canais confiáveis, feita utilizando a linguagem *Python*; por fim, o Capítulo 6 contém as conclusões finais do trabalho.

2 REVISÃO CONCEITUAL

Este capítulo tem o objetivo de elaborar uma revisão conceitual dos conceitos de Teoria da Informação e Codificação, que formam a base para a compreensão e desenvolvimento deste trabalho. Será apresentado inicialmente as definições e exemplos dos conceitos de informação mútua, entropia, capacidade de canal e alguns tipos de canais. Posteriormente, serão introduzidas as definições e os exemplos de códigos de blocos lineares, que nos ajudam a compreender a codificação de canal. Podemos encontrar as definições mais relevantes do presente capítulo nas referências de Cover e Thomas (2006) (COVER; THOMAS, 2006), Haykin (2001) (HAYKIN, 2001), Roth (2006) (ROTH, 2006), Ryan e Lin (2009) (RYAN; LIN, 2009) e Lin e Costello (2004) (LIN; COSTELLO, 2004).

2.1 TEORIA DA INFORMAÇÃO

Em 1948, Claude E. Shannon introduziu os conceitos que formam a base da Teoria da Informação (SHANNON, 1948). Sua teoria combinava elementos de matemática, engenharia e ciências, áreas que raramente eram integradas em uma única pesquisa. Nesse trabalho, Shannon explorou aspectos relacionados ao armazenamento, quantificação e transmissão de informação, com o objetivo de determinar os limites das operações de comunicação e do processamento digital de sinais. Considere X uma variável aleatória discreta com um conjunto de possiveis valores definidos por χ e função massa de probabilidade (PMF, *Probability Mass Function*).

$$p(x) = Pr\{X = x\}, x \in \chi. \tag{1}$$

Definição 2.1.1 *A entropia é a medida de incerteza de uma variável aleatória, logo a entropia H(X) de uma variável aleatória discreta X é definida por:*

$$H(X) = -\sum_{x \in \chi} p(x) log_2\, p(x)\ bits. \tag{2}$$

Por convenção, podemos definir $0log0 = 0$ pela continuidade, desde que $xlog(x) \to 0$, à medida que $x \to 0$. Podemos observar que a adição de termos de probabilidade nula não irá afetar o valor da entropia. Com isso, notamos também que a entropia é uma função da distribuição de X, não dependendo dos valores que a variável aleatória X pode tomar, mas sim de suas probabilidades.

Exemplo 2.1.1 *Considere uma PMF $p(x)$ com os valores definidos como*

$$X = \begin{cases} \frac{6}{10}, se\ x = a \\ \frac{4}{10}, se\ x = b \end{cases}. \tag{3}$$

Através da Expressão 2 podemos calcular a entropia

$$H(X) = -\left[\frac{6}{10} log_2 \left(\frac{6}{10}\right) + \frac{4}{10} log_2 \left(\frac{4}{10}\right)\right] \approx 0,83\ bits.$$

Definição 2.1.2 *Se (X, Y) são um par de variáveis aleatórias discretas com seus conjuntos de valores possíveis definidos como χ e γ respectivamente, tem-se a distribuição de probabilidade conjunta $p(x, y)$, onde $x \in \chi$ e $y \in \gamma$, então a entropia conjunta $H(X, Y)$ será*

$$H(X,Y) = -\sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)log(p(x,y), \tag{4}$$

podendo ser expressa em termos da esperança conjunta $E(p(x, y))$

$$H(X,Y) = -E[log(p(x,y))]. \tag{5}$$

Definição 2.1.3 *Se (X, Y) são um par de variáveis aleatórias discretas com distribuição conjunta $p(x, y)$, a entropia condicional é definida por:*

$$H(X|Y) = E[-logp(x|y)] = -\sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)log(p|y). \tag{6}$$

A quantidade de informação compartilhada entre duas variáveis aleatórias é conhecida como informação mútua. Essencialmente, ela representa a diminuição da incerteza sobre uma variável aleatória ao se conhecer a outra. Assim, podemos introduzir o conceito de informação mútua que pode ser empregada para medir a eficiência de uma transmissão por meio de um canal, pois quanto maior a dependência entre as duas variáveis, maior será a informação mútua e, consequentemente, a confiabilidade da transmissão

Definição 2.1.4 *Considere X e Ycomo duas variáveis aleatórias com distribuição de probabilidade conjunta $p(x, y)$ e PMF marginais $p(x)$ e $p(y)$. A informação mútua é definida como a entropia relativa entre a distribuição conjunta e o produto $p(x)p(y)$,*

$$I(X;Y) = \sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)log_2\left[\frac{p(x,y)}{p(x)p(y)}\right]. \tag{7}$$

Sabe-se que a probabilidade condicional entre x e y é

$$p(x|y) = \frac{p(x,y)}{p(y)}, \tag{8}$$

ao substituir a equação 8 na 7, tem-se:

$$I(X;Y) = \sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)log\left(\frac{p(x|y)}{p(x)}\right), \tag{9}$$

$$I(X;Y) = \sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)\frac{1}{log\ p(x)} - \sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)\frac{1}{p(x|y)}. \tag{10}$$

Para expressar a informação mútua em termos da entropia, podemos substituir as Equações 2 e 6 em (10), obtemos:

$$I(X;Y) = H(X) - H(X|Y). \tag{11}$$

A equação acima, pode ser escrita em função da entropia por qualquer uma das duas variáveis, isso implica que

$$H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{12}$$

Então, a informação mútua é descrita apenas pela soma das entropias individuais das duas variáveis com a subtração de sua entropia conjunta

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{13}$$

As definições de entropias e informação mútua para variáveis aleatórias discretas são estendidas para variáveis aleatórias contínuas. Nesse contexto, a entropia é conhecida como entropia diferencial, compartilhando muitas semelhanças com a versão para variáveis discretas.

Exemplo 2.1.2 EXEMPLO DE CÁLCULO DE INFORMAÇÃO MÚTUA

Considere duas variáveis aleatórias X e Y com os seguintes valores possíveis e a distribuição conjunta de probabilidades dada por $p(x, y)$:

$$p(x,y) = \begin{bmatrix} 0.1 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.2 \end{bmatrix},$$

onde as linhas correspondem aos valores de $X \in \{x_1, x_2\}$ e as colunas correspondem aos valores de $Y \in \{y_1, y_2, y_3\}$.

A distribuição marginal de X é obtida somando as probabilidades da distribuição conjunta ao longo das colunas:

$$p(x) = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}.$$

De forma semelhante, a distribuição marginal de Y é obtida somando as probabilidades ao longo das linhas:

$$p(y) = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix}.$$

Usando a definição de informação mútua:

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

podemos calcular cada termo individualmente.

Por exemplo, para $x = x_1$ e $y = y_1$:

$$\frac{p(x_1,y_1)}{p(x_1)p(y_1)} = \frac{0.1}{(0.4)(0.2)} = 1.25, \quad \log_2(1.25) \approx 0.32193.$$

A contribuição para $I(X;Y)$ neste caso é:

$$p(x_1, y_1) \cdot \log_2 \left(\frac{p(x_1, y_1)}{p(x_1)p(y_1)} \right) = 0.1 \cdot 0.32193 = 0.032193.$$

Repetindo este cálculo para todos os pares (x, y) e somando, obtemos o valor total de $I(X;Y)$. Para este exemplo, o resultado final é aproximadamente:

$$I(X;Y) \approx 0.214 \, \textit{bits}.$$

Esse resultado mostra a quantidade de informação compartilhada entre as variáveis X e Y, indicando o grau de dependência entre elas.

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

$$h(X) = -\int_X f(x) log\ f(x) dx. \tag{14}$$

Definição 2.1.6 *Sejam X e Y duas variáveis aleatórias contínuas, com distribuição de densidade de probabilidade $f(x)$ e $f(y)$, respectivamente, e uma distribuição conjunta $f(x, y)$, sua entropia conjunta $h(X, Y)$ é definida por*

$$h(X, Y) = -\int_X \int_Y f(x, y) log\ f(x, y) dx dy. \tag{15}$$

Definição 2.1.7 *Sejam duas variáveis aleatórias contínuas X e Y, com distribuição de densidade de probabilidade $f(x)$ e $f(y)$, com distribuição condicional $f(y|x)$, tem sua entropia condicional $h(Y|X)$ definida como*

$$h(Y|X) = -\int_X \int_Y f(x, y) log\ f(x|y) dx dy. \tag{16}$$

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

$$I(X;Y) = -\int_X \int_Y f(x, y) log \left[\frac{f(x, y)}{f(x)f(y)} \right] dx dy. \tag{17}$$

Como a definição de informação mútua de uma variável aleatória discreta e contínua são praticamente a mesma, podemos usar a relação usada na Equação 13, ou seja,

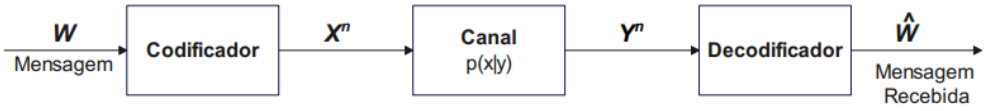
$$I(X;Y) = h(X) + h(Y) - h(X, Y). \tag{18}$$

2.2 CANAL DE COMUNIÇÃO

Um sistema de comunicações possuí diversas características que promovem a troca de informação, uma delas é o canal de comunicação. O canal é definido como o meio físico de transporte da informação de um ponto á outro. Para cada informação que deseja-se enviar, existe uma probabilidade de após

passar pelo canal, chegar de maneira correta ou incorreta. Isso acontece, pois o canal é um meio físico e está sujeito a degradação de sinal devido à fatores como ruídos e imperfeições em sua estrutura. No entanto, algumas estratégias são utilizadas para otimizar a probabilidade da informação correta chegar ao destino.

Figura 1 – Sistema de comunicação discreto



Fonte: elaborado pelo autor.

Considere que um sistema de comunicação discreto em amplitude e no tempo, seja descrito pelas partes vistas na Figura 1. Ao enviarmos uma mensagem ω pelo sistema, ela deve passar por um codificador que irá transforma-la em um conjunto de símbolos x^n, que por sua vez será transmitido pelo canal. O conjunto transmitido pelo canal sofre interferências, sendo elas construtivas ou não, mas o fato é que a mensagem não é mais a mesma, portanto, chamamos a palavra recebida no decodificador de y^n. Ao decodificar a palavra, tem-se uma mensagem $\hat{\omega}$, que não necessariamente é igual à mensagem enviada. Então, o canal discreto é definido como o conjunto de símbolos de entrada representados por um alfabeto finito $\chi = (x_1, x_2, ..., x_k)$, o conjunto de símbolos da saída será representado por um outro alfabeto finito $\gamma = (y_1, y_2, ..., y_j)$, transmitidos de um transmissor para um receptor. Podemos descrever essa transmissão de informação em termos probabilísticos, usando uma matriz de probabilidades de transição $p(y|x)$, onde o x representa o símbolo enviado e y o recebido. Em outras palavras, a matriz representa a probabilidade de receber um símbolo y dado que um símbolo x foi enviado. Essa dependência caracteriza um canal sem memória, ou seja, a transmissão atual de um símbolo não depende das transmissões anteriores ou futuras.

Um canal discreto sem memória é aquele onde a saída depende exclusivamente da entrada atual (GALLAGER, 1968). Não é condicional aos símbolos transmitidos anteriormente ou os que irão ser enviados.

Dados os alfabetos de entrada χ e saída γ, podemos definir as probabilidade de transições a partir das probabilidades condicionais. Ou seja, a probabilidade de obter o símbolo de saída y_j para um símbolo x_k enviado.

$$p(y_j|x_k) = P(\gamma = y_j|\chi = x_k) \ \forall \ 0 \leq p(y_j|x_k) \leq 1. \tag{19}$$

Com essa equação definida, é possível montar uma matriz que apresenta todas as probabilidades condicionais, em outras palavras, teremos as probabilidades de cada um dos símbolos da saída para

cada entrada isoladamente enviada.

$$M_p = \begin{bmatrix} p(y_1|x_1) & p(y_2|x_1) & ... & p(y_j|x_1) \\ p(y_1|x_2) & p(y_2|x_2) & ... & p(y_j|x_2) \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ p(y_1|x_k) & p(y_2|x_k) & ... & p(y_j|x_k) \end{bmatrix}. \tag{20}$$

Esta matriz têm dimensões $k \times j$, onde as linhas representam as entradas independentes do canal e as colounas as saídas independentes. Com essa matriz, pode-se definir as probabilidades a priori $p(x_k)$ e posteriori $p(y_j)$.

Definição 2.2.1 *Considere um canal discreto com alfabeto de entrada χ e alfabeto de saída γ, a capacidade é dada por*

$$C = max_{p(x)} I(X;Y), \tag{21}$$

onde o máximo é obtido de todas as possíveis distribuições de probabilidade $p(x)$ da entrada.

No caso binário, essa capacidade representa o limite máximo de bits que podem ser transmitidos sem que ocorra erro por uso do canal. A seguir iremos apresentar alguns canais que são de interesse neste trabalho.

2.2.1 Canal BSC

Binary Symmetric Channel (BSC) é um modelo simplificado de canal de comunicação amplamente utilizado em teoria da informação. Neste modelo, cada bit transmitido possui uma probabilidade p de ser corrompido durante a transmissão, ou seja, a probabilidade de um bit ser invertido (erro) é p, e a probabilidade do bit ser transmitido corretamente é $1 - p$ (MACKAY, 2003). O canal é denominado simétrico porque a probabilidade de erro é a mesma, independentemente do valor do bit original, seja 0 ou 1, como observado na Figura 2. Assim, o comportamento do BSC é totalmente determinado pela taxa de erro p.

Matematicamente, a probabilidade de erro de um bit transmitido x em um canal BSC é dada por:

$$P(Y = \neg X) = p, \tag{22}$$

onde X é o bit transmitido, Y é o bit recebido e $\neg X$ representa o bit invertido de X.

O canal BSC é o canal mais utilizado do caso discreto e é fundamental para o estudo da capacidade de outros canais. Ele também é utilizado para a análise de desempenho de códigos corretores de erros clássicos, como os códigos de Hamming e os códigos de Reed-Solomon.

A capacidade C de um canal simétrico binário é dada por:

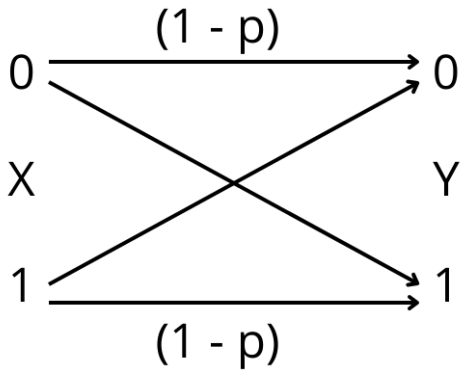
$$C = 1 - H(p)$$

onde $H(p)$ é a entropia de Shannon da distribuição de probabilidade do canal, dada por:

$$H(p) = -p \log_2 p - (1-p) \log_2 (1-p)$$

Neste contexto, p representa a probabilidade de erro do canal, ou seja, a probabilidade de que um símbolo transmitido seja recebido de forma incorreta. A capacidade do canal expressa a quantidade máxima de informação (em bits por símbolo) que pode ser transmitida com a mínima taxa de erro possível.

Figura 2 – Canal binário simétrico.

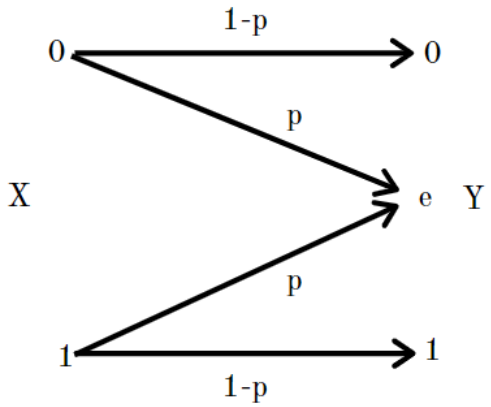


Fonte: elaborado pelo autor.

2.2.2 Canal BEC

O *Binary Erasure Channel* (BEC) é um canal binário caracterizado por apagamentos, onde alguns bits transmitidos são apagados a uma taxa arbitrária. Como ilustrado na Figura 3, este canal possui duas entradas e três saídas, sendo que uma fração p dos bits é apagada durante a transmissão. Nesse contexto, o receptor possui a capacidade de identificar exatamente quais bits foram apagados, permitindo um processo mais eficiente de recuperação da informação.

Figura 3 – Canal binário com apagamento.



Fonte: elaborado pelo autor.

Podemos calcular a capacidade desse canal como

$$C = \max_{p(x)} I(X;Y) = \max_{p(x)}((H(Y) - H(Y|X)) = \max_{p(x)} H(Y) - H(p), \tag{23}$$

considerando $Pr(X = 1) = \pi$, temos que

$$C = \max_{\pi} H(Y) - H(p) = \max_{\pi}(1-p)H(\pi) + H(p) - H(p) = \max_{\pi}(1-p)H(\pi). \tag{24}$$

Logo, sua capacidade é dada por

$$C = 1 - p. \tag{25}$$

2.2.3 Canal Gaussiano

Dada uma variável aleatória binária contínua X com distribuição gaussiana $N(\mu, \sigma^2)$, com média μ e variância σ^2, a entropia será

$$h(x) = \frac{1}{2} log_2(2\pi \mathrm{e}\sigma^2)\ bits, \tag{26}$$

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

O canal Gaussiano é amplamente utilizado para modelar a transmissão real de informação, pois é inserido um ruído AWGN, que representa um ruído aditivo branco (se referindo à todas as faixas de frequência), e com distribuição gaussiana. Então, caso a variância desse ruído seja igual a 0, pode-se dizer que o sistema não contém a presença de ruído e toda informação é enviada e recebida sem erros. Logo, quando a variância chega em valores cada vez maiores, significa que a probabilidade da mensagem recebida conter erros aumenta. Existem técnicas para escolher conjuntos de entradas que minimizam as chances de erros na recepção da informação.

