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

VINÍCIUS STEPHANO ATAIDE ROSA

Análise de Desempenho de Códigos Polares Não-Sistemáticos e Sistemáticos em Cenários para Sistemas de Quinta Geração 5G

São João da Boa Vista
2022

VINÍCIUS STEPHANO ATAIDE ROSA

Análise de Desempenho de Códigos Polares Não-Sistemáticos e Sistemáticos em Cenários para Sistemas de Quinta Geração 5G

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

Orientadora: Profa. Dra. Cintya Wink de Oliveira Benedito
Coorientador: Prof. Dr. Ivan Aritz Aldaya Garde

São João da Boa Vista
2022

R788a

Rosa, Vinícius Stephano Ataide

Análise de desempenho de códigos polares não-sistemáticos e sistemáticos em cenários para sistemas de quinta geração 5G / Vinícius Stephano Ataide Rosa. -- São João da Boa Vista, 2022

79 p. : il.

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

Orientadora: Cintya Wink de Oliveira Benedito

Coorientador: Ivan Aritz Aldaya Garde

1. Codificação. 2. Códigos corretores de erros (Teoria da informação). 3. Sistemas de comunicação móvel. 4. Telecomunicações. 5. Teoria da informação. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Engenharia, São João da Boa Vista. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

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

ANÁLISE DE DESEMPENHO DE CÓDIGOS POLARES NÃO-SISTEMÁTICOS E SISTEMÁTICOS EM CENÁRIOS PARA SISTEMAS DE QUINTA GERAÇÃO 5G

Aluno: Vinícius Stephano Ataide Rosa
Orientador: Prof.ª Dr.ª Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientadora)
- Edgar Eduardo Benitez Olivo (Examinador)
- Leonardo Terças (Examinador)

A ata da defesa com as respectivas assinaturas dos membros encontra-se no prontuário do aluno (Expediente nº 064/2021)

São João da Boa Vista, 31 de agosto de 2022

A Deus, e a meus pais, Dercio Amauri Rosa e Cristiane Alves de Ataide Rosa, dedico.

AGRADECIMENTOS

A Deus, primeiramente, por ter me sustentado, guiado e conduzido até o presente momento.

A meus pais, Cristiane Alves de Ataide Rosa e Dercio Amauri Rosa, e a meus irmãos, Victor F. Ataide Rosa e Isadora L. A. Ataide Rosa, que nunca mediram esforços para me apoiar e animar durante cada momento de minha vida.

Aos demais familiares e à família Buettner, que apesar da distância, sempre demonstraram preocupação e afeição.

À minha orientadora, Profª. Drª. Cintya Wink de Oliveira Benedito, que proporcionou tanto ensinamento, conhecimento e inspiração durante as disciplinas que levaram ao interesse pelo tema.

Ao meu coorientador, Prof. Dr. Ivan Aritz Aldaya Garde, que de prontidão, se disponibilizou para auxiliar e orientar no desenvolvimento do trabalho.

Aos membros da banca examinadora, pela disponibilidade e atenção dedicadas a este trabalho, bem como por suas sugestões a respeito do mesmo.

Aos meus amigos da graduação, Amanda Belchior, Caio Andrade, Caio Grilo, Débora Beatriz Claro, Grabriel Roncoleta, Gabrielli de Oliviera, Leticia Dal'Cól, João Matheus Pasiani, Leonardo Chiconello, Lucas Gomes, Luis Henrique Chiang, Nícholas Martins e Pedro Penna, que partilharam tantos momentos durante esta árdua caminhada.

À Faculdade de Engenharia de São João da Boa Vista, e a todos que contribuíram para que este trabalho pudesse ser realizado.

“A essência do conhecimento consiste em aplicá-lo, uma vez possuído."
(Confúcio)

RESUMO

Os conceitos de teoria da informação, apresentados por Claude E. Shannon em 1948, continuam sendo até hoje, indispensáveis para a compreensão, desenvolvimento e aprimoramento dos esquemas de codificação de canal. Aos dados de interesse são adicionados dados de redundância, a fim de se garantir a integridade dos mesmos, além da busca pela maior capacidade de canal possível. Dentre as técnicas de codificação de canal candidatas à utilização nas aplicações dos sistemas de comunicação de quinta geração (5G) estão os códigos polares, apresentados em 2009 por Erdal Arikan em sua forma não-sistemática, com melhor desempenho em relação a taxa de erro de bit, BER (*Bit Error Rate*) e menor complexidade que os códigos turbo e LDPC (*Low Density Parity Check*). Em 2011, Arikan apresentou a forma sistemática dos códigos polares, com o intuito de se obter vantagens em termos de desempenho de BER, pois nesses é possível identificar os bits de informação na mensagem codificada. Dessa forma, neste trabalho será realizado o estudo das estratégias de codificação e decodificação, nas formas não-sistemática e sistemática, dos códigos polares. Além disso, será implementado um algoritmo computacional para comparar o desempenho de ambas em cenários para sistemas de quinta geração. Por fim, os resultados mostraram que o uso dos códigos de verificação de redundância cíclica, CRC (*Cyclic Redundancy Check*), melhora consideravelmente o desempenho dos códigos polares sistemáticos ao analisar-se a taxa de erro de frame, FER (*Frame Error Rate*).

PALAVRAS-CHAVE: Codificação de Canal. Códigos Polares. Sistemáticos. Redundância Cíclica. Taxa de Erro de Bit. Taxa de Erro de Frame.

ABSTRACT

The concepts of information theory, presented by Claude E. Shannon in 1948, continue to be, until today, indispensable for understanding, developing and improving channel coding schemes. Redundancy data is added to the data of interest in order to guarantee their integrity, in addition to the pursuit for the best possible channel capacity. Among the channel coding techniques which are candidates for being used on the applications of the fifth generation (5G) of communication systems are the polar codes, presented in 2009 by Erdal Arikan in their non-systematic form, with better performance in relation to BER (Bit Error Rate) and less complexity than the turbo and LDPC (Low Density Parity Check) codes. In 2011, Arikan presented the systematic form of polar codes, seeking advantages in terms of BER performance, as it is possible to identify the bits of information in the encoded message. In such manner, the study of the strategies of coding and decoding of polar codes, on its non-systematic and systematic form will be done in this work. Furthermore, a computational algorithm was implemented to compare the performance of both in scenarios for fifth generation systems. Finally, results showed that the use of CRC (Cyclic Redundancy Check) codes improved considerably the performance of systematic polar codes when analyzing the frame error rate, FER.

KEYWORDS: Polar Codes. Systematic. Coding. Cyclic Redundancy Check. Bit Error Rate. Frame Error Rate.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

1G	Primeira geração das redes de comunicações móveis
2G	Segunda geração das redes de comunicações móveis
4G	Quarta geração das redes de comunicações móveis
5G	Quinta geração das redes de comunicações móveis
6G	Sexta geração das comunicações móveis
AWGN	Additive White Gaussian Noise
B-DMC	Binary Discret Memoryless Channel
BEC	Binary Erasure Channel
BER	Bit Error Ratio
BSC	Binary Symmetric Channel
CRC	Cyclic Redundant Check
FER	Frame Error Ratio
LDPC	Low Density Parity Check
LR	Likelilyhood Ratio
mMTC	massive Machine-Type Communication
PDF	Probability Density Function
PMF	Probability Mass Function
SCL	Successive Cancellation List
SNR	Signal-Noise Ratio
uRLLC	ultra-Reliable Low-Latency Communication
UHD	Ultra-High Definition
xMBB	extreme Mobile Broadband

LISTA DE SÍMBOLOS

X	Variável aleatória discreta
X	Alfabeto
p	Probabilidade
x	Evento de interesse
$H(X)$	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
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
F_q	Corpo finito (de Galois) de tamanho q

$\mathcal{C}$	Código linear
R	Taxa de código
w_h	Peso de Hamming
w_{min}	Peso mínimo de Hamming ou peso mínimo
$\mathbb{G}$	Matriz geradora
$\mathbb{I}_k$	Matriz identidade de ordem k
$\mathbb{P}$	Matriz de ordem $(k \times (n-k))$
$\mathbb{H}$	Matriz controle de paridade
$\mathbf{c}$	Vetor transmitido
$\mathbf{r}$	Vetor recebido
$\mathbf{e}$	Vetor erro
$\mathbf{s}$	Vetor síndrome
$I(W)$	Capacidade simétrica
$Z(W)$	Parâmetro de Bhattacharyya
R_n	Bloco de permutação
F	Kernel de Arikan
$\otimes$	Produto tensorial de matrizes
$F^{\otimes n}$	Potência de Kronecker
B_n	Matriz permutação de inversão de bit
$\mathbf{v}$	Palavra-código
k	Dimensão do código
r	Apêndice
c	Código de verificação de redundância cíclica
g	Divisor
L	Caminhos das palavras-código possíveis do decodificador em lista CRC
$Z(W)$	Limite superior na probabilidade de erro de decisão de erro de máxima verossimilhança

$\mathbf{x}$	Vetor de entrada de um codificador
$\mathbf{y}$	Vetor de saída de um codificador
A	Linhas de G_N não congeladas
$\mathbf{u}_{A^c}$	Linhas congeladas de G_N
$\hat{\mathbf{u}}$	Mensagem estimada
n_t	Número de antenas transmissoras
n_r	Número de antenas receptoras
$\mathbf{Y}$	Matriz do sinal recebido
$\mathbf{X}$	Matriz do sinal transmitido
$\mathbf{H}$	Matriz do canal de desvanecimento Rayleigh quase-estático e de frequência plana
$\mathbf{Z}$	Matriz do ruído AWGN
T	Período de símbolo
C	Capacidade de canal
$\mathbf{I}_{n_r}$	Matriz identidade de ordem n_r
μ	Módulo do ganho de canal
$\tilde{\boldsymbol{x}}$	Sinal recebido após passar pelo combinador
$\hat{\boldsymbol{x}}$	Sinal recebido após passar pelo detector
$\min$	Menor elemento
$\det(\cdot)$	Determinante de uma matriz
$\log(\cdot)$	Logaritmo
$\|\| \cdot \|\|$	Norma de Frobenius
$(\cdot)^t$	Matriz transposta

SUMÁRIO

1 INTRODUÇÃO

A demanda por banda larga móvel e de qualidade será cada vez maior, principalmente devido ao aumento do consumo de vídeos em ultra-alta definição, como os de 4K UHD (*Ultra High Definition*) e os de 8K UHD, e também devido ao crescimento no número de conexões entre máquinas, seja nas indústrias, no campo, em residências ou nas cidades inteligentes. Desde o surgimento das redes de comunicação móveis de primeira geração (1G), a constante busca pelo aprimoramento dos meios de comunicação tem levado ao aperfeiçoamento das tecnologias e sua natural substituição por uma geração mais recente que atenda às necessidades de cada época. Hoje, as redes móveis de quinta geração (5G) devem atender três requisitos principais [5GPPP 2017]: banda larga massiva, xMBB (*extreme Mobile Broadband*), a qual deve ser capaz de entregar uma taxa de dados sob demanda na ordem de gigabits por segundo; conexão massiva de bilhões de sensores e máquinas, mMTC *(massive Machine-Type Communication)*; e comunicação ultraconfiável e de baixa latência, uRLLC (*ultra-Reliable Low-Latency Communication*).