A variância nunca será igual a zero, dizer isso significa que todo sistema possui restrições de potência. Pela Figura 4, podemos extrair algumas definições através de processos estocásticos. Dado que $E[X^2 \leq P_s]$ e $E[Z^2] = \sigma^2 = N_r$, a saída é $Y = X + Z$, onde P_s é a potência do sistema e a entrada do canal e o ruído são independentes, logo

$$E[Y^2] = E[(X+Z)^2] = E[X^2] + 2E[XZ] + E[Z^2] = E[X^2] + E[Z^2] \leq P_s + N_r, \tag{27}$$

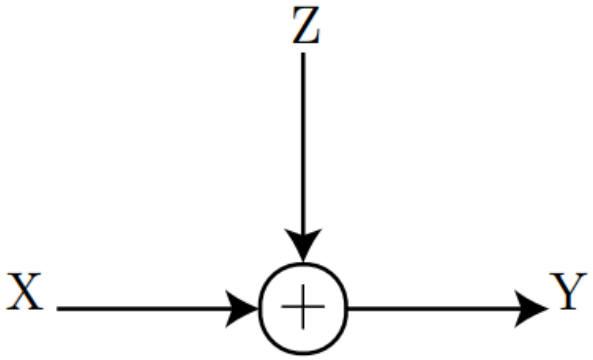
X e Z são independentes, reduzindo a informação mútua a

$$I(X;Y) = h(Y|X) = h(Y) - h(X+Z|X) = h(Y) - h(Z|X) = h(Y) - h(Z). \tag{28}$$

Portanto, as entropias relacionadas à capacidade do canal gaussiano são

$$h(Y) = \tfrac{1}{2} log_2(2\pi \mathrm{e}(P_s + N_r)), \tag{29}$$

Figura 4 – Canal Gaussiano



Fonte: Elaborado pelo autor..

$$h(Z) = \frac{1}{2} log_2(2\pi \mathrm{e} N_r). \tag{30}$$

Subtraindo uma da outra, temos

$$h(Y) - h(Z) = \frac{1}{2} log_2(2\pi \mathrm{e}(P_s N_r)) - \frac{1}{2} log_2(2\pi \mathrm{e} N_r) = \frac{1}{2} log_2 \left(\frac{P_s + N_r}{N_r} \right) = \frac{1}{2} log_2 \left(1 + \frac{P_s}{N_r} \right). \tag{31}$$

Então, a capacidade de um canal Gaussiano será

$$C_{gaussiano} = \max I(X;Y) = \frac{1}{2} log_2 \left(1 + \frac{P_s}{N_r} \right) \ bits/s. \tag{32}$$

Essa equação simplificada nos permite calcular a capacidade quando há limitação de banda. Se o canal for limitado em banda por $2B_c$, a capacidade será

$$2B_c \frac{1}{2} log_2 \left(1 + \frac{P_s}{N_r} \right) = B_c log_2 \left(1 + \frac{P_s}{N_r} \right) \ bits/s. \tag{33}$$

2.3 CÓDIGOS DE BLOCO LINEARES

Nesta seção os códigos de bloco lineares serão apresentados, com conceitos da construção da matriz geradora e matriz controle de paridade em suas formas sistemáticas. Esses conceitos são utilizados na construção de códigos polares no Capítulo 3 que é um tipo de código de bloco.

Definição 2.3.1 *Um código de bloco $\mathcal{C}(n,k)$, de comprimento n e dimensão k, transforma mensagens de comprimento k em palavras-código de comprimento n. Seja F_q um corpo finito com q elementos. Considerando o espaço vetorial F_q^n de dimensão n sobre F_q, então um código de bloco é dito linear, se e somente se, as q^k palavras-código formam um subespaço vetorial de dimensão k de F_q^n.*

Se $\mathcal{C}(n,k)$ é um código de bloco linear, então existem k vetores $\{g_0, ..., g_{k-1}\}$ linearmente independentes que formam uma base desse subespaço. Então, cada uma das palavras-código $c = [c_0, ..., c_{n-1}]$ $\in \mathcal{C}$ pode ser expressa como uma combinação linear

$$c = u_0 g_0 + ... + u_{k-1} g_{k-1}, \tag{34}$$

com o vetor de informação $u = [u_0, ..., u_{k-1}]$. Podemos enxergar a Equação 34, construindo uma matriz em que as linhas são formadas pelos vetores $\{g_0, ..., g_{k-1}\}$, obtendo o que chamamos de matriz geradora G do código de bloco linear. Esta codificação pode ser vista a partir da seguinte transformação linear

$$\mathcal{C} : F_q^k \to F_q^n. \tag{35}$$

dada por

$$c = u \cdot G. \tag{36}$$

Se a matriz geradora G está na forma

$$G = [I|P], \tag{37}$$

onde I é a matriz de identidade de dimensões $k \times k$, e P uma matriz de dimensões $k \times m$, que representa um conjunto de equações de verificação de paridade, então dizemos que a matriz geradora está na forma sistemática. Nesse caso, as palavras-código serão escritas inicialmente com o vetor de informação u seguido dos m bits de redundância, onde $m = n - k$. Uma codificação realizada a partir de uma matriz geradora na forma sistemática é chamada de codificação sistemática, caso contrário a codificação é dita não sistemática.

A partir da matriz geradora podemos obter a matriz controle de paridade, que é utilizada para realizar a decodificação de um código de bloco linear. Em uma codificação sistemática, a partir da Equação (37), a matriz controle de paridade H será dada por:

$$H = [P^T|I], \tag{38}$$

onde as novas dimensões de I é $k \times m$.

Se c é uma palavra-código então

$$H \cdot c^T = 0, \tag{39}$$

onde o zero da relação representa um vetor nulo com o comprimento de $n - k$. Os vetores resultantes dessa relação são chamados de síndrome de $\mathcal{C}$ através de H. Então, o vetor síndrome serve para checar se a palavra-código pertence ao código de bloco linear C, se a síndrome for um vetor nulo então a palavra-código pertence ao código de bloco linear.

Ao relacionar as matrizes geradoras e controle de paridade, podemos descrever por completo um código de bloco linear.

Quando consideramos códigos de blocos lineares, devemos levar em consideração duas métricas importantes, o peso e a distância de Hamming. Seja $u = (u_0, u_1, ..., u_n)$ um vetor em F_q^n. O peso de Hamming é descrito por $w_h(u)$, que é o número de componentes não-nulas nesse vetor. O menor peso de todas as palavras-código em um código de bloco, $w_{min}(\mathcal{C})$ ou simplesmente $w(\mathcal{C})$, é denominado peso mínimo de Hamming, e sua definição é dada por

$$w(\mathcal{C}) = min\{w(u) : v \in \mathcal{C}, v \neq 0\}. \tag{40}$$

A menor distância de um código de bloco linear $\mathcal{C}$, denotada por $d_{min}(\mathcal{C})$, é definida como a menor distância de Hamming entre duas palavras-código diferentes em $\mathcal{C}$, ou seja,

$$d_{min}(\mathcal{C}) = min\{d(u,v) : u, v \in \mathcal{C}, u \neq v\}. \tag{41}$$

Utilizando-se do fato de que $d(u,v) = w(u+v)$, podemos provar que $d(\mathcal{C}) = w(\mathcal{C})$, a partir de (41):

$$\begin{aligned} d_{min}(\mathcal{C}) &= min\{d(u,v) : u, v \in \mathcal{C}, u \neq v\} \\ &= min\{w(u+v) : u, v \in \mathcal{C}, u \neq w\} \\ &= min\{w(x) : x \in \mathcal{C}, x \neq 0\} \\ &= w(\mathcal{C}). \end{aligned} \tag{42}$$

Exemplo 2.3.1 *Seja um código de bloco linear $\mathcal{C} = \{000, 101, 010, 111\}$. A distância de Hamming entre cada um dos vetores é:*

$$\begin{aligned} &d_H(000,101) = 2, \quad && d_H(101,010) = 3, \quad && d_H(010,111) = 2. \\ &d_H(000,010) = 1, && d_H(101,111) = 1, \\ &d_H(000,111) = 3, \end{aligned}$$

Os resultados mostram que a distância mínima de Hamming desse código é $d(\mathcal{C}) = 1$. Então, o peso de Hamming será

$$w_H(000) = 0, \qquad w_H(101) = 2, \qquad w_H(010) = 1, \qquad w_H(111) = 3.$$

Portanto, o peso mínimo de Hamming desse código é $w(\mathcal{C}) = 1$. Também verificou-se que $d(\mathcal{C}) = w(\mathcal{C})$.

Adicionando a distância mínima d de um código de bloco linear $\mathcal{C}$, podemos agora caracterizar este código pelos parâmetros (n, k, d).

Outra característica importante é a capacidade de detecção de erros de um código de bloco linear. A detecção de erros de um código de bloco linear $\mathcal{C}$ com parâmetros (n, k, d) é de $(d-1)$ erros, enquanto que a capacidade de correção é de $\left\lfloor \frac{d-1}{2} \right\rfloor$ erros (LIN; COSTELLO, 1983).

3 CÓDIGOS POLARES

Este capítulo tem o objetivo de apresentar os códigos polares, introduzidos pela primeira vez em 2009 por Erdal Arikan. Sabe-se que na prática, nenhum canal consegue evitar totalmente os efeitos do ruído no sinal transmitido, entretanto, existe uma constante busca por métodos para a redução desses efeitos com o objetivo de aperfeiçoar os métodos de polarização de canal e buscar reproduzir fielmente os sinais transmitidos. Os códigos polares surgiram a partir da técnica de polarização de canal, que tem a finalidade de classificar quais são os canais mais ruidosos e menos ruidosos, usando os canais mais confiáveis para transmitir informação, ou seja, os canais com menor ruído e os menos confiáveis não transmitem informação.

3.1 POLARIZAÇÃO DE CANAL

Em seus estudos Arikan considerou um canal binário discreto e sem memória, *Binary-Discrete Memoryless Channel* (B-DMC). Um canal B-DMC é um canal de comunicação discreto com duas entradas e duas saídas, onde o canal não possui memória, ou seja, o comportamento do canal em determinado instante não depende das transmissões anteriores.

O canal é descrito por uma matriz de probabilidades de transição, que determina as probabilidades de saída para cada entrada possível. Um dos exemplos mais comuns de um B-DMC é o canal binário simétrico (BSC), onde a probabilidade de erro na transmissão de um bit é a mesma para 0 ou 1. Considerando um canal W com alfabeto de entrada X binário e alfabeto de saída Y, tal que:

$$W : X \longrightarrow Y. \tag{1}$$

É possível expandir esse canal para representar $N = 2^n, n \in Z^+$, usos de W, de forma que

$$W_N : X^N \longrightarrow Y^N, \tag{2}$$

e com probabilidade de transição dada por

$$W_N(y_1^N|x_1^N) = \prod_{i=1}^{N} W(y_i|x_i), \tag{3}$$

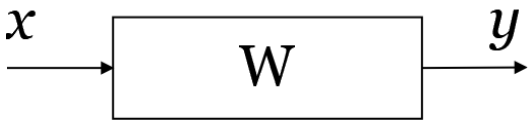
o parâmetro de maior interesse é a capacidade, dada por

$$I(W) \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}, \tag{4}$$

onde $I(W)$ é a taxa mais alta na qual a comunicação em W, cuja a frequência das entradas são iguais, é possível.

Para compreender o processo recursivo usado para realizar a combinação dos canais, inicialmente iremos utilizar o canal W_1, ou seja, $N = 2^0 = 1$ cópia do canal W, como mostrado na Figura 5 (BARNABé, 2020).

Figura 5 – Canal binário sem memória.



Fonte: elaborado pelo autor.

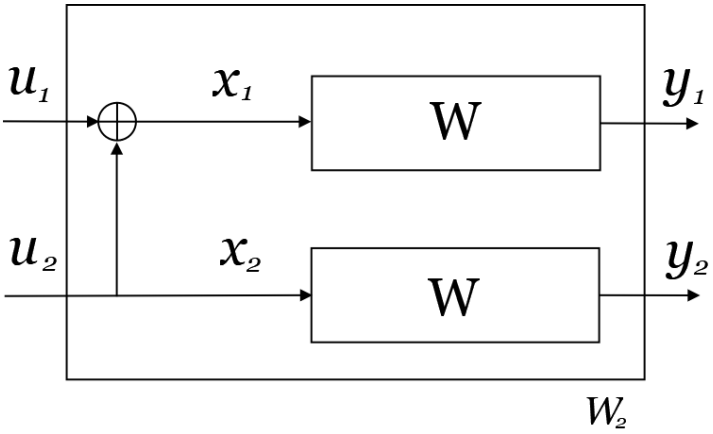
Para $N = 2$, ilustrado na (Figura 6), temos a combinação de 2 canais W, ou seja,

$$W_2 = X^2 \longrightarrow Y^2, \tag{5}$$

sendo as probabilidades de transição dadas por

$$W_2(y_1^2|u_1^2) = W(y_1|u_1 \oplus U_2)W(y_2|u_2). \tag{6}$$

Figura 6 – Combinação dos canais para $N = 2$.



Fonte: elaborado pelo autor.

Ao utilizar as probabilidades de transição, conseguimos obter as entradas x_1 e x_2, partindo-se de u_1 e u_2 e da matriz geradora, que representa as combinações desse sistema, sendo que, $\mathbf{x} = (x_1, x_2)$, $\mathbf{u} = (u_1, u_2)$ e G_2 é a matriz geradora, descrita a partir de:.

$$\mathbf{x}_1^2 = \mathbf{u}_1^2 G_2. \tag{7}$$

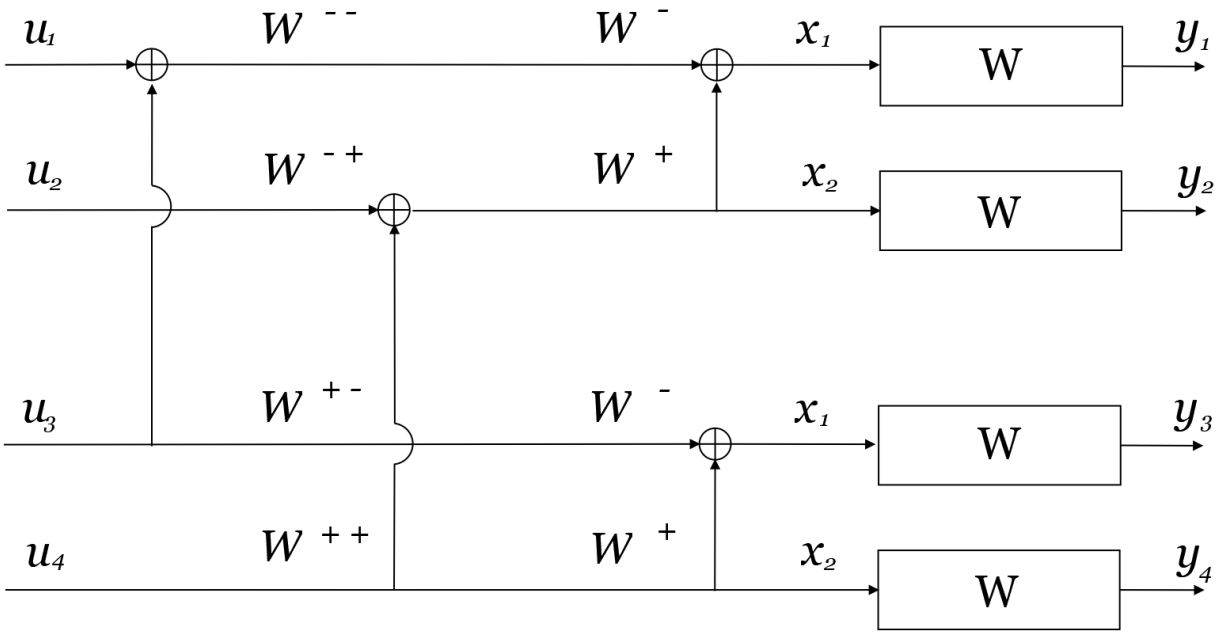
$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{8}$$

Quando passamos para um canal $N = 4$, ilustrado na Figura 7 teremos:

$$W_4 : X^4 \longrightarrow Y^4. \tag{9}$$

Nesse nível de recursão temos que as probabilidades de transição são:

Figura 7 – Canal W_4.



Fonte: elaborado pelo autor.

$$W_4(y_1^4|u_1^4) = W_2(y_1^2|u_1 \oplus u_2, u_3 \oplus u_4)W_2(y_3^4|u_2, u_4), \tag{10}$$

em que R_4 representa a operação de permutação que mapeia as entradas $\mathbf{s}_1^4 = (s_1, s_2, s_3, s_4)$ para $v_1^4 = (s_1, s_3, s_2, s_4)$, ou seja,

$$\mathbf{s}_1^4 \longrightarrow v_1^4. \tag{11}$$

Analogamente, a matriz geradora é dada por

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{12}$$

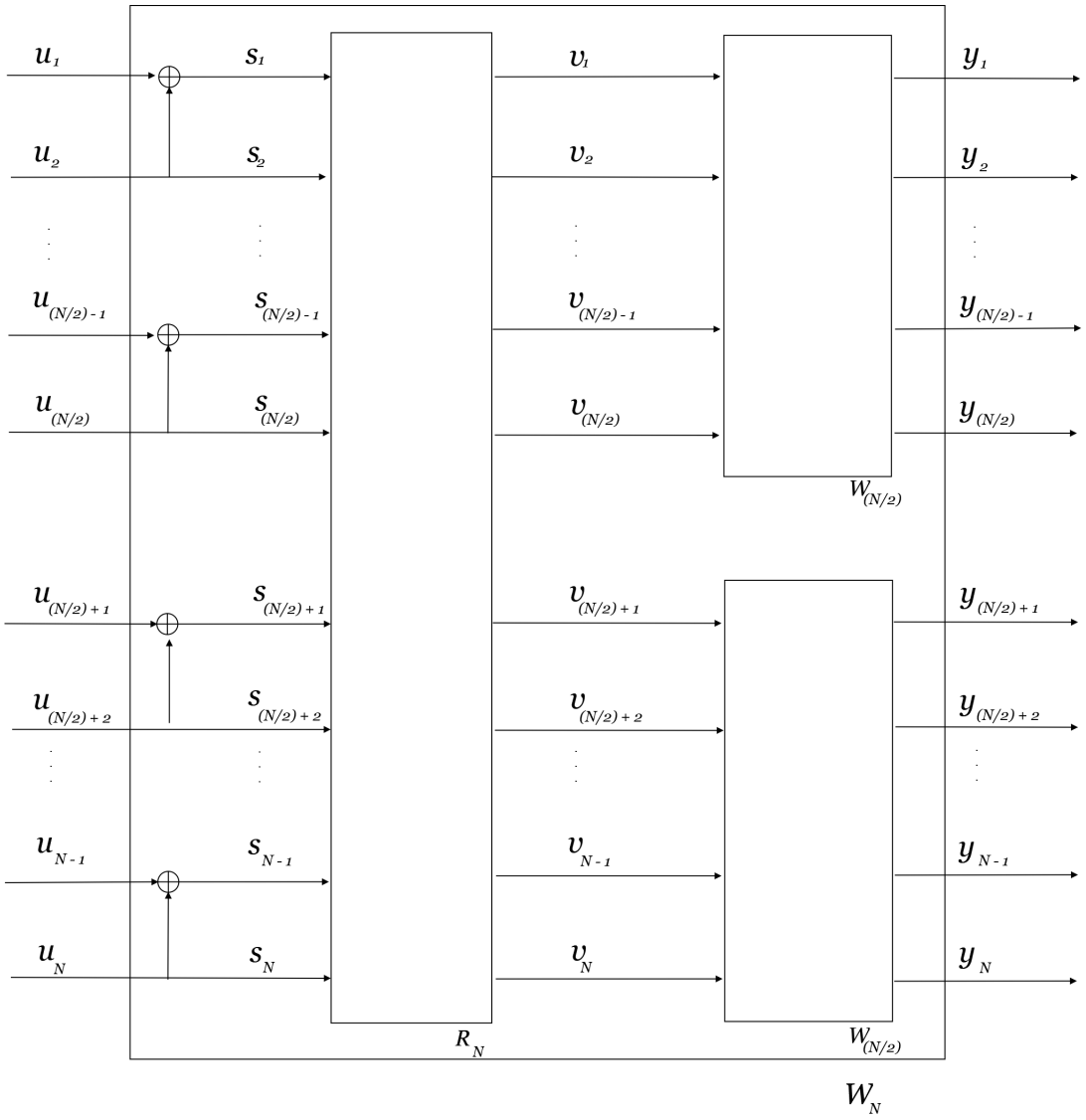
E, assim

$$\mathbf{x}_1^4 = u_1^4 G_4. \tag{13}$$

Podemos continuar aumentando o tamanho do canal e replicando as operações, temos assim uma generalização para esta recursão em um canal geral W_N, ilustrado na Figura 8, com $N = 2^n$, $n \geq 0$, de entradas $\mathbf{x}_1^N$ e saídas $\mathbf{y}_1^N$, com um bloco de permutação R_N e duas cópias de $W_{N/2}$, independentes entre si. É possível verificar que a primeira cópia independente de $W_{N/2}$ vai receber as entradas de índice ímpar, de forma semelhante, a segunda cópia independente de $W_{N/2}$ recebe somente as entradas de índice par, isso ocorre devido à permutação realizada por R_N. O vetor de entrada $\mathbf{u}_1^N$ em W_N é transformado em s_1^N, pela equação, desta vez porém com $(1 \leq i \leq N/2)$:

$$s_{2i-1} = u_{2i-1} \oplus u_{2i}, \ e \ s_{2i} = u_{2i}, \ \text{para } 1 \leq i \leq N/2. \tag{14}$$

Figura 8 – Cobinação de canais para $N = 8$.



Fonte: elaborado pelo autor.

O operador R_N realiza a permutação conhecida como embaralhamento reverso, onde s_1^N sofre uma transformação para v_1^N, que será a entrada das cópias de $W_{N/2}$:

$$s_1^N \longrightarrow v_1^N = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N). \tag{15}$$

Logo, podemos então, mapear o vetor $\mathbf{x}_1^N$ através de u_1^N e da matriz geradora

$$\mathbf{x}_1^N = u_1^N G_N. \tag{16}$$

A divisão de canais se dá através da divisão de W_N em um conjunto de canais de W_N^i,

$$W_N^{(i)} : X \longrightarrow Y^N \times X^{i-1},\ 1 \leq i \leq N, \tag{17}$$

definidos pelas probabilidades de transição,

$$W_N^{(i)}\left(y_1^N, u_1^{i-1}|u_i\right) \sum_{u_{i+1}^N \in X^{N-1}} \frac{1}{2^{N-1}} W_N\left(y_1^N|u_1^N\right), \tag{18}$$

onde $(y_1^N|u_1^N)$ são a saída e a entrada de $W_N^{(i)}$, respectivamente, com o propósito de definir quais serão os canais bons e quais serão os canais ruins para a comunicação.

O cálculo das probabilidades de transição para quaisquer tipos de canais não é simples. Para um canal BEC com probabilidade de apagamento e, Arikan apresentou os cálculos recursivos das capacidades para cada canal, como mostram as Equações (19) e (20):

$$I\left(W_N^{(2i-1)}\right) = I\left(W_i^{(N/2)}\right)^2, \tag{19}$$

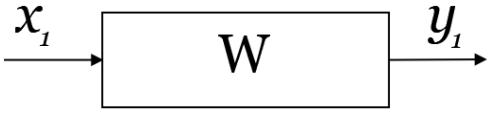
$$I\left(W_N^{(2i)}\right) = 2I\left(W_i^{(N/2)}\right) - I\left(W_i^{(N/2)}\right)^2, \tag{20}$$

em que, por (25), $I(W_1) = 1 - e$. No exemplo a seguir, os canais bons serão representados por W^+ e os canais ruins por W^-.

Exemplo 3.1.1 *Seja um canal BEC com probabilidade de apagamento $p = 0,3$ através do método de polarização de canal com $N = 1, 2, 4$ e 8. Para $N = 1$, podemos calcular a capacidade com o auxílio da Equação (25):*

$$I(W) = 1 - p = 1 - 0,3 = 0,7.$$

Figura 9 – Exemplo de canal W_1.



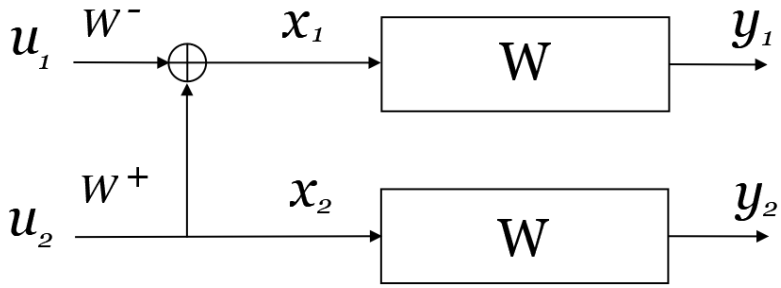
Fonte: elaborado pelo autor.

Agora, para $N = 2$ como na Figura 10, os canais W^- e W^+, dividem a capacidade total, logo tem-se

$$I(W^-) = I(W)^2 = 0,7^2 = 0,49,$$
$$I(W^+) = 2I(W) - I(W)^2 = 2(0,7) - 0,7^2 = 0,91.$$

Ao observar as duas capacidades, vemos que a capacidade de $I(W^-)$ é menor que a de $I(W^+)$. Logo, se considerássemos apenas 2 canais, o canal $I(W^-)$ seria congelado e a comunicação se daria através de $I(W^+)$. A Figura 10 ilustra a distribuição dos canais.
De maneira muito semelhante, para $N = 4$ como na Figura 11 a capacidade será dividida em 4 canais, dois canais congelados e dois canais para informação. Podemos determinar a capacidade de cada canal da seguinte forma:

Figura 10 – Exemplo de canal W_2.

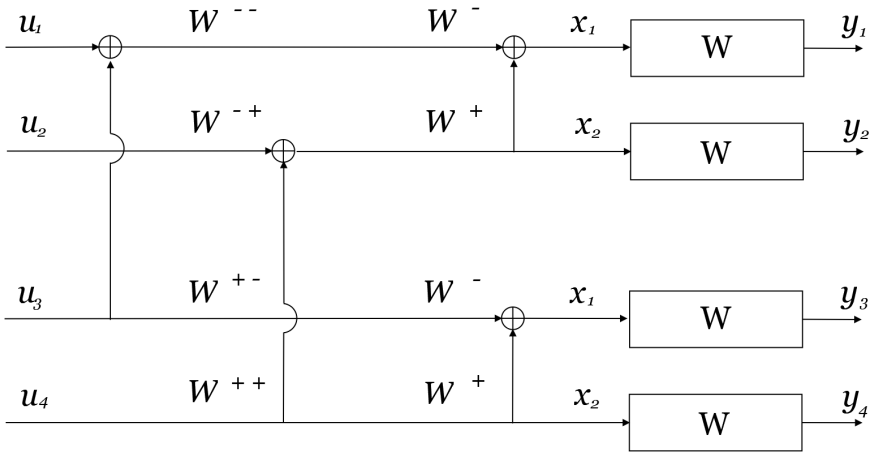


Fonte: elaborado pelo autor.

$$I(W^{--}) = I(W^-)^2 = (0,49)^2 = 0,2401,$$
$$I(W^{-+}) = 2I(W^-) - I(W^-)^2 = 2(0,49) - (0,49)^2 = 0,7399,$$
$$I(W^{+-}) = I(W^+)^2 = (0,91)^2 = 0,8281,$$
$$I(W^{++}) = 2I(W^+) - I(W^+)^2 = 2(0,91) - (0,91)^2 = 0,9919.$$

Verificamos que os canais $I(W^{+-})$ e $I(W^{++})$ possuem as maiores capacidades. Portanto, se 4 canais fossem considerados na transmissão, a comunicação se daria através destes, enquanto que $I(W^{--})$ e $I(W^{-+})$ serão congelados. A Figura 11 ilustra a distribuição dos canais.

Figura 11 – Exemplo de canal W_4.



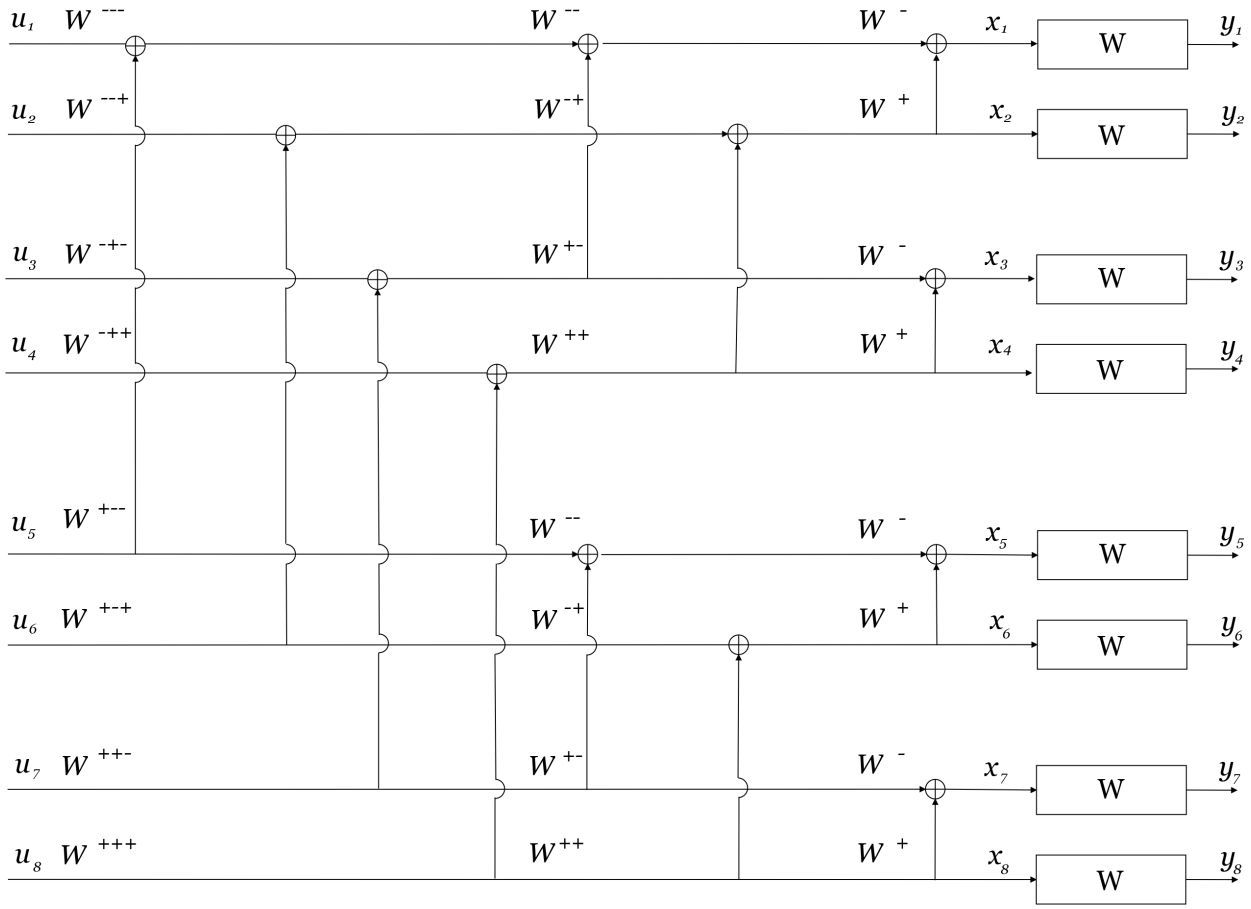
Fonte: elaborado pelo autor.

Por fim, $N = 8$, a capacidade será dividida em 8 canais, e de maneira similar à anterior, podemos calcular as mesmas:

$$I(W^{---}) = I(W^{--})^2 = (0,2401)^2 = 0,0576801,$$
$$I(W^{--+}) = 2I(W^{--}) - I(W^{--})^2 = 2(0,2401) - (0,2401)^2 = 0,42255199,$$
$$I(W^{-+-}) = I(W^{-+})^2 = (0,7399)^2 = 0,54745201,$$
$$I(W^{-++}) = 2I(W^{-+}) - I(W^{-+})^2 = 2(0,7399) - (0,7399)^2 = 0,93234799,$$
$$I(W^{+--}) = I(W^{+-})^2 = (0,8281)^2 = 0,68574961,$$
$$I(W^{+-+}) = 2I(W^{+-}) - I(W^{+-})^2 = 2(0,8281) - (0,8281)^2 = 0,97045039,$$
$$I(W^{++-}) = I(W^{++})^2 = (0,9919)^2 = 0,98386561,$$
$$I(W^{+++}) = 2I(W^{++}) - I(W^{++})^2 = 2(0,9919) - (0,9919)^2 = 0,99993439.$$

Observamos que os canais com as menores capacidades são $I(W^{---})$, $I(W^{--+})$, $I(W^{-+-})$ e $I(W^{+--})$, já os canais $I(W^{-++})$, $I(W^{+-+})$, $I(W^{++-})$ e $I(W^{+++})$ possuem as maiores capacidades, portanto irão carregar informação e os demais serão congelados. A Figura 12 ilustra a distribuições dos canais.

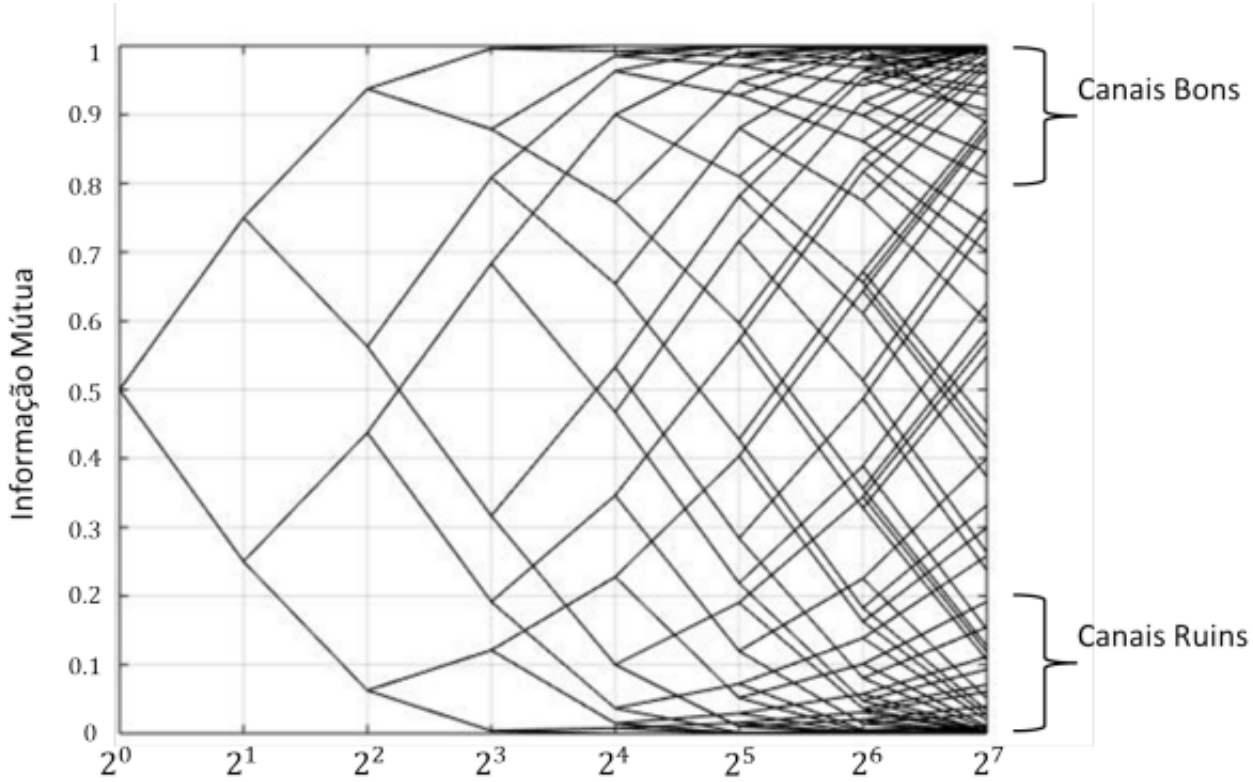
Figura 12 – Exemplo de canal W_8.



Fonte: elaborado pelo autor.

Pelo exemplo anterior, podemos verificar que conforme aumentamos o número de canais, as capacidades dos 2^N canais tende a 0 ou a 1, onde os canais que tendem a zero representam os canais ruins. Podemos observar esse comportamento pela Figura 13.

Figura 13 – Informação mútua pelo número de canais.



Fonte: retirado de (TERçAS, 2019).

3.2 CODIFICAÇÃO

Esse capítulo apresenta os dois tipos de codificação de códigos polares. Primeiramente iremos abordar a codificação não sistemática e posteriormente a sistemática.

3.2.1 Codificação não sistemática

A codificação não sistemática é realizada diretamente a partir da estratégia de combinação de canal para definir a polarização, como apresentado na Seção 3.1. E, assim como um código de bloco linear, pode ser esquematizada através da multiplicação d vetor de informação u_1^N com a matriz geradora G_N:

$$\mathbf{x}_1^N = \mathbf{u}_1^N G_N, \tag{21}$$

sendo $\mathbf{x}_1^N = (x_1, x_2, \ldots, x_N) \in X^N$ o vetor palavra-código, em que $N = 2^n$, para $n \geq 0$. A polarização visa escolher os melhores canais para transmissão de informação, portanto, a codificação também deve considerar os canais congelados e os canais de informação, sendo assim, podemos reescrever a Equação 21 como

$$\mathbf{x}_1^N = \mathbf{u}G_N = \mathbf{u}_A G_A \oplus \mathbf{u}_{A^c} G_{A^c}, \tag{22}$$

em que

- G_A é uma submatriz de G_N, construída pelas linhas de índices em A, $(u_A : u_i \in A)$.
- G_{A^c} é uma submatriz de G_N, construída pelas linhas de índices em A^c, $(u_{A^c} : u_j \in A^c)$.
- $u_{A^c} = u - u_A$.

Logo, um código polar não-sistemático pode ser definido utilizando os parâmetros (N, k, A, μ_{A^c}), onde em que N é o comprimento da palavra-código, k é a dimensão do código que define o tamanho do conjunto A formado pelas linhas de G_N não congeladas e as linhas congeladas são representadas por μ_{A^c}. Uma observação importante, é que as entradas de bits de informação e bits congelados são representados por u_i e u_j, que pertencem ao conjunto A e A^c respectivamente.

Uma outra forma de olhar para esta codificação é obter a matriz geradora utilizando um produto tensorial de matrizes, também conhecido como produto de Kroneckera. Para isso, iniciamos com $N = 2$, em que como vimos na Seção 2.3, a matriz geradora de dimensões 2×2 é expressa como:

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = F. \tag{23}$$

Para definir uma matriz geradora de dimensão $N \times N$ fazemos:

$$G_N = B_N F^{\otimes n}, \tag{24}$$

em que B_N é a matriz permutação de inversão de bit dada por

$$B_N = R_N(B_{N/2} \otimes I_2) \tag{25}$$

e $F^{\otimes n}$ é o produto de Kronecker definido por

$$F^{\otimes n} = F \otimes F \otimes \ldots F \ (n \ vezes). \tag{26}$$

Para simplificar a implementação de códigos polares, é mais comum a construção direta da matriz geradora como:

$$G_N = F^{\otimes n} = \begin{bmatrix} F^{\otimes n-1} & 0 \\ F^{\otimes n-1} & F^{\otimes n-1} \end{bmatrix}, \tag{27}$$

sendo que, por definição, $F^{\otimes 1} = F$ e $F^{\otimes 0} = 1$. Assim, se utilizamos a matriz como em (27) para a implementação dos códigos polares, a equação de codificação torna-se:

$$\mathbf{x}_1^N = \tilde{\mathbf{u}}_1^N F^{\otimes n}, \tag{28}$$

onde $\tilde{\mathbf{u}}_1^N$ é a mensagem reordenada após sua passagem pela matriz de permutação.