Em abril de 2019, a Coreia do Sul foi o primeiro país a ter uma rede 5G comercial implementada, e espera-se que esse país tenha, até 2025, aproximadamente 60% de seus usuários móveis utilizando essas redes. No mesmo mês, as primeiras redes comerciais de quinta geração foram lançadas nos Estados Unidos, nas cidades de Chicago e Mineápolis. Já no continente europeu, a Suíça foi o primeiro país a ser contemplado. Em novembro do mesmo ano, foi a vez da China, onde hoje, as cidades de Pequim e Shenzhen já possuem cobertura total 5G [GSA 2022]. No Brasil, os primeiros testes das redes 5G foram realizadas em julho de 2020, entretanto, a primeira rede comercial do chamado 5G puro, em 3,5 GHz, foi lançada em 6 de julho de 2022, na capital federal, Brasília. Atualmente, já existem estudos e pesquisas em andamento para as redes móveis de 6ª. geração, 6G, as quais podem utilizar frequências na casa dos Terahertz e envolver aplicações como Internet de todas as coisas (IoE), e telepresença holográfica [Saad, Bennis e Chen 2020, Alwis et al. 2021].

Uma das maneiras de se garantir a integridade e confiabilidade nesses sistemas é através do uso de esquemas de codificação de canal, apresentado por Claude E. Shannon na década de 1940 [SHANNON 1948]. A fim de garantir que os dados recebidos através de um canal ruidoso sejam os mesmos que os dados enviados antes da transmissão, o transmissor adiciona bits de redundância a esses dados, em outras palavras, codifica a mensagem, e o receptor implementa algoritmos capazes de decifrar esses dados, ou seja, decodificar a mensagem recebida e recuperar a mensagem original. Quanto mais aprimorados forem os esquemas de codificação, melhor pode ser o desempenho dos sistemas de comunicação.

Com relação as gerações dos sistemas de comunicações móveis, os códigos turbo [Berrou, Glavieux e Thitimajshima 1993] têm sido utilizados desde a segunda geração (2G) até a quarta geração (4G) dos sistemas de comunicação sem fio [Shah, Vyavahare e Jain 2015]. Para os sistemas de quinta geração de comunicações sem fio, apesar de já estar em funcionamento, ainda há debates sobre qual a melhor estratégia de codificação de canal utilizar para novas aplicações, sendo que os códigos turbo, os códigos de verificação de paridade de baixa densidade, LDPC (*Low Density Parity Check*) e os

códigos polares são os possíveis candidatos [Patil, Pawar e Saquib 2020].

Os códigos polares [Arikan 2009], propostos por Erdal Arikan em 2009, são códigos do tipo bloco que têm apresentado um melhor desempenho em relação à taxa de erro de bit, BER (*Bit Error Rate*) e menor complexidade que os códigos turbo e LDPC, que têm desempenho comparável aos limites de Shannon, o que torna os códigos polares forte candidatos para serem utilizados na codificação e decodificação de canal nas aplicações dos sistemas de comunicações móveis 5G [Bioglio, Condo e Land 2021].

Apesar de terem sido inicialmente apresentados em sua forma não-sistemática, ou seja, a palavra-código não apresenta o bloco dos bits de informação em sua composição, Arikan apresentou, em 2011, a estratégia de codificação sistemática de códigos polares [Arikan 2011], tendo em vista a demonstração de que essa estratégia pode oferecer vantagens em termos de desempenho de BER.

Os códigos polares, não-sistemáticos ou sistemáticos, quando aplicados a blocos de comprimentos curtos ou médios nem sempre apresentam desempenho satisfatório, principalmente quando sua decodificação é realizada através do decodificador por cancelamento sucessivo, entretanto para blocos de comprimentos longos o desempenho é superior [Tal e Vardy 2011]. Assim, foi proposto o uso de um esquema de decodificação utilizando-se um decodificador por cancelamento sucessivo em lista, que considera caminhos em cada estágio de decodificação, de forma que uma única palavra seja selecionada no último estágio da decodificação. A utilização de códigos de verificação de redundância cíclica, CRC (*Cyclic Redundancy Check*), atrelados a esse decodificador aumenta ainda mais o desempenho dos códigos polares. Os CRCs são capazes de prever qual palavra-código na lista foi a transmitida, caso a mesma seja apresentada nessa lista, até alcançando desempenho superior que os códigos turbo e LDPC [Li, Shen e Tse 2012, Niu e Chen 2012, Baicheva e Kazakov 2019].

No presente trabalho analisaremos o desempenho de códigos polares sistemáticos em cenários para sistemas 5G, além da comparação do desempenho dos mesmos em relação aos códigos não-sistemáticos, cujo estudos já foram apresentados em [MARTÃO 2018, TERÇAS 2019]. As simulações computacionais serão realizadas em linguagem de programação *Python*, que é uma linguagem de código aberto, permitindo assim amplo uso, sem a necessidade de aquisição de licenças. Para essas simulações, palavras de comprimento k serão codificadas e transmitidas através de um canal com ruído aditivo Gaussiano branco, AWGN (*Additive White Gaussian Noise*) e decodificadas por um receptor, permitindo-se assim avaliar a BER, e a taxa de erro de frame, FER (*Frame Error Rate*).

A organização dos capítulos deste trabalho se dá da seguinte forma: o Capítulo 2 abordará uma revisão dos conceitos de teoria da informação e codificação de canal; o Capítulo 3 apresentará conceitos pertinentes aos códigos cíclicos e os códigos de verificação de redundância cíclica; o Capítulo 4 apresentará os códigos polares e estratégias de codificação e decodificação, não-sistemáticos e sistemáticos, para os mesmos; o Capítulo 5 introduzirá os parâmetros a serem analisados nas simulações das implementações computacionais e os resultados atingidos; a finalização deste trabalho se dará no Capítulo 6, onde apresentaremos as conclusões obtidas.

2 REVISÃO CONCEITUAL

O presente capítulo tem o objetivo de desenvolver uma revisão conceitual dos conceitos de Teoria da Informação e Codificação, que são primordiais para a compreensão e evolução deste trabalho. Apresentaremos inicialmente as definições e exemplos dos conceitos de entropia, informação mútua, capacidade de canal e alguns tipos de canais. Posteriormente, para o estudo da codificação de canal, apresentaremos as definições e exemplos de códigos de bloco lineares. Podemos encontrar as definições mais relevantes deste capítulo nas referências [Cover e Thomas 2006, Haykin 2001, Roth 2006, Ryan e Lin 2009, Lin e Costello 2004].

2.1 TEORIA DA INFORMAÇÃO

Em 1948, Claude E. Shannon apresentou os conceitos que hoje são conhecidos como Teoria da Informação [SHANNON 1948], sua teoria envolvia conceitos de matemática, engenharia e ciências, temas esses que raramente eram combinados em um único estudo. Nela Shannon apresentou seus estudos sobre armazenamento, quantificação e comunicação da informação, com o propósito de estabelecer os limites das operações das comunicações e do processamento digital de sinais.

Seja X uma variável aleatória discreta com alfabeto X e função massa de probabilidade, PMF (*Probability Mass Function*)

$$p(x) = Pr\{X = x\}, x \in \mathrm{X}. \tag{2.1}$$

A entropia é a medida de incerteza de uma variável aleatória.

Definição 2.1.1 *A entropia $H(X)$ de uma variável aleatória discreta X é definida por:*

$$H(X) = -\sum_{x \in \mathrm{X}} p(x) \log_2 p(x). \qquad \text{[bits]} \tag{2.2}$$

Por convenção, definiremos $0 \log 0 = 0$, através da continuidade, desde que $x \log(x) \to 0$, à medida que $x \to 0$. Observemos que a adição de termos de probabilidade nula não altera o valor da entropia. Percebamos também que a entropia é uma função da distribuição de X, e que não é dependente dos valores que a variável aleatória X pode tomar, mas sim de suas probabilidades.

A esperança de uma variável aleatória $g(X)$ é

$$E_p[g(X)] = \sum_{x \in X} g(x)p(x), \tag{2.3}$$

quando a PMF é entendida pelo contexto, então $E_p[g(X)] = E[g(X)]$. Temos um interesse peculiar na esperança autorreferencial de $g(X)$ sob $p(x)$ quando $g(X) = \log\left[\frac{1}{p(x)}\right]$. A entropia de X também pode ser interpretada como o valor esperado de uma variável aleatória $\log\left[\frac{1}{p(x)}\right]$, conforme X se define de acordo com a PMF $p(x)$,

$$H(X) = E_p\left[\log\left[\frac{1}{p(x)}\right]\right]. \tag{2.4}$$

Lema 2.1.1 $H(X) \geq 0$, *pois* $0 \leq p(x) \leq 1$, *implicando que* $\log\left[\frac{1}{p(x)}\right] \geq 0$.

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

$$X = \begin{cases} \frac{7}{10}, \ se\ x = a; \\ \frac{3}{10}, \ se\ x = b. \end{cases} \tag{2.5}$$

Através de (2.3), podemos calcular a entropia

$$H(X) = -\left[\frac{7}{10}\log_2\left(\frac{7}{10}\right) + \frac{3}{10}\log_2\left(\frac{3}{10}\right)\right] \simeq 0,88.$$

Definição 2.1.2 *A entropia conjunta* $H(X,Y)$ *de um par de variáveis aleatórias discretas* (X,Y) *com uma distribuição conjunta* $p(x,y)$ *é definida como*

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x,y), \tag{2.6}$$

e pode ser expressa como

$$H(X,Y) = -E[\log p(X,Y)]. \tag{2.7}$$

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

$$H(Y|X) = E[-\log p(y|x)] = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x|y). \tag{2.8}$$

A medida da quantidade de informação que uma variável aleatória contém sobre outra variável aleatória é definida como informação mútua, em outras palavras, é a redução na incerteza de uma variável aleatória devido ao conhecimento de outra. Dessa forma, poderemos utilizar essa grandeza para quantificar a eficiência de uma transmissão através de um canal, sendo que quanto mais dependente uma variável for da outra, maior será a informação mútua e maior será a confiabilidade da transmissão.

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

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2\left[\frac{p(x,y)}{p(x)p(y)}\right], \tag{2.9}$$

também podemos escrever

$$I(X;Y) = H(X) - H(X|Y), \tag{2.10}$$

$$I(X;Y) = H(Y) - H(Y|X), \tag{2.11}$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{2.12}$$

Os conceitos de entropia podem ser estendidos às variáveis aleatórias contínuas. A entropia diferencial, que é a entropia associada a uma variável aleatória contínua, está relacionada ao menor comprimento de descrição e é similar, de muitas maneiras, à entropia de uma variável aleatória discreta.

Seja X uma variável aleatória contínua com função densidade de probabilidade, PDF (*Probability Density Function*)

$$F(X) = P[a \leq X \leq b] = \int_b^a f(x)dx. \tag{2.13}$$

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

$$h(X) = -\int_S f(x)\log f(x)dx, \tag{2.14}$$

em que S é o grupo de suporte de uma variável aleatória.

Semelhantemente ao caso discreto, a entropia diferencial só tem dependência com a densidade de probabilidade da variável aleatória. Do caso discreto, pode-se estender a definição de entropia diferencial de uma única variável aleatória a um par de variáveis aleatórias.