A Equação 29 será desenvolvida de forma a evidenciar a matriz G_N:

$$G_N = F^{\otimes n} = \begin{bmatrix} F^{\otimes n-1} & 0 \\ F^{\otimes n-1} & F^{\otimes n-1} \end{bmatrix} = \begin{bmatrix} G_{N/2} & 0_{N/2} \\ G_{N/2} & G_{N/2} \end{bmatrix}. \tag{29}$$

Exemplo 3.2.1 *Dada a mensagem* $\mathbf{u} = (1010)$*, queremos codificá-la através de um codificador polar não-sistemático com parâmetros* $(8, 4, \{4, 6, 7, 8\}, (0, 0, 0, 0))$.

Identificação dos parâmetros do código:

- $N = 8$*: comprimento do código,*
- $k = 4$*: dimensão do código,*
- $A = \{4, 6, 7, 8\}$*: índice das posições de informação.*

Construção da matriz permutação B_N

Sabemos, por (24), que $G_N = B_N F^{\otimes n}$*. Como* $N = 8$*, temos* $G_8 = B_8 F^{\otimes 3}$*. A matriz permutação* B_8 *pode ser calculada recursivamente:*

$$B_N = R_N(B_{N/2} \otimes I_2).$$

Partindo de B_2*:*

$$B_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Para B_4*:*

$$B_4 = R_4(B_2 \otimes I_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Para B_8*:*

$$B_8 = R_8(B_4 \otimes I_2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Construção da matriz geradora G_8

Agora podemos calcular a matriz geradora utilizando $G_8 = B_8 F^{\otimes 3}$. Sabemos que:

$$F^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Assim:

$$G_8 = B_8 F^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

O resultado é:

$$G_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Agora, expandimos $\mathbf{u} = (1010)$ *para* $\tilde{\mathbf{u}} = (00001010)$*, adicionando os bits congelados nas posições fora de* A*. Assim:*

$$\mathbf{x}_1^8 = \tilde{\mathbf{u}} G_8 = [00001010] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Logo, a mensagem codificada é:

$$\mathbf{x}_1^8 = [11101101].$$

3.2.2 Codificação sistemática

Em 2011 Arikan propôs a codificação sistemática de códigos polares, visto que um dos obstáculos para o uso das implementações em *hardware* dos codificadores não sistemáticos é o alto custo computacional. Como vimos para códigos de bloco lineares quaisquer, a principal característica que diferencia uma palavra-código proveniente de um codificador sistemático de um codificador polar não sistemático é que nela os bits do vetor de informação aparecem de maneira explícita na palavra-código. No caso de códigos polares, os bits não congelados aparecem de maneira explícita na palavra-código.

A codificação sistemática parte da Equação 22 utilizada na codificação não sistemática, porem a palavra-código é dividida em $\mathbf{x} = x_B + x_{B^c}$, sendo B um sub-conjunto arbitrário de $\{1, \ldots, N\}$, assim, temos que

$$\mathbf{x}_B = \mathbf{u}_A G_{AB} \oplus \mathbf{u}_{A^c} G_{A^c B}, \tag{30}$$

e

$$\mathbf{x}_{B^c} = \mathbf{u}_A G_{AB^c} \oplus \mathbf{u}_{A^c} G_{A^c B^c}, \tag{31}$$

onde G_{AB} é uma submatriz de G, que consiste do vetor de elementos $G_{i,j}$, $i \in A$, $j \in B$. Para as outras submatrizes utilizaremos os mesmos passos. O objetivo é encontrar alguns codificadores sistemáticos nos quais $\mathbf{x}_B$ irão desempenhar o mesmo papel de $\mathbf{u}_A$ nos decodificadores não sistemáticos, porém $\mathbf{u}_{A^c}$ será fixo.

Seja um código polar definido por um codificador não sistemático com os parâmetros (N, k, A, μ_{A^c}), há um codificador sistemático com parâmetros (N, k, B, μ_{A^c}), se, e somente se, A e B tiverem a mesma quantidade de elementos e G_{AB} for uma matriz inversível. Se essas condições são satisfeitas, o mapeamento de $\mathbf{x}_B \mapsto \mathbf{x} = (\mathbf{x}_B, \mathbf{x}_{B^c})$ é realizado através do cálculo de

$$\mathbf{u}_A = (\mathbf{x}_B - \mathbf{u}_{A^c} G_{A^c B})(G_{AB})^{-1}. \tag{32}$$

Logo, o resultado obtido em (32) será utilizado para o cálculo de (31).

Observando (29) vemos que a matriz geradora é uma matriz triangular inferior, na qual a diagonal principal é composta de 1. Além disso, uma matriz inversível e que ${G_N}^{-1} = G_N$. Toda submatriz $(G_N)_{AA}$ de G_N, com $A \subset \{1, \ldots, N\}$ também é uma matriz triangular inferior e possui uns em sua diagonal principal, logo também é inversível.

Ao separar a palavra-código sistemática em duas partes, $\mathbf{x}^{(1)}$ sendo a metade inicial e $\mathbf{x}^{(2)}$, a metade final da palavra. Com isso, temos que

$$\mathbf{x} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right), \tag{33}$$

de forma que,

$$\mathbf{x}^{(1)} = (x_1, \ldots, x_{N/2}),$$
$$\mathbf{x}^{(2)} = (x_{N/2+1}, \ldots, x_N).$$

Ao estender esse raciocínio a mensagem $\mathbf{u}$, reescrevemos como:

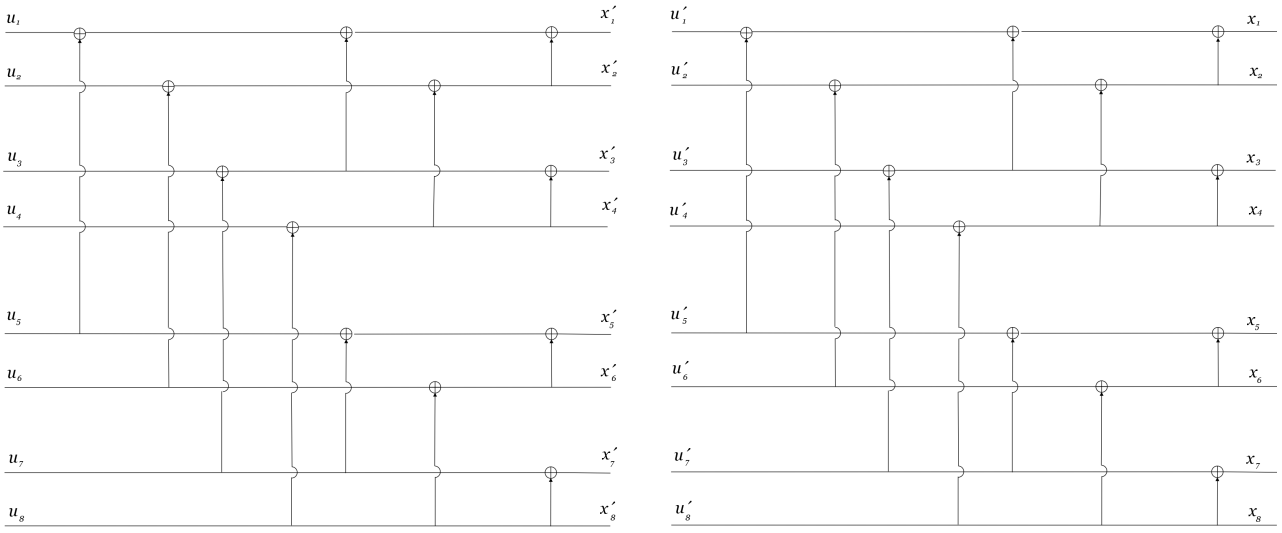
$$\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = \left(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}\right) \begin{bmatrix} G_{N/2} & 0_{N/2} \\ G_{N/2} & G_{N/2} \end{bmatrix}. \tag{34}$$

Então, $\mathbf{x}^{(1)}$ e $\mathbf{x}^{(2)}$ são definidos por:

$$\left(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\right) = \mathbf{u}^{(1)} G_{N/2}. \tag{35}$$

Outra maneira de executar a codificação polar sistemática é através do uso de dois codificadores não sistemáticos em série. A principio este codificador polar possui a mensagem $\mathbf{u} = (u_1, u_2, \ldots, u_N)$, que é a entrada do primeiro subcircuito. As posições dos bits congelados definem as linhas congeladas do circuito completo. A Figura 14 ilustra o circuito do codificador polar sistemático para o caso no qual $N = 8$.

Figura 14 – Codificador sistemático para $N = 8$.



Fonte: Elaborado pelo autor.

Na primeira etapa é realizada a operação de multiplicação da mensagem $\mathbf{u}$ pela matriz $F^{\otimes n}$, assim o vetor $\mathbf{x}'$ é a saída do primeiro circuito:

$$\mathbf{x}' = \mathbf{u} F^{\otimes n}. \tag{36}$$

Logo após, os bits de $\mathbf{x}'$ provenientes das entradas congeladas, ou seja, que não pertencem à A, são igualados a zero, e o novo vetor, denominado de $\mathbf{u}'$, será a entrada do segundo subcircuito. Com isso, podemos obter a palavra-código sistemática através de

$$\mathbf{x} = \mathbf{u}' F^{\otimes n}. \tag{37}$$

Exemplo 3.2.2 *Considerando um código polar sistemático com parâmetros*

$$(8, 4, \{4, 6, 7, 8\}, (0, 0, 0, 0)),$$

desejamos utilizar as equações recursivas de Arikan para realizar a codificação polar sistemática da mensagem $\mathbf{u} = (0101)$. *Inicialmente, precisamos definir, para este caso, as matrizes apresentadas nas Equações (30) e (31):* G_{AB}, G_{AB^c}, G_{A^cB} *e* $G_{A^cB^c}$, *a partir de* G_8. *Temos que:*

$$(G_8)_{AB} = \begin{bmatrix} G_{8_{(4,4)}} & G_{8_{(4,6)}} & G_{8_{(4,7)}} & G_{8_{(4,8)}} \\ G_{8_{(6,4)}} & G_{8_{(6,6)}} & G_{8_{(6,7)}} & G_{8_{(6,8)}} \\ G_{8_{(7,4)}} & G_{8_{(7,6)}} & G_{8_{(7,7)}} & G_{8_{(7,8)}} \\ G_{8_{(8,4)}} & G_{8_{(8,6)}} & G_{8_{(8,7)}} & G_{8_{(8,8)}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$(G_8)_{AB^c} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$(G_8)_{A^cB} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$(G_8)_{A^cB^c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Sabemos que $\mathbf{x}_B$, *por se tratar de uma codificação sistemática, obrigatoriamente terá os mesmos elementos das linhas não congeladas da mensagem original. Como* $\mathbf{u} = (0101)$, *definimos* $\mathbf{x}_B = (u_4\ u_6\ u_7\ u_8) = (0101)$. *De forma semelhante, sabemos que* $\mathbf{u}_{A^c}$ *é definida pelas linhas congeladas da mensagem original, logo* $\mathbf{u}_{A^c} = (0000)$. *Assim, temos todos os elementos necessários para calcular* $\mathbf{u}_A$*:*

$$\begin{aligned}\mathbf{u}_A &= (\mathbf{x}_B - \mathbf{u}_{A^c} G_{A^c B})(G_{AB})^{-1} \\ &= \left(0101 - 0000 \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \\ &= (0101) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = (0001).\end{aligned}$$

Finalmente, substituímos $\mathbf{u}_A$ *em* $\mathbf{x}_{B^c}$*:*

$$\begin{aligned}\mathbf{x}_{B^c} &= \mathbf{u}_A G_{AB^c} \oplus \mathbf{u}_{A^c} G_{A^c B^c} \\ &= (0001) \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \oplus (0000) \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = (0000).\end{aligned}$$

Portanto, a palavra-código sistemática é $\mathbf{x} = (00010101)$.

Observando o resultado do Exemplo (3.2.2) podemos ver claramente a informação contida na palavra-código.

3.3 DECODIFICAÇÃO

Será apresentado neste capítulo a decodificação de códigos polares (SC, *Successive Cancellation*). Posteriormente a decodificação de códigos polares através da decodificação por cancelamento sucessivo em lista (SLC, *Successive Cancellation List*).

3.3.1 Decodificação SC

Sejam a palavra-código $\mathbf{x}_1^N$ definida por um codificador polar e $\mathbf{y}_1^N$ a mensagem recebida após a transmissão através de um canal W_N. O decodificador por cancelamento sucessivo SC determina a mensagem $\hat{\mathbf{u}}_1^N$, como uma estimativa da mensagem original $\mathbf{u}_1^N$, a partir da mensagem recebida $(\mathbf{y}_1^N)$ e as posições dos bits congelados $\mathbf{u}_{A^c}$. Nessa etapa, é possível notar a real importância da polarização dos canais, pois devido ao conhecimento prévio das linhas congeladas, conseguimos evitar erros de decodificação com mais facilidade, logo, o verdadeiro desafio é determinar $\hat{\mathbf{u}}_A$ igual a $\mathbf{u}_A$. Para isso, teremos que:

$$\hat{\mathbf{u}}_i = \begin{cases} \mathbf{u}_i, \ se \ i \in A^c, \\ L_N^{(i)}(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}) \ se \ i \in A, \end{cases} \tag{38}$$

em que $L_N^{(i)}(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1})$ é a razão de verossimilhança (LR, *Likelihood Ratio*), dada por:

$$L_N^{(i)}(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}) = \begin{cases} 0, \ se \ \frac{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=0)}{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=1)} \geq 1. \\ 1, \ se \ \frac{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=0)}{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=1)} < 1. \end{cases} \tag{39}$$

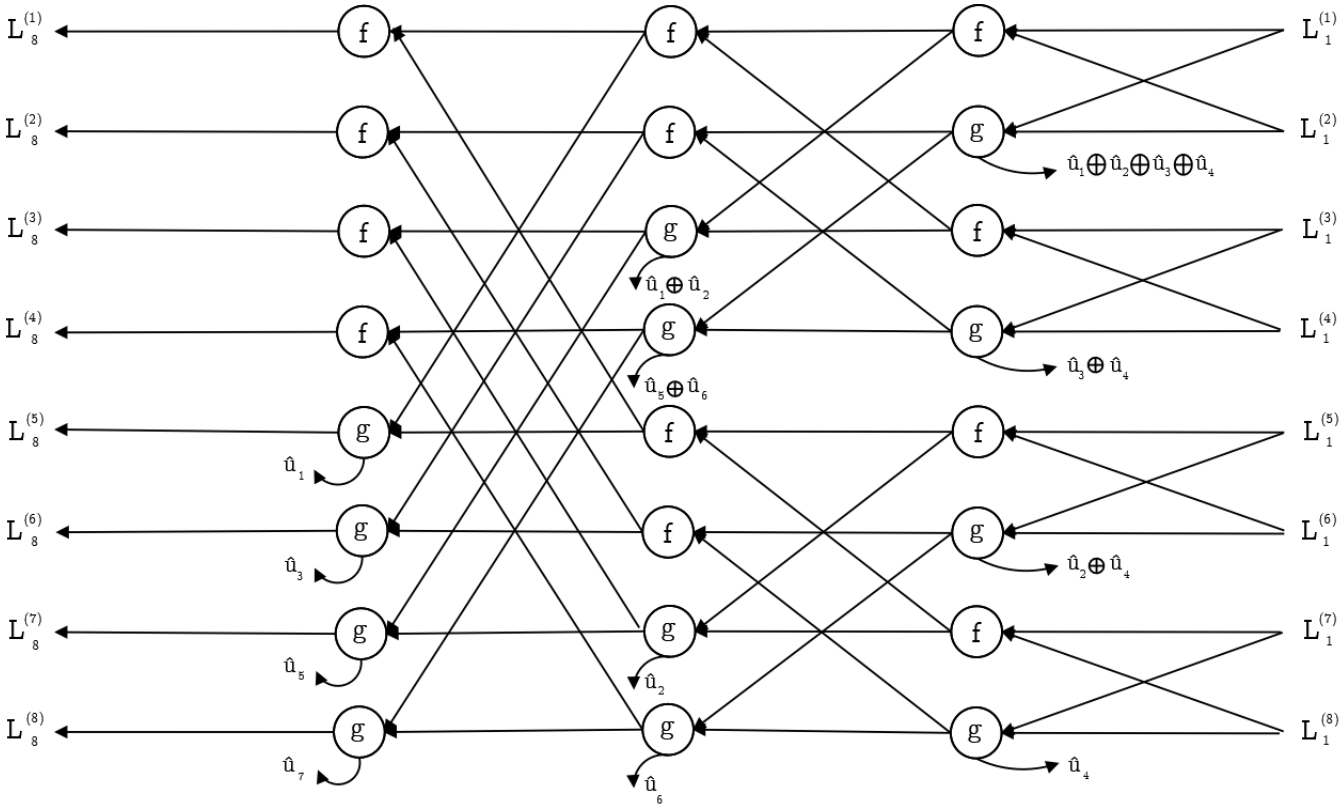
Para encontrar a LR utilizamos as seguintes relações de recursão:

$$f(a,b) = \frac{(ab)+1}{a+b}, \tag{40}$$

$$g(a,b,\hat{u}_i) = a^{1-2\hat{u}_{i-1}}b. \tag{41}$$

Dessa maneira, a partir de (39), é construída uma rede de fluxo de razão de verossimilhança, ou seja, um decodificador por cancelamento sucessivo, onde atualizamos as LR's ao longo das rotas, como ilustrado na Figura 15.

Figura 15 – Decodificador por cancelamento sucessivo para um bloco de tamanho $N = 8$.



Fonte: elaborado pelo autor.