Definição 2.1.6 *A entropia diferencial de um par de variáveis aleatórias contínuas (X,Y) função densidade de probabilidade conjunta $f(x,y)$ é definida como*

$$h(X,Y) = -\int f(x,y)\log f(x,y)dxdy. \tag{2.15}$$

Definição 2.1.7 *Se (X,Y) tem função densidade de probabilidade conjunta $f(x,y)$, podemos definir a entropia diferencial condicional $h(x|y)$ como*

$$h(X|Y) = -\int f(x,y)\log f(x|y)dxdy. \tag{2.16}$$

Como $f(x|y) = \frac{f(x,y)}{f(y)}$, podemos escrever:

$$h(X|Y) = h(X,Y) - h(Y). \tag{2.17}$$

Definição 2.1.8 *A informação mútua $I(X;Y)$ entre duas variáveis aleatórias com densidade conjunta $f(x,y)$ é definida:*

$$I(X;Y) = \int f(x;y)\log\left[\frac{f(x,y)}{f(x)f(y)}\right]dxdy. \tag{2.18}$$

Analogamente ao caso discreto, são válidas as seguintes relações:

$$I(X;Y) = h(X) - h(X|Y), \tag{2.19}$$

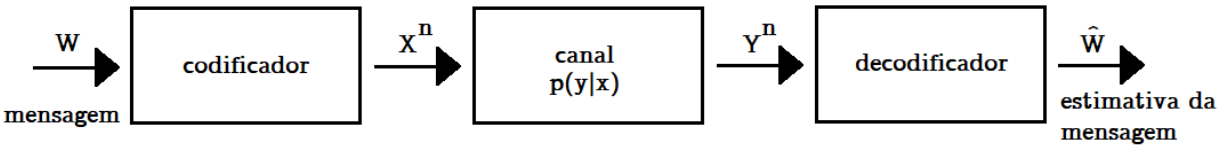
$$I(X;Y) = h(Y) - h(Y|X), \tag{2.20}$$

$$I(X;Y) = h(X) + h(Y) - h(X,Y). \tag{2.21}$$

A caracterização da capacidade de canal (o logaritmo do número de sinais distinguíveis) como a máxima informação mútua é o centro e um dos conceitos mais importantes da Teoria da Informação.

Definição 2.1.9 *Definimos um canal discreto como sendo um sistema consistindo de entrada* X*, um alfabeto de saída* Y *e uma matriz de transição de probabilidade* $p(y|x)$ *que expressa a probabilidade de receber o símbolo* y *dado que enviamos o símbolo* x*.*

Figura 1 – Sistema de comunicação.



Fonte: próprio autor.

O canal é dito ser sem memória se a distribuição probabilidade de saída depender somente da entrada e for condicionalmente independente das entradas ou saídas do canal anterior.

Definição 2.1.10 *Definimos a capacidade de informação de um canal discreto sem memória como:*

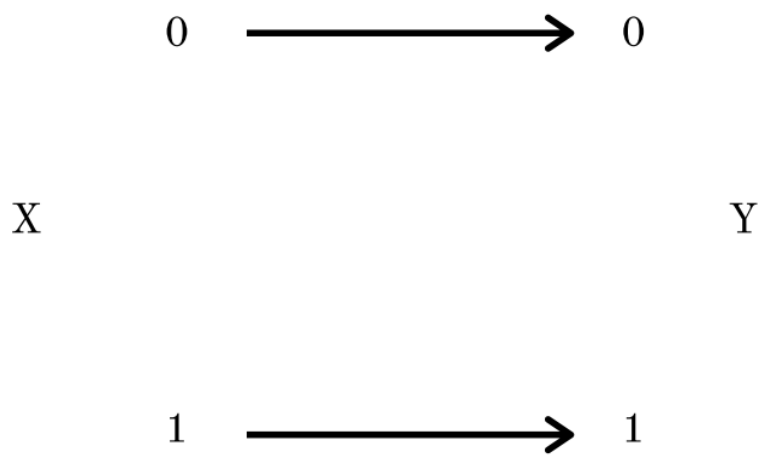
$$C = \max_{p(x)} I(X;Y), \tag{2.22}$$

em que o máximo é tomado sobre todas as possíveis distribuições da entrda $p(x)$*.*

O 2° Teorema de Shannon define que a capacidade de informação do canal é igual à capacidade operacional do canal. Veremos agora alguns tipos de canais.

Exemplo 2.1.2 *O canal binário sem ruído é aquele no qual qualquer bit é recebido sem erro, conforme apresenta a Figura 2. Portanto, um bit sem erro poder ser transmitido por uso do canal, e a capacidade é de 1 bit, que pode ser calculada através da capacidade de informação* $C = \max I(X;Y) = 1\ bit$*, obtida utilizando-se* $p(x) = \left(\frac{1}{2}; \frac{1}{2}\right)$ *.*

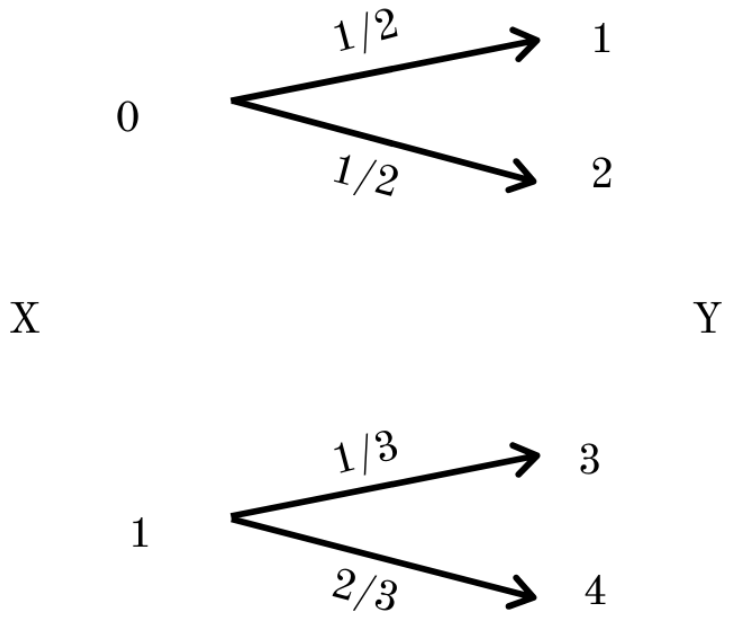
Figura 2 – Canal binário sem ruído.



Fonte: próprio autor.

Exemplo 2.1.3 *O canal ruidoso com saídas não-sobrepostas, apresentado na Figura 3, têm 2 saídas possíveis para cada uma das 2 entradas. Apesar de parecer ser ruidoso, não é, pois a saída é uma consequência aleatória da entrada, mas a entrada pode ser determinada através da saída, assim cada bit transmitido pode ser recuperado sem erro. A capacidade do canal também é 1 bit por transmissão.*

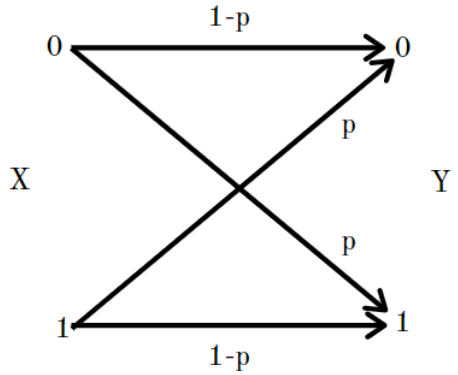
Figura 3 – Canal ruidoso com saídas não-sobrepostas.



Fonte: próprio autor.

Exemplo 2.1.4 *O canal binário simétrico, BSC (Binary Symmetric Channel), apresentado na Figura 4, tem os símbolos de entrada complementados com probabilidade p. Ele é o modelo mais simples de um canal com erros, mas ainda captura a maior parte da complexidade de problemas gerais. Quando um erro ocorre, 0 é recebido como 1 e vice-versa. Os bits recebidos não revelam onde os erros ocorreram.*

Figura 4 – BSC.



Fonte: próprio autor.

Podemos calcular a capacidade desse canal através da informação mútua e entropia.

$$I(X;Y) = H(Y) - H(Y|X), \tag{2.23}$$

$$I(X;Y) = H(Y) - \sum p(x)H(Y|X = x), \tag{2.24}$$

$$I(X;Y) = H(Y) - \sum p(x)H(p), \tag{2.25}$$

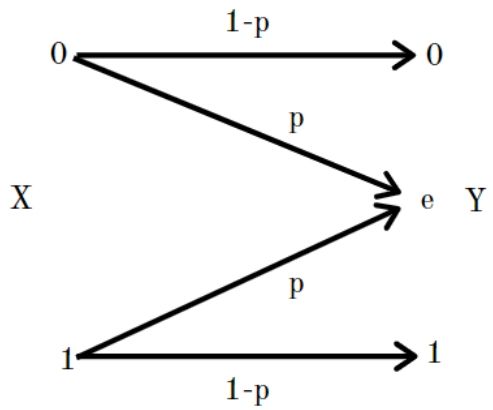
$$I(X;Y) = H(Y) - H(p) \leq 1 - H(p). \tag{2.26}$$

A inequação em (2.26) segue, pois Y é uma variável aleatória binária. A igualdade é atingida quando a distribuição de entrada é uniforme, como mostra a equação (2.27).

$$C = 1 - H(p). \tag{2.27}$$

Exemplo 2.1.5 *A analogia do BSC no qual alguns bits são perdidos (em vez de corrompidos) é o canal binário com apagamento, BEC (Binary Erasure Channel), apresentado na Figura 5. Neste canal uma fração p dos bits é apagada. O receptor sabe quais bits foram apagados. Ele tem duas entradas e três saídas.*

Figura 5 – BEC.



Fonte: próprio autor.

Podemos calcular a capacidade desse canal como

$$C = \max_{p(x)} I(X;Y) = \max_{p(x)}((H(Y) - H(Y|X)) = \max_{p(x)} H(Y) - H(p), \tag{2.28}$$

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

$$C = \max_{\pi} H(Y) - H(p) = \max_{\pi}(1-p)H(\pi) + H(p) - H(p) = \max_{\pi}(1-p)H(\pi). \tag{2.29}$$

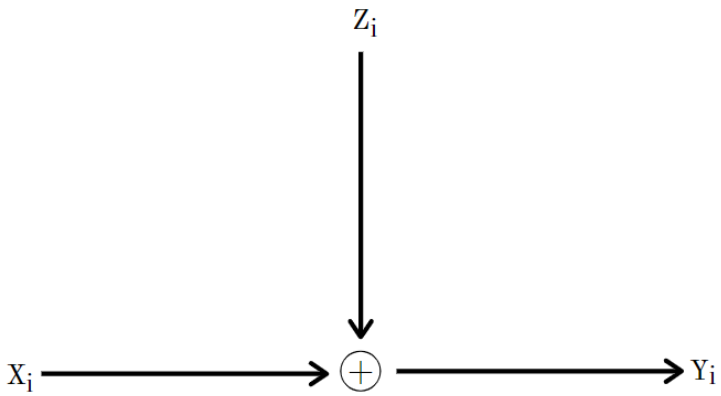
Logo,

$$C = 1 - p. \tag{2.30}$$

Exemplo 2.1.6 *O canal gaussiano, apresentado na Figura 6, é o alfabeto contínuo de canal mais importante; é um canal de tempo discreto com saída* Y_i *no tempo* (i)*, em que* Y_i *é a soma da entrada* X_i *e o ruído* Z_i*, esse ruído vem de uma distribuição gaussiana com variância* σ^2*, de acordo com (2.31),*

$$Y_i = X_i + Z_i,\ Z_i \sim \mathcal{N}, \tag{2.31}$$

Figura 6 – Canal gaussiano.



Fonte: próprio autor.

considera-se que o ruído Z_i *é independente do sinal* X_i*. Esse canal é um modelo para alguns canais de comunicação comuns, como de telefone com e sem fio e links satelitais. Se não houver considerações, a capacidade deste canal pode ser infinita. Podemos calcular a capacidade de informação desse canal, considerando uma restrição de potência:*

$$C = \max_{EX^2 \leq P} I(X;Y) = \frac{1}{2}\log\left(1 + \frac{P}{N}\right), \quad \text{[bits/transmissão]} \tag{2.32}$$

o máximo é obtido quando $X \sim \mathcal{N}(0, P)$*. Se houver limitação de banda, a capacidade do canal é dada por (2.33), na qual* B *é a banda* $[Hz]$*,* P *é a potência do sinal,* $\mathcal{N}$ *é o ruído, e* $\frac{P}{\mathcal{N}}$ *é a razão sinal-ruído, SNR (Signal-Noise Ratio),* $[dB]$*.*

$$C = B\log\left(1 + \frac{P}{\mathcal{N}}\right). \quad \text{[bits/s]} \tag{2.33}$$

2.2 CODIFICAÇÃO DE CANAL

A codificação de canal é uma estratégia implementada em sistemas de comunicação digitais, através da inserção de bits adicionais, conhecidos como bits de paridade, na mensagem a ser transmitida, o que permite que erros causados por exemplo, devido ao ruído, interferência ou *fading*, sejam detectados e corrigidos, através do uso de determinadas técnicas. Apresentaremos a seguir os códigos de bloco lineares, um dos mais importantes tipos de códigos corretores de erros.

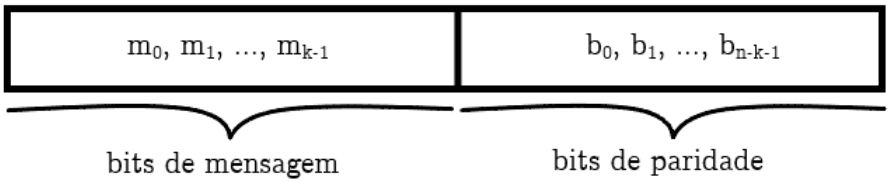
2.2.1 Códigos de Bloco Lineares

Dizemos que um código binário é linear se quaisquer duas palavras-código desse código podem ser adicionadas em uma soma módulo-2 para produzir uma terceira palavra-código no código.

Consideremos um código de bloco linear com parâmetros (n, k), no qual k bits de uma mensagem a ser transmitida são transformados em uma palavra-código de n bits adicionando $(n - k)$ bits de paridade ou bits de redundância, os quais são determinados de acordo com a técnica a ser utilizada. Os códigos de bloco nos quais os bits de mensagens são transmitidos de maneira inalterada são chamados de códigos sistemáticos.

A Figura 7, apresenta o um código de bloco linear sistemático. Temos que $(m_0, m_1, ..., m_{k-1})$ constitui um bloco de mensagem de k bits e o bloco $(b_0, b_1, ..., b_{n-k-1})$ denota os $(n - k)$ bits de paridade na palavra-código.

Figura 7 – Código de bloco linear sistemático.



Fonte: próprio autor.

Definição 2.2.1 *Seja $F_q = GF(q)$ um corpo finito (de Galois) com q elementos. Um código de bloco linear $\mathcal{C}$ com parâmetros (n, k) sobre F_q é um subespaço vetorial de F_q^n com k elementos. O código $\mathcal{C}$ possui 2^k palavras-código, que são as combinações lineares distintas e que geram o conjunto completo de $\mathcal{C}$. Dessa forma, $|\mathcal{C}| = M = q^k$, e $R = (\log_q M)/n = k/n$ é a taxa de código.*

Exemplo 2.2.1 *Avaliemos um código $\mathcal{C}$ com parâmetros $(3, 2)$, determinando a taxa de codificação, a quantidade de palavras-código e quais são elas. Podemos relacionar os parâmetros (n, k) a $(3, 2)$, logo $n = 3$ e $k = 2$. A taxa de codificação é dada por $R = \frac{k}{n} = \frac{3}{2}$. O código possui $q^k = 2^3 = 8$ palavras-código, que são:*

$$\mathcal{C} = \{000, 001, 010, 011, 100, 101, 110, 111\}.$$

Definição 2.2.2 *Seja $u = (u_0, u_1, ..., u_n)$ um vetor em F_q^n. O peso de Hamming, denotado por $w_h(u)$, 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 ou peso mínimo, e matematicamente é descrito por

$$w(\mathcal{C}) = min\{w(u) : v \in \mathcal{C}, v \neq 0\}. \tag{2.34}$$

Definição 2.2.3 *A distância mínima 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,*

$$d_{min}(\mathcal{C}) = min\{d(u, v) : u, v \in \mathcal{C}, u \neq v\}. \tag{2.35}$$

Utilizando-se do fato de que $d(u, v) = w(u + v)$, podemos provar que $d(\mathcal{C}) = w(\mathcal{C})$, a partir de (2.35):

$$\begin{aligned} d_{min}(\mathcal{C}) &= min\{d(u, v) : u, v \in \mathcal{C}, u \neq v\} \\ &= min\{w(u + v) : u, v \in \mathcal{C}, u \neq w\} \\ &= min\{w(x) : x \in \mathcal{C}, x \neq 0\} \\ &= w(\mathcal{C}). \end{aligned} \tag{2.36}$$

Exemplo 2.2.2 *Avaliemos o código $\mathcal{C} = \{000, 101, 010, 111\}$. A distância de Hamming entre cada um dos vetores é:*

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

Analisando cada um dos resultados, podemos chegar à conclusão de que a distância mínima de Hamming desse código é $d(\mathcal{C}) = 1$. Agora determinemos o peso de cada uma das palavras-código desse código:

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

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).

Teorema 2.2.1 *A capacidade de 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 $\left(\frac{d-1}{2}\right)$ erros.*

Seja $\mathcal{C}$ um código de bloco linear com parâmetros (n, k, d), como $\mathcal{C}$ é um subespaço vetorial k-dimensional de F_q^n, é possível encontrar k vetores linearmente independentes que geram $\mathcal{C}$. Podemos

arranjar tais vetores como linhas de uma matriz $k \times n$, que será denominada matriz geradora do código $\mathcal{C}$.

Definição 2.2.4 *A matriz $\mathbb{G}$ cujas linhas geram o código de bloco linear $\mathcal{C}$ é denominada matriz geradora de $\mathcal{C}$, e apresentada por*

$$\mathbb{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}, \tag{2.37}$$

em que $\mathbf{g}_{i0} = (\mathbf{g}_{i1}, \ldots, \mathbf{g}_{i,n-1})$, para $0 \leq i < k$.

Se $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ é a mensagem a ser codificada, então a palavra-código é dada por:

$$\begin{aligned} \mathbf{v} = \mathbf{u} \cdot G &= (u_0, u_1, \ldots, u_{k-1}) \cdot \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} \\ &= u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \ldots + u_{k-1}\mathbf{g}_{k-1}. \end{aligned} \tag{2.38}$$

De (2.38) vemos que um código linear (n, k) é completamente especificado pelas k linhas da matriz geradora $\mathbb{G}$, dessa forma o codificador somente necessita armazenar as k linhas de $\mathbb{G}$ para formar uma combinação linear dessas mesmas linhas, com base na mensagem $\mathbf{u}$.

Definição 2.2.5 *Uma matriz geradora é chamada sistemática se tem sua forma como:*

$$\mathbb{G} = \begin{bmatrix} \mathbb{I}_k & \mathbb{P} \end{bmatrix}, \tag{2.39}$$

na qual $\mathbb{I}_k$ é a matriz identidade de $k \times k$, ou de ordem k, já $\mathbb{P}$ é uma matriz de ordem $k \times (n-k)$.

Exemplo 2.2.3 *A palavra-código $\mathbf{v} = (1010110)$, de um código com parâmetros $(7, 3)$ é recebida e desejamos saber qual foi a mensagem enviada, conhecendo-se a matriz geradora $\mathbb{G}$, na forma sistemática:*

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

A mensagem enviada é $u = (u_0u_1u_2)$, portanto, utilizando-se (2.38), obtemos:

$$\begin{cases} u_0 = 1, \\ u_1 = 0, \\ u_2 = 1, \\ u_0 + u_2 = 0, \\ u_0 + u_1 = 1, \\ u_2 = 1, \\ u_0 + u_1 + u_2 = 0. \end{cases}$$

Identificamos que todas as equações foram satisfeitas, logo, por associação temos que $\mathbf{u} = (101)$ *foi a mensagem enviada.*

Definição 2.2.6 *Se uma matriz geradora de um código de bloco linear $\mathcal{C}$ com parâmetros (n, k, d) está em sua forma sistemática, então há uma matriz correspondente, denominada matriz controle de paridade*

$$\begin{aligned} \mathbb{H} &= \begin{bmatrix} \mathbb{I}_{n-k} & \mathbb{P}^T \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & \dots & 0 & p_{0,0} & p_{1,0} & \dots & h_{k-1,0} \\ 0 & 1 & \dots & 0 & p_{0,1} & p_{1,1} & \dots & h_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ddots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \dots & h_{k-1,n-k-1} \end{bmatrix}. \end{aligned} \tag{2.40}$$

Pode-se provar que $\mathbb{G} \cdot \mathbb{H}^T = 0$. Assim, para verificarmos se um vetor recebido $\mathbf{v}$ é uma palavra-código de $\mathcal{C}$, basta verificar se

$$\mathbf{v} \cdot \mathbb{H}^T = 0 \tag{2.41}$$

Caso essa igualdade não seja atingida, então o vetor recebido não pertence ao código.

Exemplo 2.2.4 *Consideremos um código $\mathcal{C}$ com parâmetros $(8, 3)$ e matriz geradora*

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

Desejamos verificar se as mensagens recebidas

$$\mathbf{v}_0 = (01001110),\ e$$

$$\mathbf{v}_1 = (11001101),$$

pertencem ao código. A partir da matriz $\mathbb{G}$, podemos determinar a matriz $\mathbb{H}$ conforme (2.40), já que a mesma se encontra na forma sistemática, caso contrário seria necessário que realizar operações elementares entre as linhas da matriz de modo a torná-la sistemática,

$$\mathbb{H} = \begin{bmatrix} \mathbb{I}_{n-k} & \mathbb{P}^T \end{bmatrix} = \begin{bmatrix} -\mathbb{P}^T & \mathbb{I}_{n-k} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Para verificarmos a condição (2.41), devemos encontrar a matriz controle de paridade transposta:

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

Finalmente:

$$\mathbf{v}_0 \cdot \mathbb{H}^T = (01001110) \cdot \mathbb{H}^T = (0000),$$
$$\mathbf{v}_1 \cdot \mathbb{H}^T = (11001101) \cdot \mathbb{H}^T = (1110),$$

e então, podemos concluir que $\mathbf{v}_0 \in \mathcal{C}$, e $\mathbf{v}_1 \notin \mathcal{C}$.