Exemplo 3.3.1 *Uma transmissão obteve o vetor* $\mathbf{y}$*:*

$$\mathbf{y} = (1,883; 2,4848; -2,2381; 2,4803; 0,7942; -2,4148; -1,3290; 0,2813).$$

Sabemos que o código é definido pelos parâmetros $(8, 4, \{4, 6, 7, 8\}), (0, 0, 0, 0)$ *e que a transmissão foi realizada através de um canal AWGN no qual* $E_b/N_0 = 1\ dB$*. Deseja-se decodificar a mensagem a fim de identificar a mensagem original transmitida. Podemos utilizar as Equações 40, 41, 38 e 39 e decodificar a mensagem recebida através do método de cancelamento sucessivo. A* **??** *nos apresenta o sentido da decodificação e as operações que são realizadas em cada uma das etapas da decodificação.*

- ***Etapa 1****: inicialmente, determinamos os nós de primeiro estágio possíveis de serem calculados somente com as informações originais.*
  - *Cálculo de* $L_2^{(1)}$*:*

$$L_2^{(1)} \to f\left(L_1^{(1)}, L_1^{(2)}\right) = f(1,8883; 2,4848) = \frac{(1,8883 \cdot 2,4848) + 1}{1,8883 + 2,4848} = 1,3016.$$

  - *Cálculo de* $L_2^{(3)}$*:*

$$L_2^{(3)} \to f\left(L_1^{(3)}, L_1^{(4)}\right) = f(-2,2381; 2,4803) = \frac{(-2,2381 \cdot 2,4803) + 1}{-2,2381 + 2,4803} = -18,7909.$$

  - *Cálculo de* $L_2^{(5)}$*:*

$$L_2^{(5)} \to f\left(L_1^{(5)}, L_1^{(6)}\right) = f(0,7942; -2,4148) = \frac{(0,7942 \cdot (-2,4148)) + 1}{0,7942 + (-2,4148)} = 0,5664.$$

  - *Cálculo de* $L_2^{(7)}$*:*

$$L_2^{(7)} \to f\left(L_1^{(7)}, L_1^{(8)}\right) = f(-1,3290; 0,2813) = \frac{(-1,3290 \cdot 0,2813) + 1}{-1,3290 + 0,2813} = -0,5976.$$

Determinamos então os nós de segundo estágio possíveis de serem calculados com as informações obtidas:

  - *Cálculo de* $L_4^{(1)}$*:*

$$\begin{aligned} L_4^{(1)} \to f\left(L_2^{(1)}, L_2^{(3)}\right) &= f(1,3016; -18,7909) \\ &= \frac{(1,3016 \cdot (-18,7909)) + 1}{1,3016 + (-18,7909)} = 1,3413. \end{aligned}$$

  - *Cálculo de* $L_4^{(5)}$*:*

$$\begin{aligned} L_4^{(5)} \to f\left(L_2^{(5)}, L_2^{(7)}\right) &= f(1,3016; -18,7909) \\ &= \frac{(1,3016 \cdot (-18,7909)) + 1}{1,3016 + (-18,7909)} = -21,2025. \end{aligned}$$

Em posse dos elementos determinados anteriormente, somos capazes de calcular um dos nós de terceiro estágio:

– *Cálculo de* $L_8^{(1)}$*:*

$$L_8^{(1)} \to f\left(L_4^{(1)}, L_4^{(5)}\right) = f(1,3413; -21,2025)$$
$$= \frac{(1,3413 \cdot (-21,2025)) + 1}{1,3413 + (-21,2025)} = 1,3413.$$

Como $L_8^{(1)} = 1,3413 > 1$*, logo* $\hat{u}_1 = 0$*.*

- ***Etapa 2****: as informações obtidas na Etapa 1 nos permitem determinar outro nó de terceiro estágio.*

  – *Cálculo de* $L_8^{(5)}$*:*

$$L_8^{(5)} \to g\left(L_4^{(1)}, L_4^{(5)}, \hat{u_1}\right) = 1,3413^{(1-2(0))}(-21,2025) = -28,4389.$$

Apesar de $L_8^{(5)} = -28,4389$*, temos conhecimento de que se trata de um bit congelado, portanto* $\hat{u}_2 = 0$*.*

- ***Etapa 3****: utilizando elementos calculados na Etapa 1, somos capazes de determinar outros nós de segundo estágio.*

  – *Cálculo de* $L_4^{(3)}$*:*

$$L_4^{(3)} \to g\left(L_2^{(1)}, L_2^{(3)}, \hat{u_2}\right) = 1,3016^{(1-2(0))}(-18,7909) = -24,4582.$$

  – *Cálculo de* $L_4^{(7)}$*:*

$$L_4^{(7)} \to g\left(L_2^{(5)}, L_2^{(7)}, \hat{u_2}\right) = 0,5664^{(1-2(0))}(-0,5976) = -0,3385.$$

Os nós $L_4^{(3)}$ *e* $L_4^{(7)}$ *fornecem-nos as informações necessárias para a determinação de* $L_8^{(3)}$*:*

  – *Cálculo de* $L_8^{(3)}$*:*

$$L_8^{(3)} \to f\left(L_4^{(3)}, L_4^{(7)}\right) = f(-24,4582; -0,3385)$$
$$= \frac{(-24,4582 \cdot (-0,3385)) + 1}{-24,4582 + (-0,3385)} = -0,3742.$$

Apesar de $L_8^{(3)} = -0,3385$*, sabemos que se trata de um bit congelado, portanto* $\hat{u}_3 = 0$*.*

- ***Etapa 4****: as informações obtidas na Etapa 3 nos permitem calcular o nó de terceiro estágio* $L_8^{(7)}$*.*

  – *Cálculo de* $L_8^{(7)}$*:*

$$L_8^{(7)} \to g\left(L_4^{(3)}, L_4^{(7)}, \hat{u}_3\right) = g(-24,4582; -0,3385, 0)$$
$$= -24,4582^{(1-2(0))}(-0,3385) = 8,2791.$$

Como $L_8^{(7)} = 8,2791 > 1$*, logo* $\hat{u}_4 = 0$*.*

- ***Etapa 5****: calculamos os demais nós de primeiro estágio com as informações obtidas anteriormente.*

  – *Cálculo de* $L_2^{(2)}$*:*

$$\begin{aligned} L_2^{(2)} &\to g\left(L_1^{(1)}, L_1^{(2)}, \hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3 \oplus \hat{u}_4\right) \\ &= g(1,8813; 2,4848; 0 \oplus 0 \oplus 0 \oplus 0) = g(1,8813; 2,4848; 0) \\ &= 1,8813^{(1-2(0))}(2,4848) = 4,6920. \end{aligned}$$

  – *Cálculo de* $L_2^{(4)}$*:*

$$\begin{aligned} L_2^{(4)} &\to g\left(L_1^{(3)}, L_1^{(4)}, \hat{u}_3 \oplus \hat{u}_4\right) \\ &= g(-2,2381; 2,4803; 0 \oplus 0) = g(-2,2381; 2,4803; 0) \\ &= -2,2381^{(1-2(0))}(2,4803) = -5,5512. \end{aligned}$$

  – *Cálculo de* $L_2^{(6)}$*:*

$$\begin{aligned} L_2^{(6)} &\to g\left(L_1^{(5)}, L_1^{(6)}, \hat{u}_2 \oplus \hat{u}_4\right) \\ &= g(0,7942; -2,4148; 0 \oplus 0) = g(0,7942; -2,4148; 0) \\ &= -2,2381^{(1-2(0))}(2,4803) = -1,9178. \end{aligned}$$

  – *Cálculo de* $L_2^{(8)}$*:*

$$\begin{aligned} L_2^{(8)} &\to g\left(L_1^{(7)}, L_1^{(8)}, \hat{u}_4\right) = g(0,7942; -2,4148; 0) \\ &= -1,3290^{(1-2(0))}(0,2813) = -0,3738. \end{aligned}$$

Utilizando os valores obtidos para os nós de primeiro estágio, calculamos outros nós de segundo estágio:

  – *Cálculo de* $L_4^{(2)}$*:*

$$\begin{aligned} L_4^{(2)} &\to f\left(L_2^{(2)}, L_2^{(4)}\right) = f(4,68920; -5,5512) \\ &= \frac{(4,68920 \cdot (-5,5512)) + 1}{(4,68920 + (-5,5512))} = 29,1506. \end{aligned}$$

  – *Cálculo de* $L_4^{(6)}$*:*

$$\begin{aligned} L_4^{(6)} &\to f\left(L_2^{(6)}, L_2^{(8)}\right) = f(-1,9178; -0,3738) \\ &= \frac{(-1,9178 \cdot (-0,3738)) + 1}{-1,9178 + (-0,3738)} = -0,7492. \end{aligned}$$

Com os valores de $L_4^{(2)}$ e $L_4^{(6)}$ podemos calcular o nó $L_8^{(2)}$, de terceiro estágio:

- *Cálculo de $L_8^{(2)}$:*

$$L_8^{(2)} \to f\left(L_4^{(2)}, L_4^{(6)}\right) = f(29,1506; -0,7492)$$
$$= \frac{(29,1506 \cdot (-0,7492)) + 1}{29,1506 + (-0,7492)} = -0,7338.$$

Apesar de $L_8^{(2)} = -0,7338 < 1$, sabemos que se trata de uma linha congelada, portanto $\hat{u}_5 = 0$.

- ***Etapa 6**: podemos determinar outro nó de terceiro estágio com informações obtidas na Etapa 5.*

  - *Cálculo de $L_8^{(6)}$:*

$$L_8^{(6)} \to g\left(L_4^{(2)}, L_4^{(6)}, \hat{u}_5\right) = g(29,1506; -0,7492; 0)$$
$$= 29,1506^{(1-2(0))}(-0,7492) = -21,8396.$$

Como $L_8^{(6)} = -21,8396 < 1$, logo $\hat{u}_6 = 1$.

- ***Etapa 7**: calculamos os nós restantes de segundo estágio:*

  - *Cálculo de $L_4^{(4)}$:*

$$L_4^{(4)} \to g\left(L_2^{(2)}, L_2^{(4)}, \hat{u}_5 \oplus \hat{u}_6\right) = g(4,6920; -5,5512; 0 \oplus 1)$$
$$= g(4,6920; -5,5512; 1) = 4,6920^{(1-2(1))}(-5,5512) = -1,1831.$$

  - *Cálculo de $L_4^{(8)}$:*

$$L_4^{(8)} \to g\left(L_2^{(6)}, L_2^{(8)}, \hat{u}_6\right) = g(-1,9178; -0,7338; 1)$$
$$= -1,9178^{(1-2(1))}(-0,7338) = 0,3826.$$

Os nós $L_4^{(4)}$ e $L_4^{(8)}$ fornecem-nos as informações necessárias para o cálculo de $L_8^{(4)}$:

  - *Cálculo de $L_8^{(4)}$:*

$$L_8^{(4)} \to f\left(L_4^{(4)}, L_4^{(8)}\right) = f(-1,1831; 0,3826) = \frac{(-1,1831 \cdot 0,3826) + 1}{-1,1831 + 0,3826} = -0,6838.$$

Como $L_8^{(4)} = -0,6838 < 1$, portanto $\hat{u}_7 = 1$.

- ***Etapa 8**: determinamos o nó remanescente de terceiro estágio.*

– *Cálculo de* $L_8^{(8)}$*:*

$$L_8^{(8)} \to g\left(L_4^{(4)}, L_4^{(8)}, \hat{u}_7\right) = g(-1,1831; 0,3826; 1)$$
$$= -1,1831^{(1-2(1))}(0,3826) = -0,3234.$$

Como $L_8^{(8)} = -0,3234 < 1$*, logo* $\hat{u}_8 = 1$.

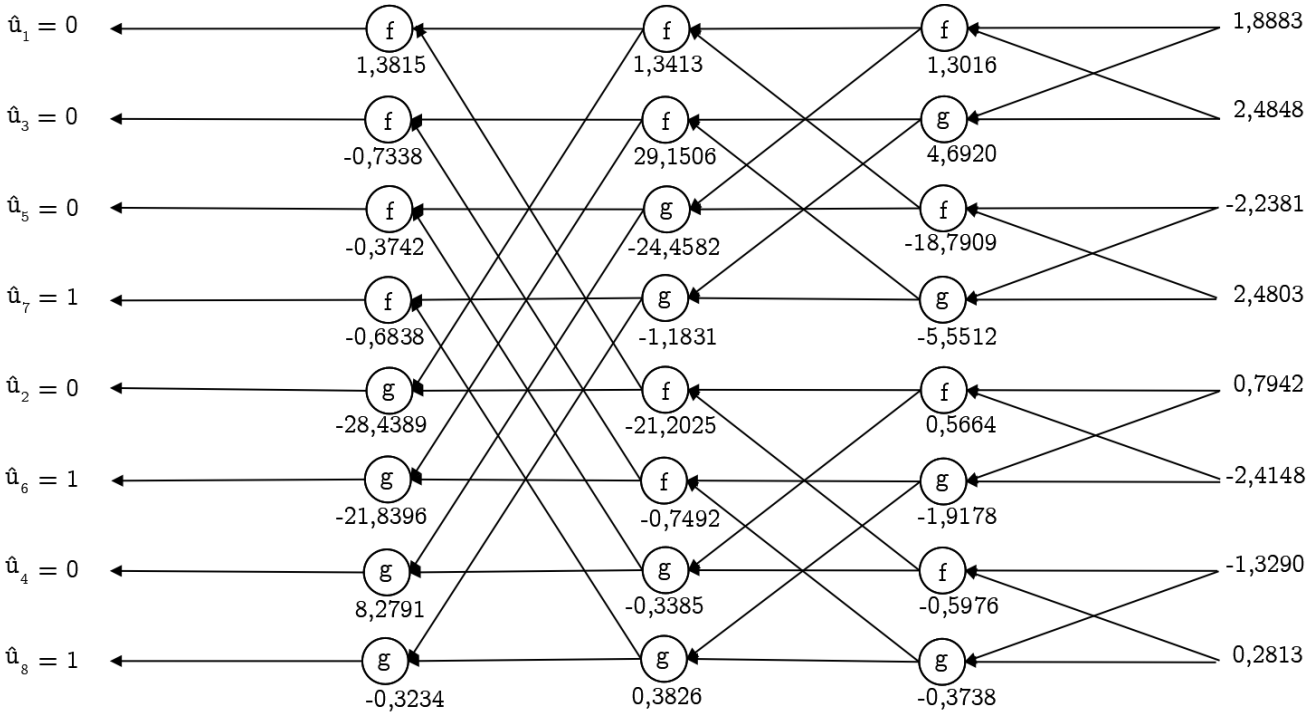
Após a determinação de cada um dos elementos de $\hat{\mathbf{u}} = (\hat{u}_1, \ldots, \hat{u}_8)$*, podemos escrever que o vetor decodificado é* $\hat{\mathbf{u}} = (00010101)$.

O Exemplo 3.3.1 foi retirado de (ROSA, 2022).

Então, sua rede final seguindo as equações de recurso é ilustrada na Figura 16.

Podemos notar que as atualizações dependem do nó anterior, gerando essa rede de atualizações que no final nos entrega os valores para a decisão, onde para valores maiores que 1, o bit decodificado será 0. Já para valores menores que 1, o bit decodificado será 1.

Figura 16 – Exemplo decodificação SC.



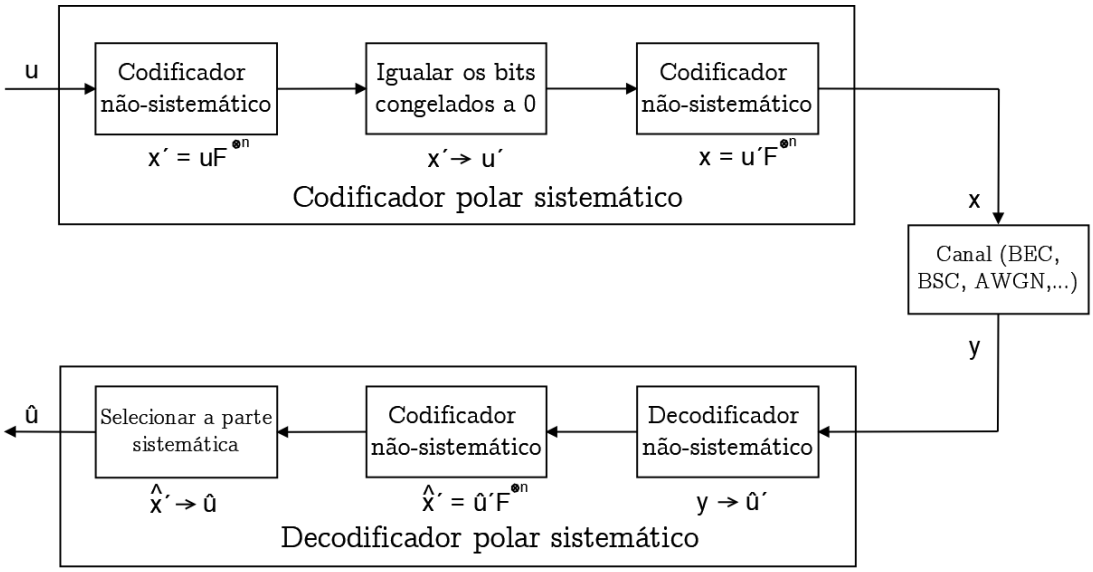
Fonte: retirado de (ROSA, 2022).

3.3.2 Decodificação SCL

Este método de decodificação utiliza de um decodificador composto por três sub-estágios. O primeiro é um decodificador polar SC, o segundo é um codificador polar não sistemático e o terceiro é um seletor, conforme observado na Figura 17.

A mensagem passa pelo codificador não sistemático e em sua saída temos a palavra-código, que por sua vez ira ser transmitida por um canal ruidoso, como um canal BEC, BSC ou AWGN. Na

Figura 17 – Esquema de decodificação SCL para códigos polares.



Fonte: elaborado pelo autor.

saída desse canal temos o vetor $\mathbf{y}$ que é recepcionado pelo decodificador SCL. O decodificador polar por cancelamento sucessivo em lista, estima $\hat{\mathbf{u}}$, que corresponde à mensagem original $\mathbf{u}$. Entretanto, no lugar de uma decodificação polar direta, o decodificador realiza em seu primeiro estágio uma decodificação SC.

O código $\tilde{\mathcal{C}}$ possui o mesmo conjunto de palavras-código que o código sistemático $\mathcal{C}$, porem utiliza um mapeamento diferente de mensagens para palavras-código. Dessa forma, a mensagem $\mathbf{y}$ é decodificada e torna-se $\hat{\mathbf{u}}^{'}$, através de um decodificador SCL. O decodificador SC, mantém somente um caminho L (ou lista) para a decodificação, já o SCL tem como característica principal dividir cada caminho de decodificação em dois caminhos diferentes. Em um dos caminhos temos $\hat{u}_1 = 0$, e no outro caminho temos $\hat{u}_1 = 1$ (se u_i for um bit não congelado). Em seu segundo estágio, o decodificador tem um codificador não sistemático, que codifica a mensagem estimada $\hat{\mathbf{u}}^{'}$ para obter uma palavra-código $\hat{\mathbf{x}}^{'}$. No último estágio, o decodificador SCL possui um seletor que estima a mensagem original $\hat{\mathbf{u}}$, fazendo $\hat{\mathbf{u}} = \hat{\mathbf{x}}_B$.

Exemplo 3.3.2 *Vamos considerar um exemplo de decodificação SCL de um código polar com os parâmetros $N = 4$, $K = 4$ e $L = 2$, vetor transmitido $x = [1001]$ e o vetor recebido $y = [1101]$.*

Neste exemplo, utilizaremos o algoritmo de decodificação em lista com $L = 2$ e o canal BSC com probabilidade de erro de $p = 0,1$. Para isso, começaremos com a probabilidade de cada bit transmitido ser 0 ou 1, dado o vetor recebido $\mathbf{y} = [1101]$.