2.2.2 Decodificação

Suponhamos que em uma transmissão, $\mathbf{c}$ seja o vetor transmitido e $\mathbf{r}$ seja o vetor recebido. Caso $\mathbf{c} = \mathbf{r}$, então podemos afirmar que não houve erros na transmissão, entretanto, caso $\mathbf{c} \neq \mathbf{r}$, então houve erro(s) na transmissão.

Definição 2.2.7 *O vetor erro, $\mathbf{e}$, é definido por*

$$\mathbf{e} = \mathbf{r} - \mathbf{c}. \tag{2.42}$$

Exemplo 2.2.5 *Se $\mathbf{c} = (1100110011)$ foi o vetor transmitido e $\mathbf{r} = (1000100011)$ foi o vetor recebido, queremos determinar $\mathbf{e}$. Utilizando (2.42), temos que:*

$$\mathbf{e} = \mathbf{r} - \mathbf{c} = (1100110011) - (1000100011) = (0100010000).$$

Definição 2.2.8 *O vetor síndrome, $\mathbf{s}$, é definido por*

$$\mathbf{s} = \mathbf{r}\mathbb{H}^T. \tag{2.43}$$

A partir de relações entre a matriz controle de paridade $\mathbb{H}$ e o vetor erro $\mathbf{e}$, temos:

$$\mathbf{e}\mathbb{H}^T = (\mathbf{r}-\mathbf{c})\mathbb{H}^T = \mathbf{r}\mathbb{H}^T - \mathbf{c}\mathbb{H}^T \overset{0}{\cancel{\phantom{\mathbb{H}^T}}} = \mathbf{r}\mathbb{H}^T. \tag{2.44}$$

Dessa forma, dizemos que $\mathbf{r}$ e $\mathbf{e}$ possuem a mesma síndrome.

2.2.2.1 Algoritmo para Correção de 1 Erro

Seja $\mathcal{C}$ um código de bloco linear com distância mínima $d \geq 3$, vetor transmitido $\mathbf{c}$, vetor recebido $\mathbf{r}$, e vetor erro $\mathbf{e}$. Se $\mathbf{r}$ e $\mathbf{e}$, possuem a mesma síndrome, ou seja, $\mathbf{e}\mathbb{H}^T = \mathbf{r}\mathbb{H}^T = 0$, podemos dizer que não houve erros durante a transmissão e consideramos $\mathbf{r} = \mathbf{c}$.

Caso $\mathbf{e}\mathbb{H}^T \neq 0$, e ocorreu apenas 1 erro na transmissão, então $\mathbf{e} = (0, \ldots, \alpha, \ldots, 0)$, com $\alpha \neq 0$ na i-ésima posição. Logo $\mathbf{e}\mathbb{H}^T = \alpha h_i$, sendo que h_i é a i-ésima coluna da matriz controle de paridade. Para o vetor recebido, fazemos o processo inverso

$$\mathbf{r}\mathbb{H}^T = \alpha h_i, \tag{2.45}$$

e assim, o vetor erro $\mathbf{e}$ será apenas o vetor com todas as componentes nulas, salvo na i-ésima posição, α, na qual ocorreu o erro.

Exemplo 2.2.6 *Em uma transmissão, considere* $\mathbf{r} = (01111)$ *o vetor recebido. Conhecendo-se* $\mathbb{H}$*, desejamos determinar a quantidade de erros, e caso essa quantidade seja igual a um, corrigi-lo. Se*

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

Inicialmente, encontramos $\mathbb{H}^T$*:*

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

e assim,

$$\mathbf{r}\mathbb{H}^T = (001).$$

A síndrome $s = (001)$ *se refere a 5ª. coluna de* $\mathbb{H}$*, por isso* $\mathbf{e} = (00001)$*, então:*

$$\mathbf{c} = \mathbf{r} + \mathbf{e} = (01111) + (00001) = (01110).$$

Desta vez, verificamos que $\mathbf{c}\mathbb{H}^T = 0$*, portanto o erro foi corrigido.*

3 CÓDIGOS CÍCLICOS E CÓDIGOS CRC

Este capítulo apresentará um estudo sobre os códigos cíclicos e mais especificamente sobre uma categoria de códigos cíclicos conhecida como verificação de redundância cíclica, CRC (*Cyclic Redundancy Check*), que são essencialmente um tipo de códigos de blocos lineares. As principais referências deste capítulo foram [Forouzan 2008, Morelos-Zaragoza 2006, Murata e Ochiai 2017, Baicheva e Kazakov 2019].

3.1 CÓDIGOS CÍCLICOS

Os códigos cíclicos são códigos de bloco lineares, entretanto têm uma característica especial: se uma palavra-código é deslocada ciclicamente, ou seja, em rotação, obtemos outra palavra-código.

Definição 3.1.1 *Um código cíclico C é um código de bloco linear (n, k), com a propriedade de que todo deslocamento cíclico de uma palavra-código resulta em outra palavra-código. Os deslocamentos cíclicos de qualquer palavra finita $(c_0, c_1, \ldots, c_{n-1}) \in C$ são gerados através da escrita da sequência e do deslocamentos dos bits para a esquerda ou direita, quantas vezes for conveniente, de forma que o bit de uma extremidade se desloque ciclicamente para a outra, tal que:*

$$(c_{n-1}, c_0, \ldots, c_{n-2}) \in C,$$

de forma que todas a n-uplas são palavras-códigos do código, então:

$$(c_{n-2}, c_{n-1}, \ldots, c_{n-3}) \in C,$$

$$(c_{n-3}, c_{n-2}, \ldots, c_{n-1}) \in C,$$

$$\ldots$$

$$(c_1, c_2, \ldots, c_{n-1}, c_0) \in C.$$

Exemplo 3.1.1 *Consideremos um código $(3, 2)$ dado por*

$$c = \{000, 011, 101, 110\}.$$

Podemos observar que todos os deslocamentos cíclicos das palavras-código são palavras-código.

A representação dos códigos cíclicos também pode ser feita utilizando-se polinômios de grau $(n-1)$, com coeficientes zeros e uns no caso de códigos binários, de forma que o polinômio nulo equivale à palavra nula, e que os $(2^k - 1)$ polinômios não nulos são as palavras-código não nulas. Assim,

$$c = (c_0, c_1, \ldots, c_{n-1}) \leftrightarrow c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}. \tag{3.1}$$

Podemos obter os demais polinômios por meio da multiplicação de $c(x)$ por x:

$$x \times c(x) = x \times (c_0 + c_1x + \ldots + c_{n-1}x^{n-1}) = c_{n-1} + c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1}. \tag{3.2}$$

A codificação dos códigos cíclicos pode ser realizada utilizando um polinômio $g(x)$, denominado polinômio gerador. Este polinômio é único e deve satisfazer as condições de ter grau $(n-k)$ e ser um divisor de (x^n+1), ou seja, fazendo

$$\frac{g(x)}{x^n+1}, \tag{3.3}$$

obtém-se resto 0. Dessa forma, $c(x)$ é uma palavra-código se, e somente se, $g(x)$ for um divisor de $c(x)$.

Exemplo 3.1.2 *Desejamos comprovar se o polinômio $g(x) = 1 + x + x^3$ é de fato um polinômio gerador de um código cíclico com parâmetros $(7, 4)$. Inicialmente, observamos que $g(x)$ tem grau $7-4 = 3$. Além disso, para verificar que $g(x)$ é um polinômio gerador de um código $(7, 4)$, precisamos verificar que ele é um divisor de $x^n + 1$. Utilizando a Equação 3.3, e realizando a divisão módulo-2 de polinômios, temos que o quociente é $x^4 + x^2 + x = 1$ e o resto é igual a zero. Portanto, o polinômio em questão é o polinômio gerador.*

Por serem códigos de blocos lineares, os códigos cíclicos podem ser obtidos aplicando a codificação em uma mensagem de comprimento k, transformando-a em uma palavra-código de comprimento n. Seja C um código cíclico (n, k) com polinômio gerador

$$g(X) = g_0 + g_1X + g_2X^2 + \ldots + g_{n-k}X^{n-k}.$$

Para uma mensagem $m = (m_0, m_1, \cdots, m_{k-1})$ de comprimento k, na forma polinomial temos

$$m(x) = m_0 + m_1x + \cdots + m_{k-1}x^{k-1}. \tag{3.4}$$

Para codificar, inicialmente realizamos a multiplicação de $m(x)$ por x^{n-k} de modo a obter um polinômio de grau maior ou igual a $n-k$. Após isso, divide-se $x^{n-k}m(x)$ pelo polinômio gerador $g(x)$ de grau $(n-k)$, obtendo-se o resto $r(x)$. Adiciona-se o resto $r(x)$ a $x^{n-k}m(x)$, tendo como resultado o polinômio código $c(x)$. Assim, a palavra-código será dada na forma polinomial por

$$c(x) = r(x) + x^{n-k}m(x). \tag{3.5}$$

Exemplo 3.1.3 *Desejamos codificar a mensagem $m = (1001)$ em um código com parâmetros (7,4), com polinômio gerador $g(x) = 1 + x + x^3$. Podemos representá-la em sua forma polinomial como $m(x) = 1 + x^3$. Fazendo a multiplicação por $x^{n-k} = x^3$, temos $x^3m(x) = x^3(1 + x^3) = x^3 + x^6$, dividindo-se por $g(x)$, temos como resto $b(x) = x + x^2$. De forma que a palavra-código polinomial é:*

$$c(x) = r(x) + x^3m(x) = x + x^2 + x^3 + x^6,$$

Tabela 1 – Síndromes de erros

Líder	Polinômio líder	Polinômio síndrome	Síndrome
0000000	0	0	000
1000000	1	1	100
0100000	x	x	010
0010000	x^2	x^2	001
0001000	x^3	$1+x$	110
0000100	x^4	$x+x^2$	011
0000010	x^5	$1+x+x^2$	111
0000001	x^6	$1+x^2$	101

Fonte: próprio autor.

em notação binária, $c = (0111001)$.

A decodificação de um código cíclico pode ser feita utilizando o vetor síndrome assim como visto nos códigos de bloco lineares. Se o vetor síndrome s encontrado for diferente do vetor nulo, consideramos que houve erro na transmissão e, se possível, utilizamos uma estratégia de decodificação para encontrar este erro.

Seja $c = (c_0, c_1, \ldots, c_{n-1})$ a palavra-código transmitida por um canal ruidoso e $c' = (c'_0, c'_1, \ldots, c'_{n-1})$ a palavra recebida. Na forma polinomial, temos que $c'(x) = c'_0 + c'_1 x, \ldots, c'_{n-1}x^{n-1}$. O polinômio síndrome $s(x)$ de $c'(x)$, que tem grau $(n-k-1)$ ou menor, será o resto da divisão de $c'(x)$ pelo polinômio gerador $g(x)$, ou seja,

$$c'(x) = q(x)g(x) + s(x). \tag{3.6}$$

Se não houver erros na transmissão, $s(x) = 0$, ou seja, a palavra recebida será considerada a palava enviada. Caso haja erros na transmissão, teremos $s(x) \neq 0$. Como os códigos cíclicos são códigos de bloco lineares, tem-se que a síndrome da palavra recebida é igual à síndrome do erro. Se,

$$c'(x) = c(x) + e(x), \tag{3.7}$$

em que $e(x)$ é o polinômio erro, então o polinômio síndrome de $c'(x)$ e $e(x)$ são iguais. Conhecendo-se a tabela da síndrome dos erros, é possível corrigir a palavra recebida.