O primeiro passo é a inicialização das probabilidade. Para o primeiro bit $y_1 = 1$, temos as seguintes probabilidades de transição:

$$P(y_1 = 1 \mid u_1 = 0) = p = 0,1$$

$$P(y_1 = 1 \mid u_1 = 1) = 1 - p = 0,9$$

Logo, para u_1, as duas hipóteses são:

Caminho 1: $u_1 = 0$ *com probabilidade* $0,1$

Caminho 2: $u_1 = 1$ *com probabilidade* $0,9$

Agora, para o segundo bit $y_2 = 1$*, e dado que temos duas hipóteses para* u_1*, vamos expandir as probabilidades para* u_2*:*

Para $u_1 = 0$*, temos:*

$$P(y_2 = 1 \mid u_2 = 0) = p = 0,1$$

$$P(y_2 = 1 \mid u_2 = 1) = 1 - p = 0,9$$

Para $u_1 = 1$*, temos:*

$$P(y_2 = 1 \mid u_2 = 0) = 1 - p = 0,9$$

$$P(y_2 = 1 \mid u_2 = 1) = p = 0,1$$

Agora, expandimos as possibilidades para u_2*, com base nas duas hipóteses para* u_1*:*

Para $u_1 = 0$*, temos as duas possibilidades:*

Caminho 1.1: $u_1 = 0, u_2 = 0$ *com probabilidade* $0,1 \times 0,1 = 0,01$

Caminho 1.2: $u_1 = 0, u_2 = 1$ *com probabilidade* $0,1 \times 0,9 = 0,09$

Para $u_1 = 1$*, temos:*

Caminho 2.1: $u_1 = 1, u_2 = 0$ *com probabilidade* $0,9 \times 0,9 = 0,81$

Caminho 2.2: $u_1 = 1, u_2 = 1$ *com probabilidade* $0,9 \times 0,1 = 0,09$

Agora, mantemos os dois caminhos mais prováveis:

Caminho 2.1: $u_1 = 1, u_2 = 0$ *com probabilidade* $0,81$

Caminho 1.2: $u_1 = 0, u_2 = 1$ *com probabilidade* $0,09$

Para o terceiro bit $y_3 = 0$*, calculamos as probabilidades para* u_3*:*

Para $u_1 = 1$ *e* $u_2 = 0$*, temos:*

$$P(y_3 = 0 \mid u_3 = 0) = 1 - p = 0,9$$

$$P(y_3 = 0 \mid u_3 = 1) = p = 0,1$$

Para $u_1 = 0$ *e* $u_2 = 1$*, temos:*

$$P(y_3 = 0 \mid u_3 = 0) = 1 - p = 0,9$$

$$P(y_3 = 0 \mid u_3 = 1) = p = 0,1$$

Agora, expandimos os caminhos para o terceiro bit:
Para $u_1 = 1, u_2 = 0$*, temos:*

Caminho 2.1.1: $u_1 = 1, u_2 = 0, u_3 = 0$ *com probabilidade* $0,81 \times 0,9 = 0,729$

Caminho 2.1.2: $u_1 = 1, u_2 = 0, u_3 = 1$ *com probabilidade* $0,81 \times 0,1 = 0,081$

Para $u_1 = 0, u_2 = 1$*, temos:*

Caminho 1.2.1: $u_1 = 0, u_2 = 1, u_3 = 0$ *com probabilidade* $0,09 \times 0,9 = 0,081$

Caminho 1.2.2: $u_1 = 0, u_2 = 1, u_3 = 1$ *com probabilidade* $0,09 \times 0,1 = 0,009$

Agora, mantemos os dois caminhos mais prováveis:

Caminho 2.1.1: $u_1 = 1, u_2 = 0, u_3 = 0$ *com probabilidade* $0,729$

Caminho 1.2.1: $u_1 = 0, u_2 = 1, u_3 = 0$ *com probabilidade* $0,081$

Por fim, para $y_4 = 1$*, temos as seguintes probabilidades para* u_4*:*
Para $u_1 = 1, u_2 = 0, u_3 = 0$*, temos:*

$$P(y_4 = 1 \mid u_4 = 0) = p = 0,1$$

$$P(y_4 = 1 \mid u_4 = 1) = 1 - p = 0,9$$

Expandindo para o quarto bit:
Para $u_1 = 1, u_2 = 0, u_3 = 0$*, temos:*

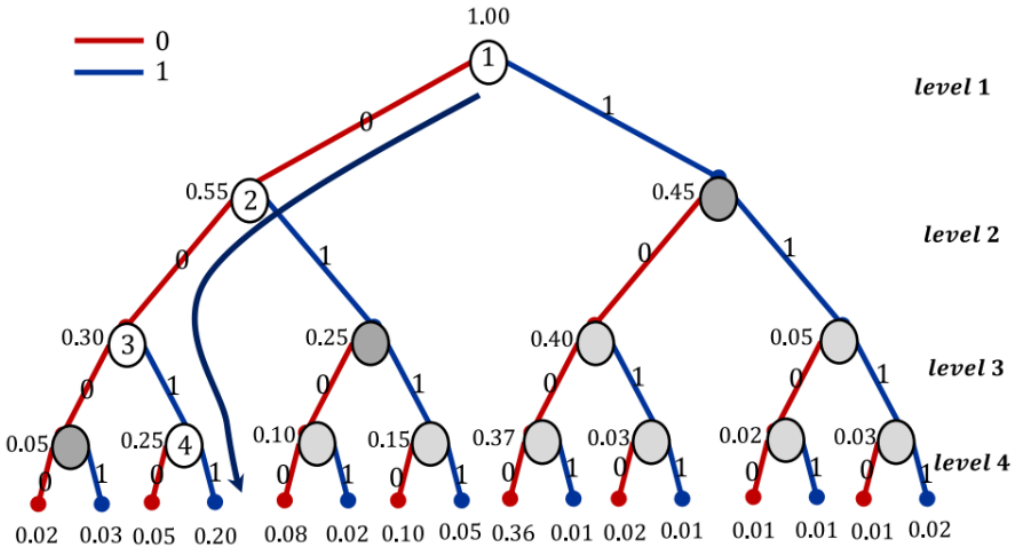
Caminho 2.1.1.1: $u_1 = 1, u_2 = 0, u_3 = 0, u_4 = 0$ *com probabilidade* $0,729 \times 0,1 = 0,0729$

Caminho 2.1.1.2: $u_1 = 1, u_2 = 0, u_3 = 0, u_4 = 1$ *com probabilidade* $0,729 \times 0,9 = 0,6561$

A sequência mais provável é $\mathbf{u} = [1, 0, 0, 1]$*, que corresponde ao vetor transmitido.*

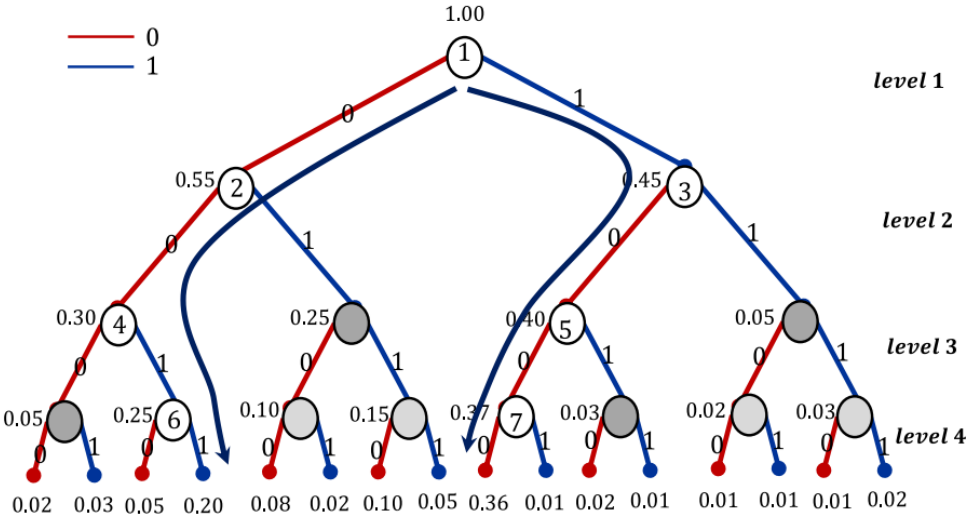
Conforme a quantidade de caminhos aumenta além da quantidade pré-definida de L, o decodificador SCL desconsidera os caminhos ou listas menos prováveis e mantém somente as listas mais prováveis (LI; SHEN; TSE, 2012). Essa diferença entre os decodificadores pode ser observada pela Figura 18 e Figura 19.

Figura 18 – Decodificador SC em estrutura de árvore, L = 1.



Fonte: Retirado de (ALAMDAR-YAZDI; KSCHISCHANG, 2011).

Figura 19 – Decodificador SCL em estrutura de árvore, L = 2.



Fonte: Retirado de (ALAMDAR-YAZDI; KSCHISCHANG, 2011).

4 MÉTODOS DE POLARIZAÇÃO

Este capítulo apresenta alguns métodos de polarização de canal que podem ser utilizados na construção de códigos polares, com foco na sua aplicação em tecnologias 5G NR e utilizando canais AWGN. Os métodos que iremos considerar são o parâmetro de Bhattacharyya e a Densidade de Evolução por Aproximação Gaussiana (DEGA), técnicas amplamente utilizadas para avaliar a confiabilidade dos canais e modelar a evolução das razões de verossimilhança (LLR). A principal contribuição deste trabalho reside na simplificação e otimização dessas técnicas, visando torná-las mais eficientes para sistemas práticos e com recursos computacionais limitados, demonstrando seu desempenho por meio de simulações.

4.1 BHATTACHARYYA

O método de polarização proposto por Arikan consiste em encontrar a capacidade do canal BEC, entretanto, quando partimos para transmissões reais, o canal sofre interferências sejam elas construtivas ou destrutivas. O canal que melhor se encaixa nesse cenário é o AWGN, porém o cálculo de sua capacidade não é trivial, logo, foi necessário o desenvolvimento de métodos para realizar a polarização desses canais.

O primeiro método apresentado neste trabalho é o de Bhattacharyya, que através da função de Bhattacharyya mede a confiabilidade do canal. Inicialmente, podemos considerar um canal qualquer W sem memória antes de partimos para o AWGN. Então o parâmetro $Z(W)$ pode ser definido por

$$Z(W) = \sum_{y} \sqrt{W\left(y|0\right) W\left(y|1\right)}, \tag{1}$$

sendo $W\left(y|x\right)$ as probabilidades de transição do canal. Onde a probabilidade de erro de bloco em códigos polares é verificada por

$$P_e(N, K, A, u_{A_c}) \leq \sum_{i \in A} Z(W_N^{(i)}). \tag{2}$$

A definição na Equação (1) leva em conta um canal discreto, porem, é razoável expandir essa definição para canais contínuos por conta da semelhança entre os dois tipos de canais. Para isso, utilizamos a densidade de distribuição de probabilidade no lugar da própria distribuição de probabilidade. Então, é realizada a integração no lugar da soma para canais contínuos. Dessa maneira podemos expressar $Z(w)$ como

$$Z(W) = \int \sqrt{W\left(y|0\right) W\left(y|1\right)} dy \tag{3}$$

Logo, as propriedades recursivas da polarização do canal podem ser calculadas pelas iterações do parâmetro Bhattacharyya através de

$$Z(W_N^{(i)}) \leq Z(W_{2N}^{(2i-1)}) \leq Z(W_N^{(i)}) - Z(W_N^{(i)})^2, i \in \{1, \ldots, N\} \tag{4}$$

e

$$Z(W_{2N}^{(2i)}) = Z(W_N^{(i)})^2, i \in \{1, \ldots, N\}. \tag{5}$$

A maneira mais simples de construir códigos polares para canais AWGN, é assumir que as atualizações do parâmetro de Bhattacharyya são próximas, ou seja, à medida que é feita a transformação polar, o parâmetro Bhattacharyya para cada posição de bit transformado permanece relativamente consistente com o parâmetro original do bit correspondente. Com isso, podemos simplificar para

$$Z(W_N^{(2i-1)}) = 2Z(W_N^{(i)}) - Z(W_N^{(i)})^2, \tag{6}$$

$$Z(W_{2N}^{(2i)}) = Z(W_N^{(i)})^2. \tag{7}$$

Nota-se que o parâmetro depende do estágio anterior do canal de bit i, e não do próximo canal de bit. A razão para isso é que quaisquer duas entradas em um nó são processadas pela mesma sequência de nós nos estágios anteriores.

Como o algoritmo depende do estágio anterior, é necessário encontrar o parâmetro de Bhattacharyya do canal AWGN do estágio subjacente, logo, para o primeiro parâmetro assumimos uma transmissão binária com potência unitária e uma variação de ruído de σ^2, assim temos que

$$Z_{AWGN} = exp\left(-\frac{1}{2\sigma^2}\right) = exp\left(-\frac{SNR}{2}\right). \tag{8}$$

Exemplo 4.1.1 *Para $N = 8$ canais AWGN com $SNR = 3dB$, vamos determinar pelo método de Bhattacharyya quais canais serão congelados e quais irão transmitir informação.*

Utilizando a equação (8) para encontrar o primeiro parâmetro, temos

$$Z_{AWGN} = exp\left(-\frac{10^{\frac{3}{10}}}{2}\right) = 0,3687.$$

Agora, utilizando as Equações (6) e (7), são encontrados os próximos dois parâmetros

$$Z(W_N^{(2i-1)}) = 2 \cdot 0,3687 - 0,3687^2 = 0,6014$$

e,

$$Z(W_{2N}^{(2i)}) = 0,3687^2 = 0,1359.$$

De maneira análoga, cada um deles passara pelo mesmo processo até que os parâmetros sejam gerados oito parâmetros finais. Com o objetivo de evitar a monotonia do trabalho, os cálculos posteriores foram omitidos, porém seguindo os passos anteriores facilmente pode-se chegar nos resultados a seguir:

$$Z(W_N^{(1)}) = 0,9747$$
$$Z(W_N^{(2)}) = 0,7074$$
$$Z(W_N^{(3)}) = 0,5918$$
$$Z(W_N^{(4)}) = 0,1307$$
$$Z(W_N^{(5)}) = 0,4424$$
$$Z(W_N^{(6)}) = 0,0641$$
$$Z(W_N^{(7)}) = 0,0365$$
$$Z(W_N^{(8)}) = 0,0003.$$

Após obter os resultados é possível escolher as posições congeladas, no caso $N/2$ posições serão congeladas. Para os maiores valores de Battacharyya atribuiremos um zero (congelado), e para os menores valores atribuímos o bit 1 *(informação). Sendo assim, o vetor final com as posições é*

$$P = [00010111].$$

Pelo exemplo observamos que o método não possuí uma complexidade alta. Ao decorrer do trabalho serão apresentados os resultados das simulações geradas com o método e consequentemente podemos classificar seu desemepenho.

4.2 DENSIDADE DE EVOLUÇÃO POR APROXIMAÇÃO GAUSSIANA

A medida de Bhattacharyya é uma ferramenta estatística que avalia o grau de sobreposição ou similaridade entre duas distribuições de probabilidade. Conforme o comprimento da palavra-código aumenta, o código se torna mais propenso a erros. Para lidar com isso, foi introduzida a Densidade de Evolução por Aproximação Gaussiana (DEGA), sendo este o método mais amplamente utilizado na construção de códigos polares em canais AWGN. O DEGA busca modelar a evolução das densidades das razões de verossimilhança (LLR) ao longo do processo de decodificação, partindo do canal até a decisão final. No entanto, esse processo não é simples, pois o comportamento das LLRs muda a cada estágio e nem sempre segue uma distribuição Gaussiana. Mesmo assim, a DEGA suaviza essa complexidade, assumindo que as LLRs são aproximadamente Gaussianas, permitindo que o decodificador rastreie apenas a média e a variância dessas LLRs.

A razão de verossimilhança de um canal AWGN binário é

$$LR_{AWGN} = \frac{2y}{\sigma^2}, \tag{9}$$

onde y é a saída do canal. Supondo que seja feita uma transmissão totalmente de zeros (sem informação), o valor médio da LLR é dado por

$$\mu = \frac{2}{\sigma}, \tag{10}$$

e a variância

$$V = \frac{4}{\sigma^2} = 2\mu \tag{11}$$

Está relação entre a média e a variância é assumida em todos os estágios, ou seja, em qualquer estágio temos uma LLR com distribuição $N(\frac{2}{\sigma^2}, \frac{4}{\sigma^2})$ ou simplesmente $N(\mu_i^{(s)}, 2\mu_i^{(s)})$. Desta forma, não é necessário rastrear a variância, mas somente a média.

Para fazer as atualizações, a DEGA utiliza das seguintes equações:

$$\mu = \begin{cases} \phi^{-1}(1-(1-\phi(\mu_i^{(s-1)}))^2), \textit{para o nó f}, \\ 2\mu_i^{(s-1)} \qquad\qquad\qquad\qquad\quad , \textit{para o nó g}, \end{cases} \tag{12}$$

onde ϕ é a função dada pela aproximação

$$\phi = \begin{cases} exp(-0,4527x^{0.86}+0.0218), & 0 < x < 10, \\ \sqrt{\frac{\pi}{x}} exp\left(-\frac{x}{4}\right)\left(1-\frac{10}{7x}\right), & x \geq 10, \end{cases} \tag{13}$$

e a função inversa de ϕ^{-1} no intervalo de $\mathbf{0 < x < 10}$ pode ser escrita como:

$$\phi^{-1} = 2.51316(-log_{10}(0.97843x))^{1.16279}. \tag{14}$$

Para obter a inversa ϕ^{-1} no intervalo de $\mathbf{x \geq 10}$ são necessários alguns passos. Considere a função $\phi(x)$ definida como

$$\sqrt{\frac{\pi}{x}} exp\left(-\frac{x}{4}\right)\left(1-\frac{10}{7x}\right),\ x \geq 10,$$

onde queremos encontrar a função inversa ϕ^{-1}. Em outras palavras, se $\mathbf{y} = \phi(\mathbf{x})$ então $x = \phi^{-1}(y)$ para todos os $\mathbf{x} \in \mathbf{D}(\phi)$ e $y \in Im(\phi)$.

Expandindo o argumento da função exponencial e reescrevendo a expressão para ϕ, temos

$$\phi(x) = x^{-\frac{1}{2}}\sqrt{\pi} exp\left(-\frac{x}{4}+\frac{5}{14}\right).$$

Seja $\mathbf{K} = \sqrt{\pi} exp^{\frac{5}{14}}$ uma constante, o que nos permite simplificar $\phi(x)$ como

$$\phi(x) = Kx^{-\frac{1}{2}}e^{-\frac{x}{4}} \iff \frac{\phi(x)}{K} = x^{-\frac{1}{2}}e^{-\frac{x}{4}}.$$

Elevando ambos os lados ao quadrado, temos

$$\frac{\phi^2(x)}{K^2} = x^{-1} exp\left(-\frac{x}{2}\right).$$

Portanto, para todo $\phi(x) \neq 0$,

$$\frac{\phi^2(x)}{K^2} = x exp\left(\frac{x}{2}\right). \tag{15}$$

Nota-se que o lado direito da Equação 15 se assemelha à função $\mathbf{f(x) = xe^x}$, cuja a função inversa e conhecida como a função W de Lambert (CORLESS et al., 1996). Isso implica que

$\mathbf{W(z) = x \iff z = xe^x}$. Observe que se dividirmos ambos os lados da Equação 15 por 2, obtemos

$$\frac{K^2}{2\phi^2(x)} = \frac{x}{2} exp\left(\frac{x}{2}\right)$$

Portanto, usando a definição da função W,

$$W\left(\frac{K^2}{2\phi^2(x)} = \frac{x}{2}\right)$$

e consequentemente

$$\phi^{-1} = 2W\left(\frac{\pi exp(5/7)}{2y^2})\right), \tag{16}$$

onde $\mathbf{x = \phi^{-1}}$ e $\mathbf{y} = \phi(\mathbf{x})$.