Exemplo 3.1.4 *Consideremos o código* $(7,4)$ *com polinômio gerador* $g(x) = 1+x+x^3$ *e polinômio controle de paridade* $h(x) = 1+x+x^2+x^4$. *Este código é capaz de corrigir 1 erro e tem tabela síndrome apresentada na Tabela 1. Seja* $r = (0110001)$ *a palavra recebida, temos que* $c'(x) = x+x^2+x^6$. *A síndrome de* $c'(x)$ *é o resto da divisão de* $(x+x^2+x^6)/(1+x+x^3)$, *que tem como resultado* $s(x) = 1+x$, *que pela Tabela 1, nos mostra que* $e(x) = x^3$. *Portanto,*

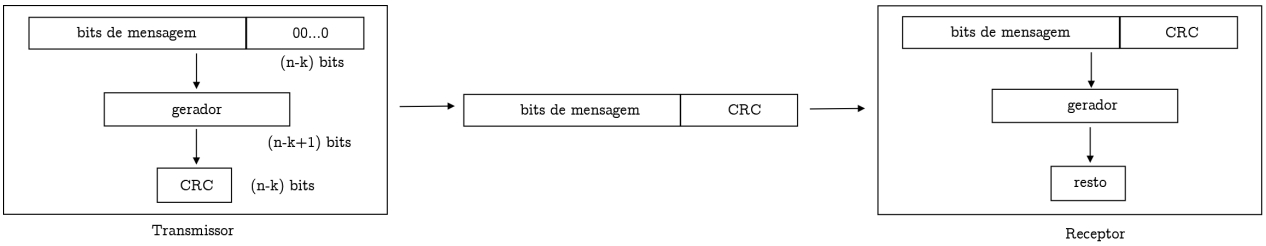
$$\begin{aligned} c(x) &= c'(x) + e(x) \\ &= x + x^2 + x^6 + x^3. \end{aligned}$$

3.2 CÓDIGOS CRC

O primeiro registro dos códigos de verificação de redundância cíclica, CRC (*Cyclic Redundancy Check*), datam de 1961, através da publicação realizada por W. Wesley Peterson, esses códigos são um tipo de códigos de bloco lineares (n, k), que fazem o uso das mesmas estratégias de codificação e decodificação dos códigos cíclicos, ou seja, utilizando um polinômio $g(x)$ de grau $n - k$ chamado de polinômio gerador. Porém, no caso dos códigos CRC o $g(x)$ nem sempre será um divisor de $x^n + 1$ como visto em (3.3), ele será será apenas predefinido e conhecido por ambas as partes do sistema. Dessa forma, os códigos CRC obtidos são na maioria dos casos códigos não-cíclicos.

Esses códigos também são um tipo de código de detecção de erro sistemático, onde um grupo de bits de controle de erro, os quais são o resto de uma divisão de um polinômio de mensagem por um polinômio gerador, são inseridos no final do bloco de mensagem, conforme apresenta o esquemático da Figura 8. E além da capacidade de correção de erros de bits, possuem capacidade considerável para a detecção de erros em rajada, que são vários erros em um mesmo bloco.

Figura 8 – Geração e verificação de CRC.



Fonte: próprio autor.

Definição 3.2.1 *Um código de verificação de redundância cíclica CRC, de parâmetros (n, k), é aquele no qual os k bits de uma mensagem m são codificados em uma palavra-código de n bits através da adição de $q = n - k$ bits obtidos utilizando um gerador g predefinido de $n - k + 1$ bits. Na forma polinomial, o resto $r(x)$ obtido da divisão de $x^{n-k}m(x)$ pelo gerador $g(x)$ de grau $n-k$, é adicionado a $x^{n-k}m(x)$ de modo a obtermos o polinômio código, ou seja,*

$$\begin{aligned} c(x) &= x^{n-k}m(x) + r(x) \\ &= c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \ldots + c_1x + c_0. \end{aligned} \tag{3.8}$$

Ao resto (ou redundância) $r = (r_{n-k-1}, r_{n-k-2}, \ldots, r_1, r_0)$ de tamanho $q = n - k$ chamamos de q-CRC.

De acordo com a definição anterior, a codificação CRC é realizada da mesma forma que a codificação de um código cíclico. O resto da divisão r de tamanho $q = n - k$ também conhecidos como bits de verificação de paridade ou redundância, é o vetor q-CRC utilizado em muitas aplicações para incorporar alguma outra técnica de codificação e decodificação.

Exemplo 3.2.1 *Consideremos a mensagem $m = (10110111)$, a ser codificada por um codificador CRC com parâmetros $(12, 8)$ e um gerador previamente estabelecido como $g = (10011)$ com 5 bits.*

Podemos escrever a mensagem bem como o gerador na forma polinomial

$$m(x) = x^7 + x^5 + x^4 + x^2 + x + 1,$$
$$g(x) = x^4 + x^1 + 1.$$

A mensagem $m(x)$ *será aumentada em* $q = 11 - 7 = 4$ *bits, tornando-se*

$$x^4 m(x) = x^{11} + x^9 + x^8 + x^6 + x^5 + x^4.$$

Através da divisão de polinômios módulo-2, podemos encontrar o resto ou redundância:

$$\begin{array}{llllllll|lll}
x^{11}+ & x^9+ & x^8+ & & x^6+ & x^5+ & x^4 & & x^4+ & x^1+ & 1 \\
\cline{9-11}
x^{11}+ & & x^8+ & x^7 & & & & & x^7+ & x^5+ & x^3 \\
\cline{1-5}
 & x^9+ & & x^7+ & x^6+ & x^5+ & x^4 & & & & \\
 & x^9+ & & & x^6+ & x^5 & & & & & \\
\cline{2-6}
 & & & x^7+ & & & x^4 & & & & \\
 & & & x^7+ & & & x^4+ & x^3 & & & \\
\cline{4-8}
 & & & & & & & x^3 & & &
\end{array}$$

Desta forma, obtemos a polinômio código $c(x) = x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3$*, ou equivalentemente a palavra-código* $c = (101101111000)$ *e resto 4-CRC* $r = (1000)$*.*

Analogamente à decodificação de códigos cíclicos, a decodificação de um código CRC pode ser feita através do cálculo da síndrome. Os bits são recebidos pelo decodificador, que similarmente ao codificador realiza a divisão módulo-2. A síndrome, que é o resto desta divisão é analisada, de forma que se a síndrome for composta somente por bits zeros, não houve erros na transmissão, logo retira-se os bits adicionais do CRC e obtém-se a mensagem original. Entretanto, se houver algum bit não-nulo, infere-se que houve erros na transmissão e aplica-se um algoritmo de decodificação apropriado para a aplicação. Neste trabalho, iremos utilizar os códigos CRC atrelados aos códigos polares sistemáticos, os quais serão apresentados no Capítulo 4. Veremos neste caso que um algoritmo de decodificação para um código CRC será apresentado juntamente com um algoritmo de decodificação para códigos polares sistemáticos.

Exemplo 3.2.2 *Consideremos um código CRC de parâmetros* $(12, 8)$ *com polinômio gerador predefinido* $g(x)$*:*

$$g(x) = x^4 + x^2 + 1.$$

Suponhamos que a palavra recebida foi $c' = (110011000011)$ *que na forma polinomial pode ser escrita como*

$$c'(x) = x^{11} + x^{10} + x^7 + x^6 + x^1 + 0,$$

Através da divisão de polinômios módulo-2, encontramos a síndrome (1100)*, que nos diz que a transmissão teve erros, pois* $s \neq 0$*. Assim, é necessário aplicar uma estratégia de decodificação adequada para identificar esse erro.*

$$
\begin{array}{lllll|lllll}
x^{11}+ & x^{10}+ & x^{6}+ & x+ & 1 & x^{4}+ & x^{2}+ & x+ & 1 & \\
\cline{6-10}
x^{11}+ & x^{9}+ & x^{8}+ & & & x^{7}+ & x^{6}+ & x^{5}+ & x^{3}+ & x \quad 1 \\
\cline{1-5}
 & x^{10}+ & x^{9}+ & x^{8}+ & x^{7}+ & x^{6}+ & x+ & 1 & & \\
 & x^{10}+ & x^{8}+ & x^{7}+ & & & & & & \\
\cline{2-6}
 & & x^{9}+ & x^{6}+ & x+ & 1 & & & & \\
 & & x^{9}+ & x^{7}+ & & & & & & \\
\cline{3-5}
 & & & x^{7}+ & x+ & 1 & & & & \\
 & & & x^{7}+ & x^{5}+ & x^{4}+ & & & & \\
\cline{4-6}
 & & & & x^{5}+ & x^{4}+ & x+ & 1 & & \\
 & & & & x^{5}+ & x^{3}+ & x & & & \\
\cline{5-7}
 & & & & & x^{4}+ & x^{3}+ & 1 & & \\
 & & & & & x^{4}+ & x^{2}+ & 1 & & \\
\cline{6-8}
 & & & & & & x^{3}+ & x^{2} & &
\end{array}
$$

4 CÓDIGOS POLARES

Este capítulo tem o intuito de apresentar os códigos polares, que foram introduzidos pela primeira vez em 2009 por Erdal Arikan. Esses códigos possibilitam uma transmissão de blocos de tamanho longos próximos ao limite de Shannon.

De maneira prática, nenhum canal possui a capacidade de evitar a totalidade dos efeitos de ruído nos sinais transmitidos, entretanto, a busca por métodos para a redução desses efeitos com a finalidade de uma reprodução mais fiel foi um dos fatores responsáveis pelo desenvolvimento dessa técnica. Estes códigos são derivados da polarização de canal, técnica em que os canais são classificados como canais ruidosos, ruins, e canais bons. Dessa forma, os bits de informação são transmitidos somente através dos canais bons, que são os canais com maior capacidade, a fim de que o efeito de ruído seja menor.

A Seção 4.1 apresenta a técnica da polarização de canal, posteriormente na Seção 4.2 apresenta-se a codificação não-sistemática e sistemática dos códigos polares. Já as técnicas de decodificação por cancelamento sucessivo, no caso não-sistemático, e por cancelamento sucessivo em lista, no caso sistemático, são apresentadas na Seção 4.3. As principais referências desta seção podem ser encontradas em [Arikan 2009, Lu e Goto 2011, Şaşoğlu 2011, Erazo 2017, Presman 2015, Wang et al. 2019, Sarkis et al. 2015, Vangala, Hong e Viterbo 2016, Arikan 2011].