Pela complexidade apresentada, podemos aproximar numericamente a função inversa ϕ^{-1} através de funções *piecewise*. Essas funções são úteis para modelar situações onde o comportamento da função muda em diferentes regiões do domínio, como é o caso da função inversa ϕ^{-1}.

As *picewise function* são:

$$\phi^{-1} = \begin{cases} x(0.2202x + 0.06448), & x \leq 1, \\ x(0.062883x + 0.3678) - 0.1627, & 1 < x \leq 3.5, \\ x(0.009005x + 0.7694) - 0.9507, & 3.5 < x \leq 12, \\ 0.9861x - 2.3152, & x > 12. \end{cases} \tag{17}$$

Com essa estrategia conseguimos diminuir a complexidade do método, sendo muito útil para aplicações reais que não dispõem de muito recurso computacional.

A tabela a seguir mostra os resultados de ϕ^{-1} utilizando as funções *picewise* e a Equação 16. Podemos observar que os valores são extremamente próximos, portanto é possível validar e utilizar as *picewise* para otimizar o método. A variável $\mathbf{x}$ representa um valor intermediário relacionado com a

Tabela 1 – Comparação entre a função *piecewise* e a função W de Lambert.

x	Piecewise	W de Lambert
3.1697863849222268	1.6349672502648627	1.6356564188994864
1.6349672502648627	0.6067346281466468	0.6043971397971949
6.3395727698444535	4.288879886170222	4.278546095208556
0.6067346281466468	0.12018379418298919	0.13420723287722258
4.288879886170222	2.5148065729747184	2.5166299341113842
3.2699345005297253	1.7123566032917543	1.711129165023591
12.679145539688907	10.18770541668723	10.36905227851895

Fonte: elaborado pelo autor.

razão de verossimilhança logarítmica (LLR), que, neste caso, é usada como entrada para calcular a inversa de ϕ. O primeiro valor foi escolhido aleatoriamente, os demais valores são evoluídos pela 12.

Com isso podemos encontrar a função inversa ϕ^{-1} mais facilmente. A Figura 15 apresentada na Seção 3.3 também pode representar as etapas de polarização de canal por meio do método DEGA, onde a evolução acontece por meio dos nós f e g.

5 RESULTADOS E DISCUSSÕES

Os resultados encontrados nesse trabalho foram obtidos através de simulações computacionais utilizando a linguagem de programação *Python* para analisar o desempenho de códigos polares em um canal AWGN, além disso, cada simulação contou com 10^6 repetições para que o resultado obtido fosse confiável, ou seja, garantir que houvesse a convergência. O código foi reproduzido em um dispositivo com o processador AMD Ryzen 3 3200G com Radeon Vega Graphics 3.60 GHz e 16 GB de RAM. O código utilizado para analise está disponível em (BARROS, 2024).

Inicialmente, foi configurado um ambiente com bibliotecas que facilitam a manipulação de dados e simulações, o NumPy (HARRIS et al., 2020) e o SciPy (VIRTANEN et al., 2020). Essas bibliotecas permitem criar o ambiente matemático necessário para trabalhar com os métodos de polarização Bhattacharyya e DEGA, além de modelar o canal AWGN.

Além disso, foi configurado os principais parâmetros da simulação, incluindo:

- Tamanho do código polar (N) e número de bits de informação (k), que determinam o tamanho da sequência e a quantidade de bits de dados;
- Relação Sinal-Ruído, usada para ajustar o nível de ruído no canal;
- Número de mensagens simuladas, representando a quantidade de transmissões repetidas para obtenção de uma média de erros;
- O tipo de codificação (sistemática ou não sistemática).

Esses parâmetros também serão variados no contexto dos métodos Bhattacharyya e DEGA, pois influenciam a escolha de índices congelados diretamente, afetando o desempenho do código para as condições de SNR configuradas. A SNR em decibel é dada por:

$$SNR_{db} = 10log_{10}(\frac{P_s}{N_r}), \tag{1}$$

onde $\mathbf{P_s}$ é a potência do sinal, e $\mathbf{N_r}$ a potência do ruído.

Como explicado anteriormente, o parâmetro de Bhattacharyya calcula uma medida que indica a confiabilidade dos canais subjacentes no código polar. Canais com um valor de Bhattacharyya mais baixo são mais confiáveis e, portanto, reservados para os bits de informação, enquanto canais com valores altos são congelados.

Já a Densidade de Evolução por Aproximação Gaussiana determina a confiabilidade dos canais, criando uma distribuição de densidade, onde os maiores valores são destinados a canais de informação e os menores a canais congelados. Ambos os métodos geram uma lista de posições de bits, que será utilizada na etapa de codificação da mensagem.

Com os índices congelados definidos pelo método de Bhattacharyya ou DEGA, codificamos a mensagem. Na codificação polar, os bits de informação são posicionados nos índices mais confiáveis (conforme indicados pelos métodos de polarização), enquanto os bits congelados são posicionados nos

locais menos confiáveis, assumindo sempre o valor zero. Esta etapa garante que o código polar esteja otimizado para o cenário específico do canal.

Para a etapa de transmissão, a palavra-código é modulada pela modulação BPSK (*Binary Phase Shift Keying*), na qual cada bit é transformado em um sinal: $+1$ para bits 0 e -1 para bits 1.

Com a palavra-código definida e modulada, podemos realizar a transmissão pelo canal AWGN, onde um ruído gaussiano é introduzido simulando um ambiente de comunicação ruidoso. Esse ruído é ajustado pela SNR definida inicialmente. Durante a transmissão, o sinal codificado passa por esse canal e sofre alterações que representam a interferência do ruído.

Após a mensagem ser transmitida, chegamos na etapa de decodificação. O método de decodificação por cancelamento sucessivo, reconstrói a mensagem bit a bit. Durante a decodificação, são usadas as informações de polarização dos métodos Bhattacharyya e DEGA para maximizar a chance de recuperação precisa da sequência de informação, permitindo a reconstrução dos bits originais, mesmo em meio ao ruído. Para diminuir a complexidade do código e consequentemente o recurso computacional exigido, foi utilizado $L = 1$ para a decodificação SCL, pois equivale a decodificação SC.

Após a decodificação, avaliamos o desempenho do código polar calculando a razão de erro de bit. Esse cálculo compara os bits transmitidos com os bits decodificados para determinar a porcentagem de erros. O desempenho do código varia conforme o método de polarização utilizado, pois cada método define os índices congelados de forma diferente, afetando a robustez do código polar em um cenário de ruído. Como o objetivo é a transmissão de dados com a menor probabilidade de erro, o 3GPP impôs que a razão de erro de bit deve ser de pelo menos $\mathbf{10^{-6}}$, ou seja, para cada um milhão de bits enviados acontecerá um erro. A BER pode ser encontrada pela seguinte equação:

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

onde N_b é o número de bit transmitidos e n_e é o número de bits errados. Em um canal de comunicação ruidoso do tipo AWGN com uma modulação BPSK, a partir da SNR, é possível obter uma função para encontrar a razão de erro de bit. Para a decisão utiliza-se o valor de sinal de 0, ou seja, caso o valor seja menor que 0, o sistema entenderá que foi recebido o bit 1 e, ao contrário o sistema entenderá que o bit recebido foi 0. Acontecerá um erro se o bit enviado for $0 \rightarrow s = -1$ e o valor do ruído for $n < -1$. O contrário também é valido, se $1 \rightarrow s = -1$ e o valor do ruído seja $n > +1$, acontecerá um erro.

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

Sabemos que os símbolos $\mathbf{+1}$ **e** -1 são equiprováveis, ou seja, as probabilidades de cada um ser enviado é de $1/2$. Já, para o caso das probabilidades dos ruídos, é necessário retornar à função densidade de probabilidade de um ruído gaussiano para $\mu = \mathbf{0}$

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

logo, as duas probabilidades $\mathbf{Prob\{n < -1\}}$ e $\mathbf{Prob\{n < +1\}}$ podem ser representadas através da função cumulativa complementar $\mathbf{Q}(\frac{\mathbf{1}}{\sigma})$. Então, a equação da BER em função da SNR será

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

Para um sistema sem codificação, cada bit enviado é interpretado como informação, porem, para todos os casos onde as palavras são codificadas, aumentar o número de bits de informação reduz a taxa, mas consequentemente para enviar a mesma mensagem é necessário uma potência maior. Por esse motivo, a métrica recomendada para calcular a razão de erro de bit é a energia por bit de informação

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

Como a SNR pode ser defida como a relação entre a energia por bit de informção e a potência do ruído definida por $\frac{\mathbf{N_0}}{\mathbf{2}} = \sigma^2$,

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

Então, podemos reescrever a BER como

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

Primeiramente foi simulada a transmissão de códigos polar por um canal AWGN com modulação BPSK, utilizando o método de Bhattacharyya para polarizar o canal, e uma codificação não sistemática.

Analisando a Figura 20 vemos que, conforme o tamanho da palavra aumenta, menor é a razão de erro de bit, porem outro parâmetro igualmente importante é a SNR, pois ela mostra como o sistema se comporta com mais potência. Quando a SNR atinge $4dB$ e um tamanho de palavra $N = 256$, temos uma BER de 10^{-6}, como exigido pelo 3GPP. No entanto, não podemos afirmar ainda qual a real influência da codificação, mas já podemos perceber que um método de polarização eficiente é extremamente importante.

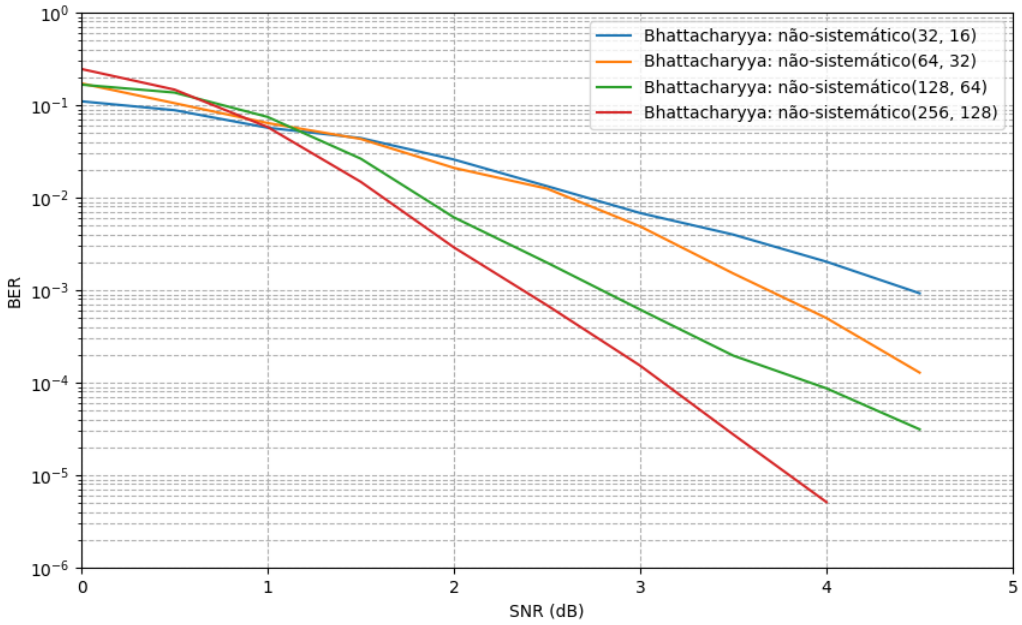
Agora a Figura 21 mostra o desempenho da transmissão quando o método de Bhattacharyya é implementado com uma codificação sistemática.

Ao analisar o resultado apresentado na figura, observa-se que o desempenho neste caso é semelhante ao do cenário anterior. A razão de erro de bit diminui à medida que o tamanho da palavra aumenta e a potência do sistema melhora. Para facilitar a comparação entre as duas simulações, a Figura 22 apresenta o desempenho do método de Bhattacharyya aplicado à codificação sistemática e não sistemática.

É evidente que a codificação sistemática apresenta melhor desempenho em todos os tamanhos de palavras e razões sinal-ruído. Esse comportamento se deve ao fato de que, na codificação sistemática, os bits de informação são preservados de forma mais eficaz, pois são explicitamente alocados na palavra-código. Essa característica, aliada à eficiente aplicação do método de Bhattacharyya, resulta em uma transmissão mais confiável e robusta, destacando a superioridade dessa abordagem.

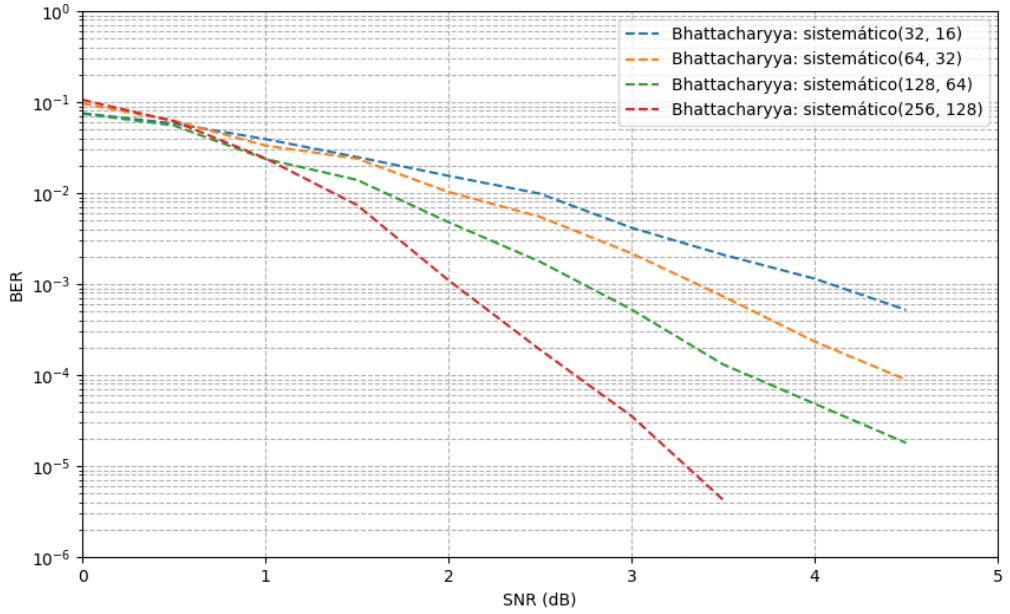
Agora, começaremos a analise do método DEGA a partir da Figura 23. Ao analisa-la, percebemos o mesmo comportamento do método anterior, porém é possível notar que para $N = 256$ e uma SNR de $4.5dB$, a razão de erro de bit alcança o valor mais baixo dentre as simulações até o momento.

Figura 20 – BER x SNR utilizando o método de Bhattacharyya com uma codificação não sistemática.



Fonte: Elaborado pelo autor.

Figura 21 – BER x SNR utilizando o método de Bhattacharyya com uma codificação sistemática.

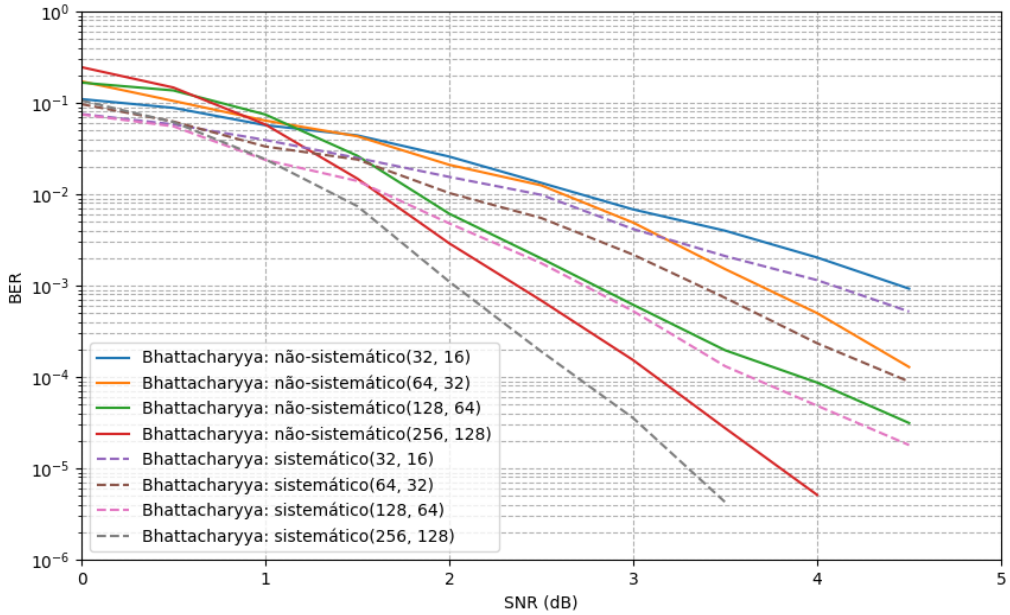


Fonte: Elaborado pelo autor.

Para observar a influência da codificação sistemática na implementação do método DEGA, podemos analisar a Figura 24. O desempenho é muito parecido novamente, por tanto, utilizaremos a Figura 25 para analisar o desempenho do método DEGA de forma assertiva, através da razão de erro de bit pela SNR do sistema. Ao analisa-la, percebe-se que o comportamento para com o código polar sistemática também é melhor nesse caso, reforçando mais uma vez o que foi dito anteriormente.

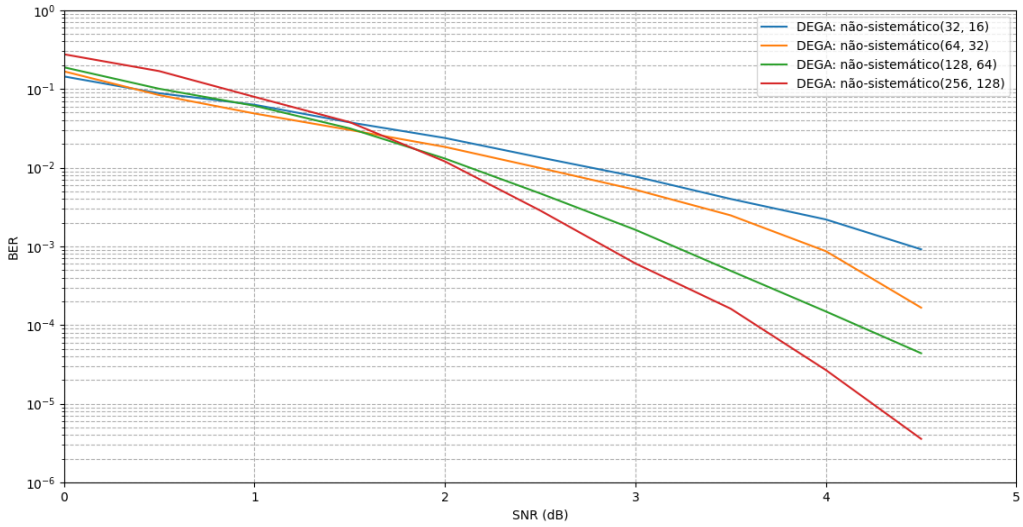
Para finalizar as discussões, vamos comparar lado a lado os dois métodos com codificação sistemática e não sistemática.