4.1 POLARIZAÇÃO DE CANAIS

Consideremos um canal binário discreto e sem memória, B-DMC (*Binary-Discrete Memoryless Channel*, W com alfabeto de entrada X binário e alfabeto de saída Y, tal que:

$$W : X \longrightarrow Y. \tag{4.1}$$

Podemos expandir esse canal para representar $N = 2^n, n \in \mathbb{Z}^+$, usos de W, de forma que

$$W_N : X^N \longrightarrow Y^N, \tag{4.2}$$

e com probabilidade de transição dada por

$$W_N(y_1^N|x_1^N) = \prod_{i=1}^{N} W(y_i|x_i). \tag{4.3}$$

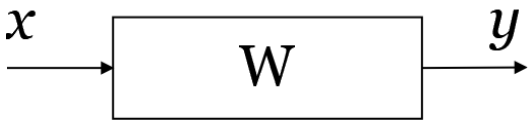
O parâmetro de interesse principal para analisar a eficiência deste canal é a capacidade dada por

$$I(W) \triangleq \sum_{y\in Y}\sum_{x\in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}, \tag{4.4}$$

em que $I(W)$ é a taxa mais alta na qual a comunicação em W, com entradas de igual frequência, é possível.

Iniciamos o processo recursivo para mostrar como é realizada a combinação dos canais, com o canal $W_1 \triangleq W$, ou seja, $N = 2^0 = 1$ cópia do canal W, como mostrado na figura Figura 9.

Figura 9 – Canal binário sem memória.



Fonte: próprio autor.

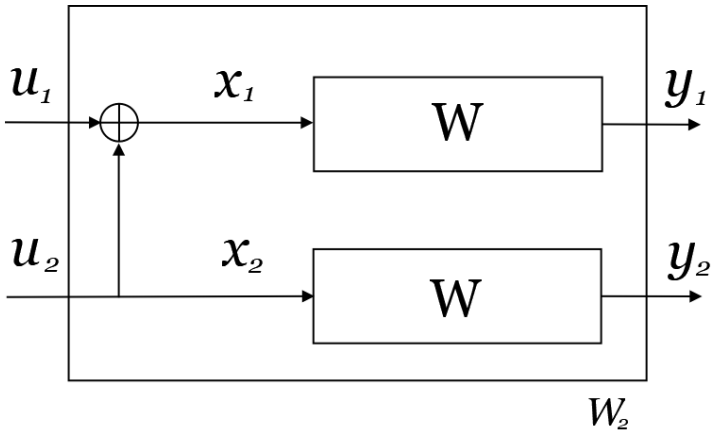
Para o canal $N = 2$, ilustrado na figura (Figura 10), temos a combinação de 2 canais W, ou seja,

$$W_2 = X^2 \longrightarrow Y^2, \tag{4.5}$$

sendo as probabilidades de transição dadas por

$$W_2(y_1^2|u_1^2) = W(y_1|u_1 \oplus U_2)W(y_2|u_2). \tag{4.6}$$

Figura 10 – Canal W_2.



Fonte: próprio autor.

Utilizando as probabilidades de transição, podemos 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 W_2.

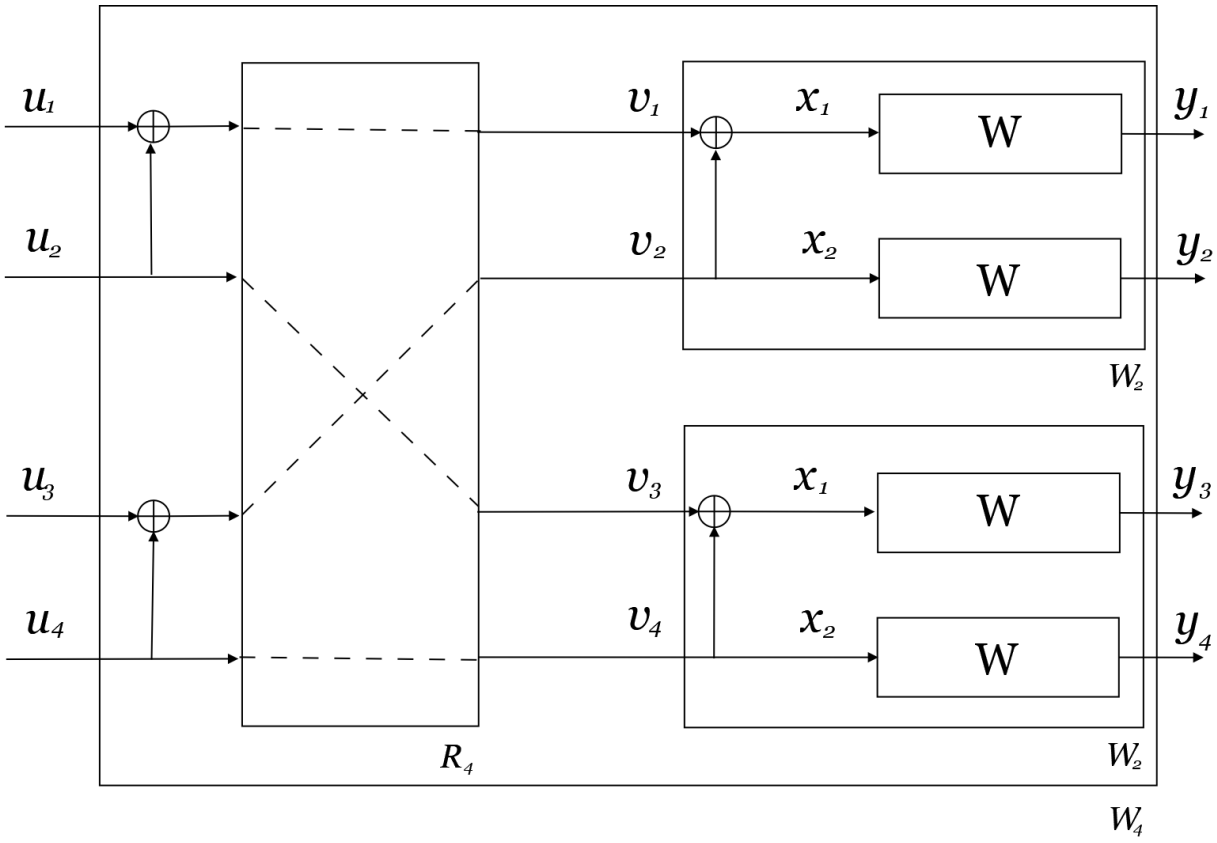
$$\mathbf{x}_1^2 = \mathbf{u}_1^2 G_2. \tag{4.7}$$

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{4.8}$$

O canal para $N = 4$, ilustrado na figura Figura 11, é a combinação de dois canais W_2

$$W_4 : X^4 \longrightarrow Y^4. \tag{4.9}$$

Figura 11 – Canal W_4.



Fonte: próprio autor.

Para esse nível de recursão temos que as probabilidades de transição são:

$$W_4(y_1^4|u_1^4) = W_2(y_1^2|u_1 \oplus u_2, u_3 \oplus u_4)W_2(y_3^4|u_2, u_4), \tag{4.10}$$

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,

$$\mathbf{s}_1^4 \longrightarrow v_1^4. \tag{4.11}$$

Analogamente, podemos descrever a matriz geradora por

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{4.12}$$

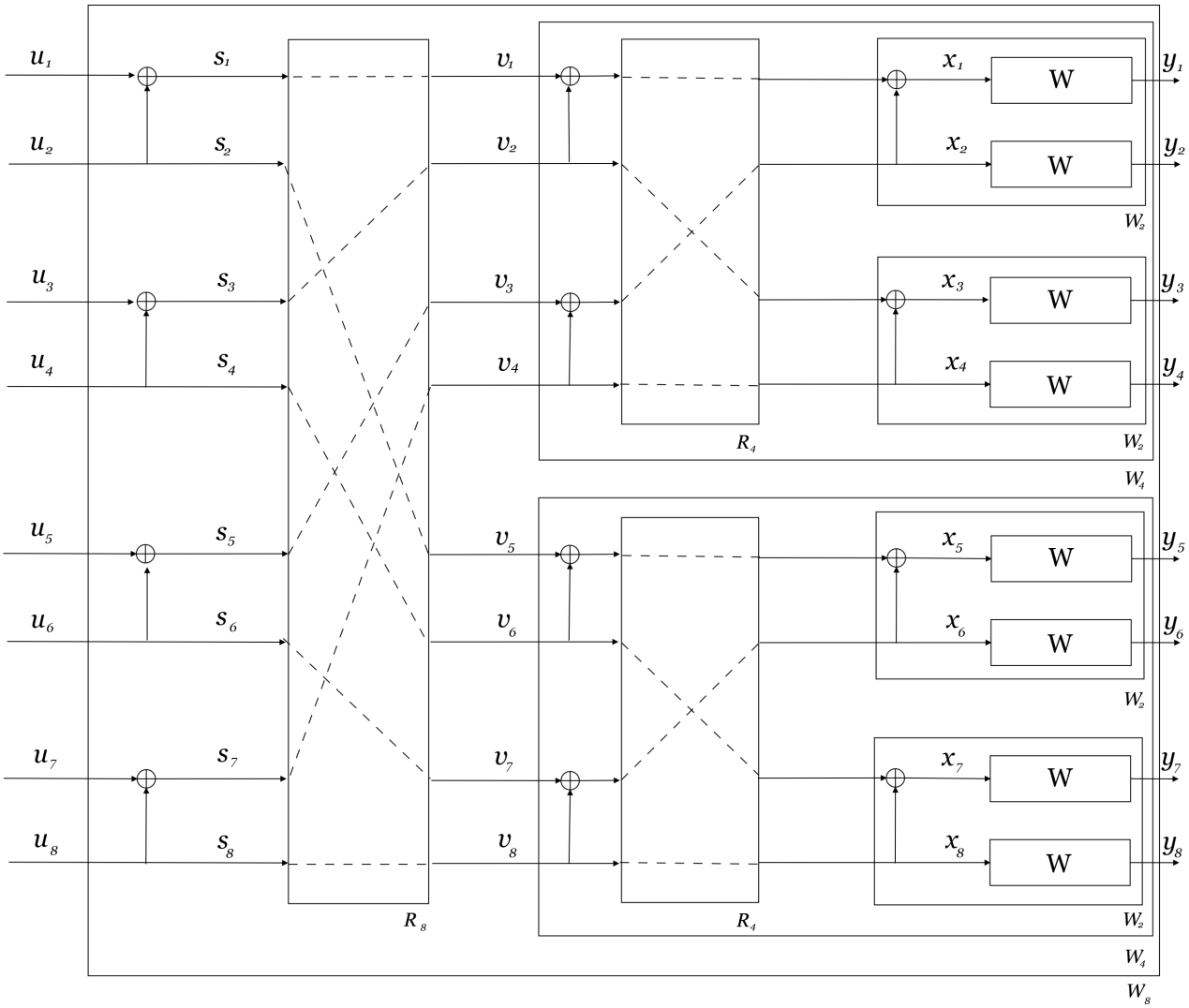
E, assim

$$\mathbf{x}_1^4 = u_1^4 G_4. \tag{4.13}$$

O próximo nível de recursão é o canal combinado para $N = 8$, apresentado na Figura 12. Nesse caso, há a combinação de duas cópias de W_4.

$$W_8 : X^8 \longrightarrow Y^8. \tag{4.14}$$

Figura 12 – Canal W_8.



Fonte: próprio autor.