Figura 22 – Comparação entre a implementação do método de Bhattacharyya com uma codificação sistemática e não sistemática.



Fonte: Elaborado pelo autor.

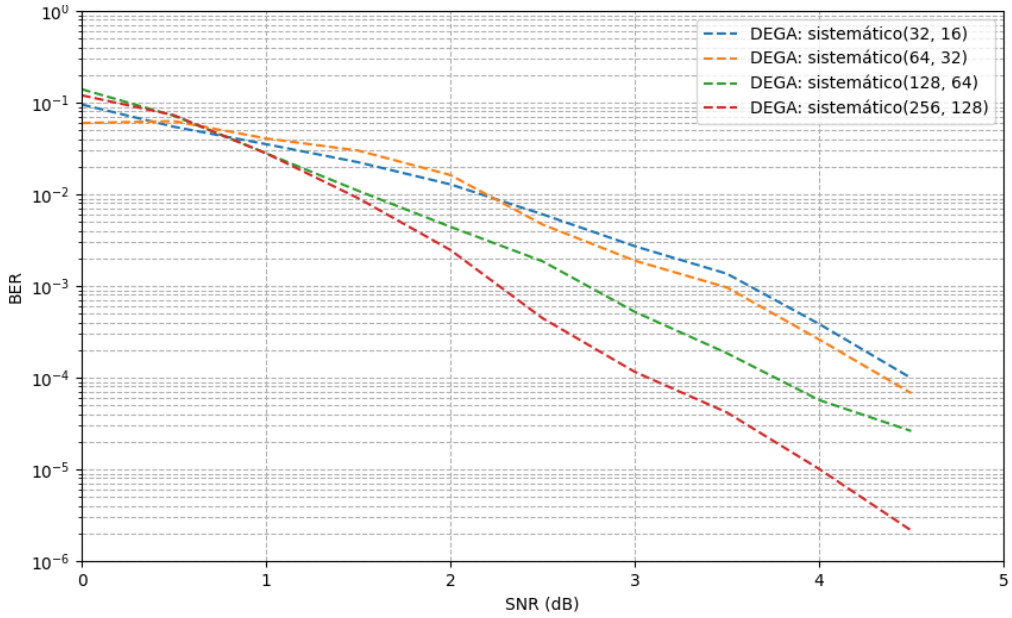
Figura 23 – BER x SNR utilizando o método DEGA com uma codificação não sistemática.



Fonte: Elaborado pelo autor.

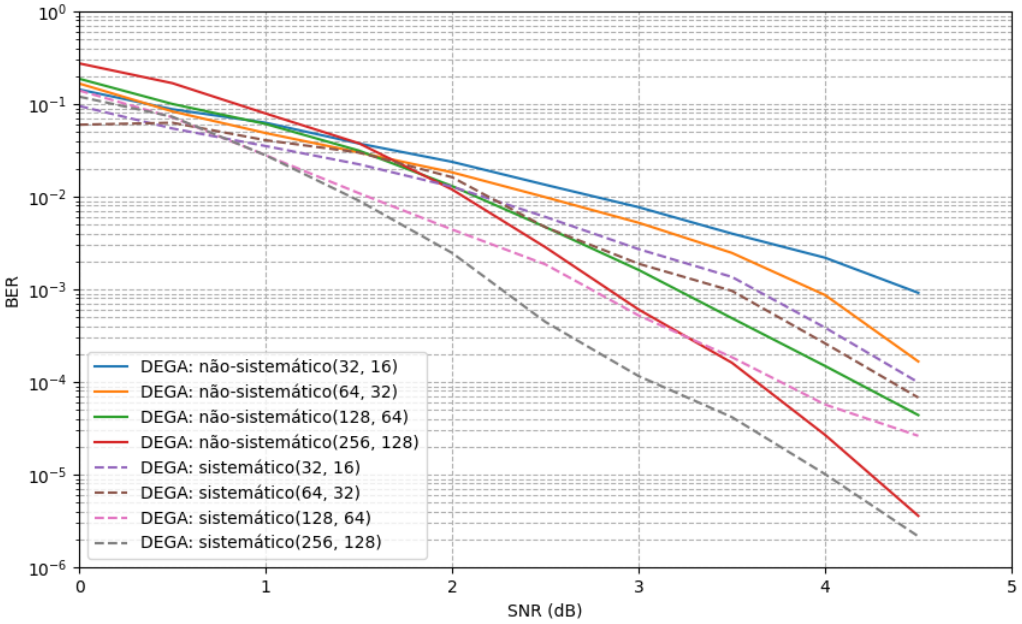
Pela Figura 26 nota-se que o método de Bhattacharyya desempenha melhor conforme o tamanho da palavra-código e a SNR aumenta. Isso acontece pois a complexidade do DEGA é maior, logo exige mais do sistema para atingir resultados satisfatórios, e quando atinge a razão sinal ruído de $4.5dB$ tem-se um valor de BER muito interessante. Para continuar compreendendo os métodos, vamos analisar o resultado da comparação entre os dois métodos com a codificação sistemática disponível na Figura 27. Novamente observamos que o método de Bhatthacharyya desempenhou melhor.

Figura 24 – BER x SNR utilizando o método DEGA com uma codificação sistemático.



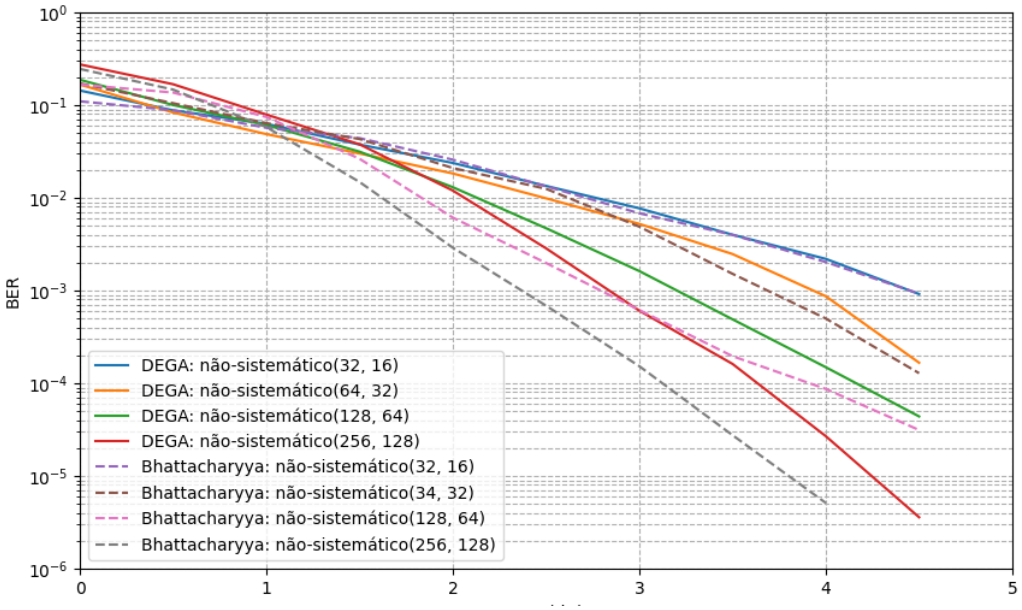
Fonte: elaborado pelo autor.

Figura 25 – Comparação entre a implementação do método de DEGA com uma codificação sistemática e não sistemática.



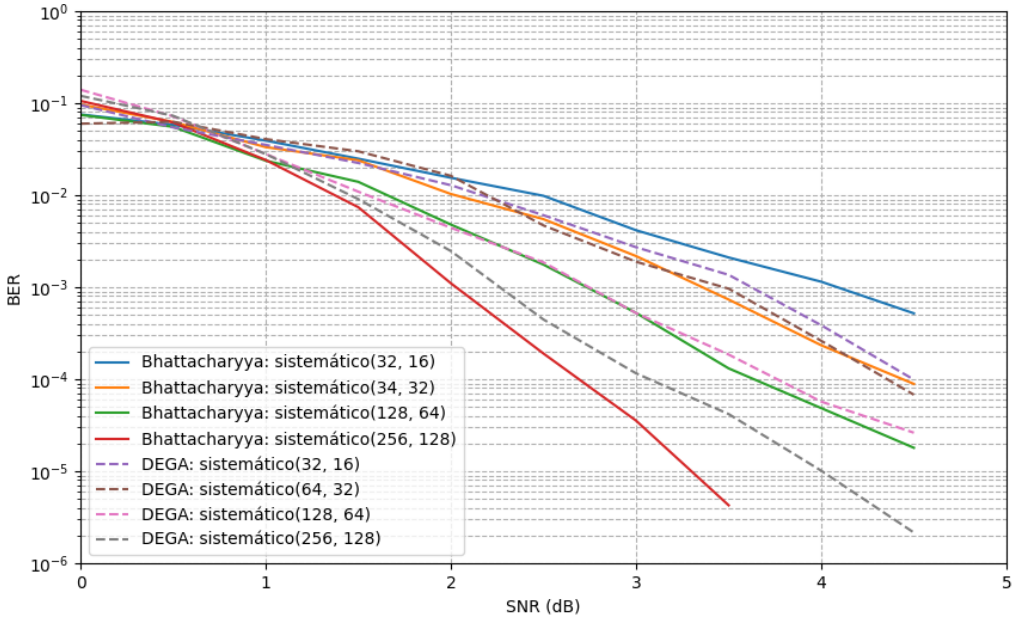
Fonte: Elaborado pelo autor.

Figura 26 – Comparação entre a implementação do método Bhattacharyya e DEGA com uma codificação não sistemática.



Fonte: elaborado pelo autor.

Figura 27 – Comparação entre a implementação do método Bhattacharyya e DEGA com uma codificação sistemática.



Fonte: Elaborado pelo autor.

6 CONCLUSÃO

Este trabalho teve como objetivo analisar e comparar métodos de polarização de canal aplicados à construção de códigos polares em cenários de comunicação 5G NR, mais especificamente em transmissões sujeitas a distorções que podem ser descritas pelo canal AWGN. A partir das simulações realizadas, foi possível aprofundar a compreensão sobre como o método de Bhattacharyya e DEGA influenciam o desempenho dos códigos polares em diferentes configurações e condições.

A polarização de canal é um dos pilares fundamentais para o sucesso dos códigos polares, uma vez que determina quais canais físicos serão utilizados para transmitir informações confiáveis. Nesse contexto, os métodos de polarização Bhattacharyya e DEGA possuem características distintas que os tornam mais adequados a diferentes cenários. O método de Bhattacharyya, pode ser amplamente utilizado em cenários onde a simplicidade e o baixo custo computacional prevaleçam, como por exemplo, um sistema embarcado. Essa conclusão foi tomada após analisar sua eficiência em uma SNR relativamente baixa, e se houver a necessidade de diminuir ainda mais o gasto computacional, implementar o método de Bhattacharyya juntamente com uma codificação sistemática pode ser a viável. Por outro lado, o método DEGA demonstrou vantagens significativas em aplicações que demandam alta precisão, como em sistemas onde a razão de erro de bit precisa ser extremamente baixa, como por exemplo, serviços de *streaming* onde as taxas de trasmissão de dados devem ser altas, com baixa latência e pouco ou nenhum erro, pois pode prejudicar seriamente o resultado final.

Outro ponto relevante é que, no contexto do 5G NR, a alta densidade de dispositivos conectados e a crescente demanda por altas taxas de transmissão tornam essencial o uso de métodos de polarização que garantam não apenas eficiência, mas também escalabilidade. A polarização de canal, aliada à codificação e decodificação eficientes dos códigos polares, desempenha um papel crucial na garantia de comunicação confiável e na mitigação de erros. Dessa forma, é evidente que os métodos Bhattacharyya e DEGA são complementares, podendo ser escolhidos conforme as necessidades específicas de cada aplicação, sejam elas relacionadas à otimização de recursos ou à maximização do desempenho.

Adicionalmente, o estudo reforça a relevância de explorar diferentes construções de códigos polares, sistemáticas e não sistemáticas, dependendo do ambiente de comunicação. Enquanto as construções sistemáticas oferecem benefícios relacionados à preservação da integridade dos dados transmitidos, as construções não sistemáticas, especialmente quando aliadas ao método DEGA, demonstraram uma redução significativa na taxa de erro em cenários desafiadores. Este aspecto é particularmente relevante para o 5G NR, onde a confiabilidade e a baixa latência são características indispensáveis.

Por fim, a pesquisa nos permitiu compreender como os métodos de polarização impactam diretamente a capacidade de transmissão e a confiabilidade do sistema. Em um cenário tecnológico em rápida evolução, como o 5G NR, a escolha adequada de técnicas de polarização é essencial para atender às demandas por maior eficiência, precisão e escalabilidade. Este trabalho, portanto, não apenas confirmou a eficiência dos métodos Bhattacharyya e DEGA, mas também destacou a importância de sua aplicação estratégica em diferentes cenários, fornecendo uma base sólida para futuras pesquisas e avanços na área de codificação de canais e comunicação sem fio.

Com base nos resultados apresentados, sugere-se que trabalhos futuros explorem outras técnicas de polarização e construções híbridas, que combinem as vantagens dos métodos estudados. Além disso, seria interessante avaliar o impacto dessas técnicas em canais mais complexos, como canais com desvanecimento, para ampliar a aplicabilidade dos resultados obtidos e contribuir para o avanço das tecnologias de comunicação.

REFERÊNCIAS

3GPP. *Study on New Radio (NR) Access Technologies*. 2018. <https://www.3gpp.org/DynaReport/38801.htm>. 3GPP TR 38.801 v15.0.0.

3GPP. *The 3rd Generation Partnership Project*. 2023. Available at: <https://www.3gpp.org> (accessed on November 20, 2024).

ALAMDAR-YAZDI, A.; KSCHISCHANG, F. R. A simplified successive-cancellation decoder for polar codes. **IEEE Communications Letters**, v. 15, n. 12, p. 1378–1380, December 2011.

ANDREWS, J. G. et al. What will 5G be? **IEEE Journal on Selected Areas in Communications**, v. 32, n. 6, p. 1065–1082, 2014.

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.

BARNABé, J. P. C. **Códigos polares aplicados em canais de múltiplo acesso**. Dissertação (Dissertação (mestrado)) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2020. 1 recurso online (95 p.). Disponível em: <https://hdl.handle.net/20.500.12733/1638634>.

BARROS, M. G. D. **Matbarry's GitHub Repositories**. 2024. <https://github.com/Matbarry?tab=repositories>. Available at: <https://github.com/Matbarry?tab=repositories>.

BEKO, M. et al. Massive MIMO techniques for 5G and beyond—opportunities and challenges. **Electronics**, v. 10, n. 14, p. 1667, 2021. Disponível em: <https://www.mdpi.com/2079-9292/10/14/1667>.

BHATTACHARYYA, A. On a measure of the distance between two probability distributions. **Bulletin of the Calcutta Mathematical Society**, v. 35, n. 1, p. 99–109, 1943.

CORLESS, R. M. et al. On the lambert w function. **Advances in Computational Mathematics**, v. 5, n. 1, p. 329–359, December 1996.

COVER, T. M.; THOMAS, J. A. **Elements of Information Theory**. 2nd. ed. Hoboken, NJ: Wiley-Interscience, 2006.

GALLAGER, R. G. **Information Theory and Reliable Communication**. [S.l.]: Wiley, 1968.

GLOBALDATA. 5G advances to fulfill industry need for high speed, low latency. **GlobalData**, 2022. Accessed on 2024-11-23. Disponível em: <https://www.globaldata.com/media/disruptor/5g-is-often-promoted-to-radically-improve-wireless-connectivity-compared-to-the-existing-4g/>.

HARRIS, C. R. et al. Array programming with numpy. **Nature**, Nature Publishing Group, v. 585, n. 7825, p. 357–362, 2020.

HAYKIN, S. **Communication Systems**. 4th. ed. New York: Wiley, 2001.

LI, B.; SHEN, H.; TSE, D. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check. **IEEE Communications Letters**, v. 16, p. 2044–2047, 2012. Disponível em: <https://api.semanticscholar.org/CorpusID:2244250>.

LIN, S.; COSTELLO, D. J. **Error Control Coding: Fundamentals and Applications**. Englewood Cliffs, NJ: Prentice Hall, 1983.

LIN, S.; COSTELLO, D. J. **Error Control Coding: Fundamentals and Applications**. 2nd. ed. Upper Saddle River, NJ: Prentice Hall, 2004.

MACKAY, D. J. C. **Information Theory, Inference, and Learning Algorithms**. [S.l.]: Copyright Cambridge University Press, 2003.

OSSEIRAN, A.; MONSERRAT, J. F.; MARSCH, P. **5G Mobile and Wireless Communications Technology**. Cambridge: Cambridge University Press, 2016.

PARKVALL, S. et al. 5G NR - the new 5G radio-access technology. **IEEE Communications Standards Magazine**, v. 1, n. 4, p. 24–30, 2017.

PROAKIS, J. G.; SALEHI, M. **Digital communications**. [S.l.]: McGraw-hill, 2008.

ROSA, V. S. A. **Análise de desempenho de códigos polares não-sistemáticos e sistemáticos em cenários para sistemas de quinta geração 5G**. São João da Boa Vista: Trabalho de conclusão de curso - Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Engenharia, São João da Boa Vista, 2022.

ROSLEE, M.; TIANG, J. J.; REHMAN, A. U. A survey on 5G coverage improvement techniques: Issues and future challenges. **Sensors**, MDPI, v. 23, n. 4, p. 2356, 2023.

ROTH, R. M. **Introduction to Coding Theory**. 2nd. ed. Cambridge, UK: Cambridge University Press, 2006.

RYAN, W. E.; LIN, S. **Channel Codes: Classical and Modern**. Cambridge, UK: Cambridge University Press, 2009.

SHAFI, M. et al. 5G: A tutorial overview of standards, trials, challenges, deployment, and practice. **IEEE Journal on Selected Areas in Communications**, v. 35, n. 6, p. 1201–1221, 2017.

SHANNON, C. E. A mathematical theory of communication. **The Bell System Technical Journal**, Bell Labs, v. 27, n. 3, p. 379–423, July 1948.

TAL, I.; VARDY, A. List decoding of polar codes. **IEEE Transactions on Information Theory**, v. 61, n. 5, p. 2213–2226, 2011.

TERçAS, L. **Codificação de canal para comunicações ultra confiáveis em sistemas 5G**. São João da Boa Vista: Trabalho de conclusão de curso - Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Engenharia, São João da Boa Vista, 2019.

TRIFONOV, P. Efficient design and decoding of polar codes. **IEEE Transactions on Communications**, v. 60, n. 11, p. 3221–3227, 2012.

VIRTANEN, P. et al. Scipy 1.0: fundamental algorithms for scientific computing in python. **Nature Methods**, Nature Publishing Group, v. 17, n. 3, p. 261–272, 2020.