Temos também neste caso a permutação do vetor de entradas $\mathbf{u}_1^8$ sendo transformado em $\mathbf{s}_1^8$, através de:

$$s_{2i-1} = u_{2i-1} \oplus u_{2i},\ e\ s_{2i} = u_{2i},\ para\ 1 \leq i \leq 8, \tag{4.15}$$

ou seja,

$$\mathbf{u}_1^8 = (u_1, u_2, u_3 \ldots, u_8) \longrightarrow \mathbf{s}_1^8 = (s_1, s_2, s_3, \ldots, s_8). \tag{4.16}$$

Dessa forma, o bloco R_8 operará sobre $\mathbf{s}_1^8$ através de uma matriz de permutação, descrita por

$$R_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.17}$$

Podemos também obter as probabilidades de transição do canal, conforme

$$W_8(\mathbf{y}_1^8|\mathbf{u}_1^8) = W(\mathbf{y}_1^4|u_1 \oplus u_2, u_3 \oplus u_4, u_5 \oplus u_6, u_7 \oplus u_8). \tag{4.18}$$

Dessa forma, podemos calcular x_1^8 a partir de $\mathbf{u}_1^8$

$$\mathbf{x}_1^8 = \mathbf{u}_1^8 G_8, \tag{4.19}$$

em que

$$G_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \tag{4.20}$$

é a matriz geradora para $N = 8$.

Podemos generalizar a recursão para um canal geral W_N, apresentado na Figura 13, 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.

Podemos verificar que a primeira cópia independente de $W_{N/2}$ recebe somente 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.

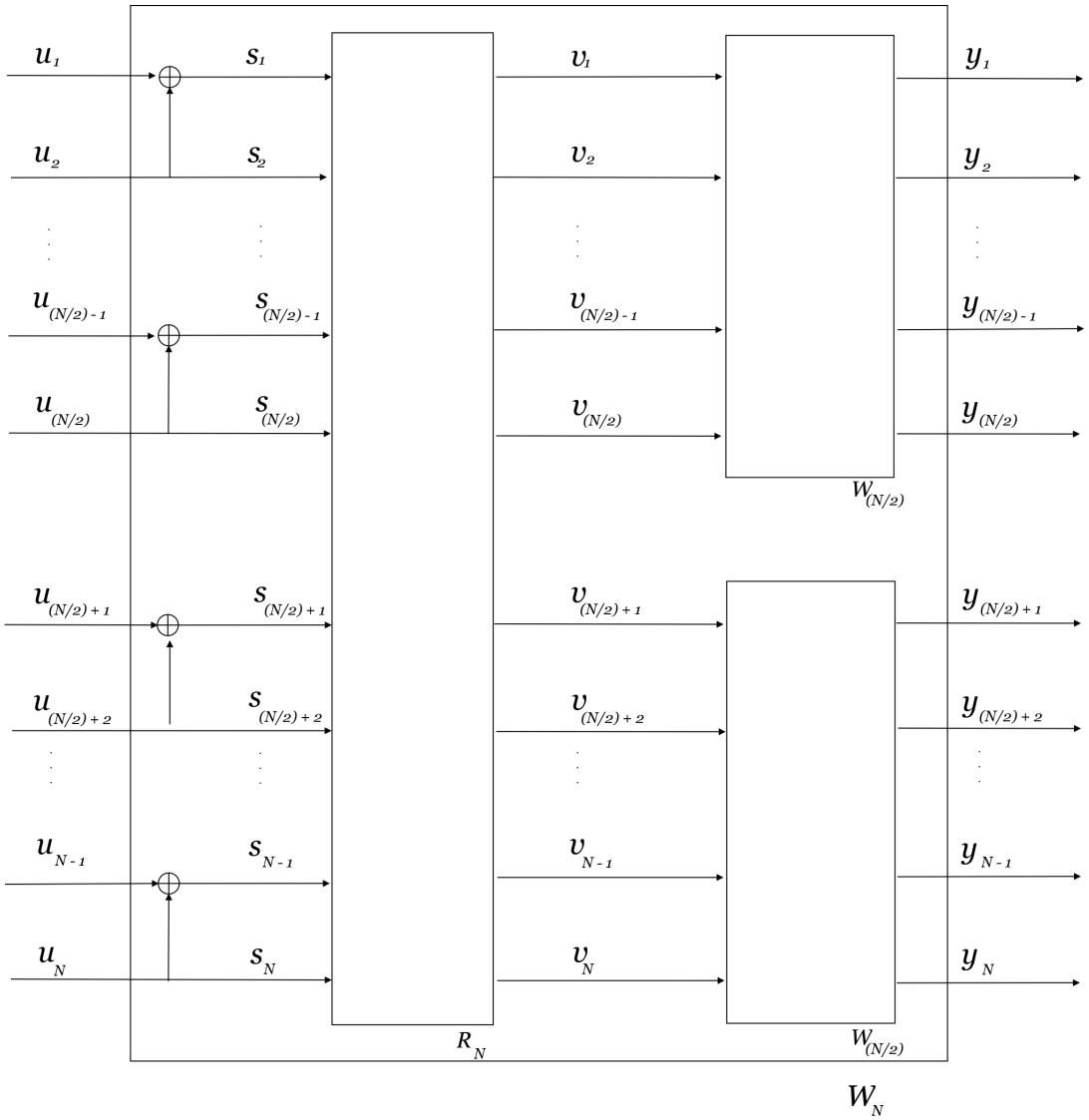
A entrada $\mathbf{u}_1^N$ em W_N é transformada em s_1^N, através de , desta vez porém com $(1 \leq i \leq N/2)$:

$$s_{2i-1} = u_{2i-1} \oplus u_{2i}, \ e \ s_{2i} = u_{2i}, \ \text{para } 1 \leq i \leq N/2. \tag{4.21}$$

O operador R_N realiza a permutação conhecida como embaralhamento reverso, de forma a transformar s_1^N em v_1^N, que será a entrada das cópias de $W_{N/2}$:

$$s_1^N \longrightarrow v_1^N = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N). \tag{4.22}$$

Figura 13 – Canal W_N.



Fonte: próprio autor.

Podemos então, mapear $\mathbf{x}_1^N$ através de u_1^N e da matriz geradora

$$\mathbf{x}_1^N = u_1^N G_N. \tag{4.23}$$

O processo de divisão de canais se dá então através da divisão de W_N em um conjunto de canais de W_N^i,

$$W_N^{(i)} : X \longrightarrow Y^N \times X^{i-1},\ 1 \leq i \leq N, \tag{4.24}$$

definidos pelas probabilidades de transição,

$$W_N^{(i)}\left(y_1^N, u_1^{i-1} | u_i\right) \triangleq \sum_{u_{i+1}^N \in X^{N-1}} \frac{1}{2^{N-1}} W_N\left(y_1^N | u_1^N\right), \tag{4.25}$$

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

O processo de cálculo da probabilidades de transição para quaisquer tipos de canais não é trivial, de forma que, se analisarmos um canal BEC com probabilidade de apagamento $e = 1/2$, como feito por [Arikan 2009], verificamos a simplificação dos cálculos das capacidades para cada canal, como mostram (4.26) e (4.27):

$$I\left(W_N^{(2i-1)}\right) = I\left(W_i^{(N/2)}\right)^2, \tag{4.26}$$

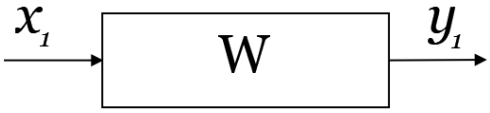
$$I\left(W_N^{(2i)}\right) = 2I\left(W_i^{(N/2)}\right) - I\left(W_i^{(N/2)}\right)^2. \tag{4.27}$$

Nessa caso, $I(W_1) = 1 - e$; os canais bons serão descritos por W^+ e os canais ruins por W^-.

Exemplo 4.1.1 *Vamos determinar as capacidades de canais BEC com probabilidade de apagamento* $p = 0,4$ *através do método de polarização de canal com* $N = 1, 2, 4e8$*. Para* $N = 1$*, podemos calcular a capacidade com o auxílio de (2.30):*

$$I(W) = 1 - p = 1 - 0,4 = 0,6.$$

Figura 14 – Exemplo de canal W_1.



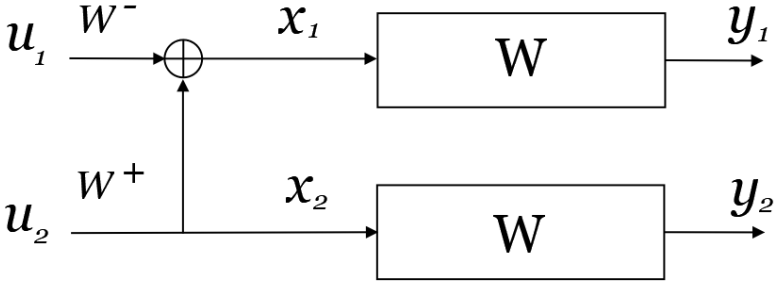
Fonte: próprio autor.

Para $N = 2$ *(Figura 15), os canais* W^- *e* W^+*, dividem a capacidade total; podemos calculá-las através de:*

$$I(W^-) = I(W)^2 = 0,6^2 = 0,36,$$
$$I(W^+) = 2I(W) - I(W)^2 = 2(0,6) - 0,6^2 = 0,84.$$

Como a capacidade de $I(W^-)$ *é menor que a de* $I(W^+)$*, esse canal será congelado e a comunicação se dará através de* $I(W^+)$*.*

Figura 15 – Exemplo de canal W_2.

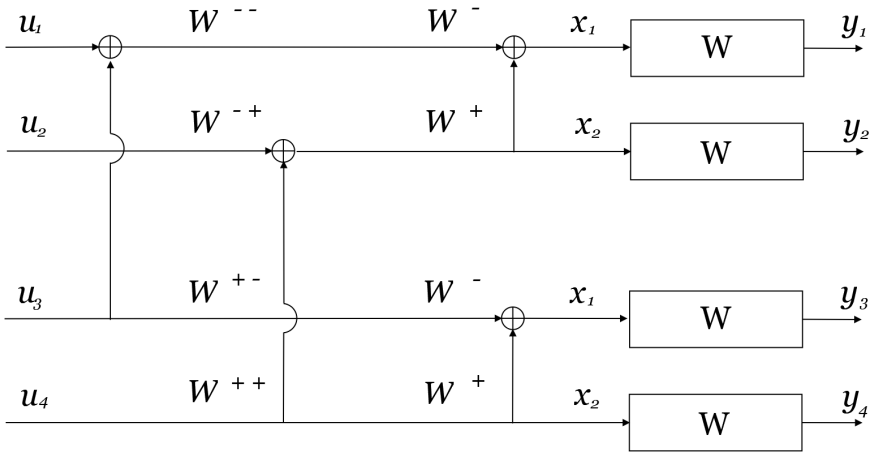


Fonte: próprio autor.

De maneira semelhante, para $N = 4$ (Figura 16) a capacidade será dividida em 4 canais; podemos determinar a capacidade de cada canal da seguinte forma:

$$
\begin{aligned}
&I(W^{--}) = I(W^{-})^2 = (0,36)^2 = 0,1296,\\
&I(W^{-+}) = 2I(W^{-}) - I(W^{-})^2 = 2(0,36) - (0,36)^2 = 0,5904,\\
&I(W^{+-}) = I(W^{+})^2 = (0,84)^2 = 0,7056,\\
&I(W^{++}) = 2I(W^{+}) - I(W^{+})^2 = 2(0,84) - (0,84)^2 = 0,9744.
\end{aligned}
$$

Figura 16 – Exemplo de canal W_4.



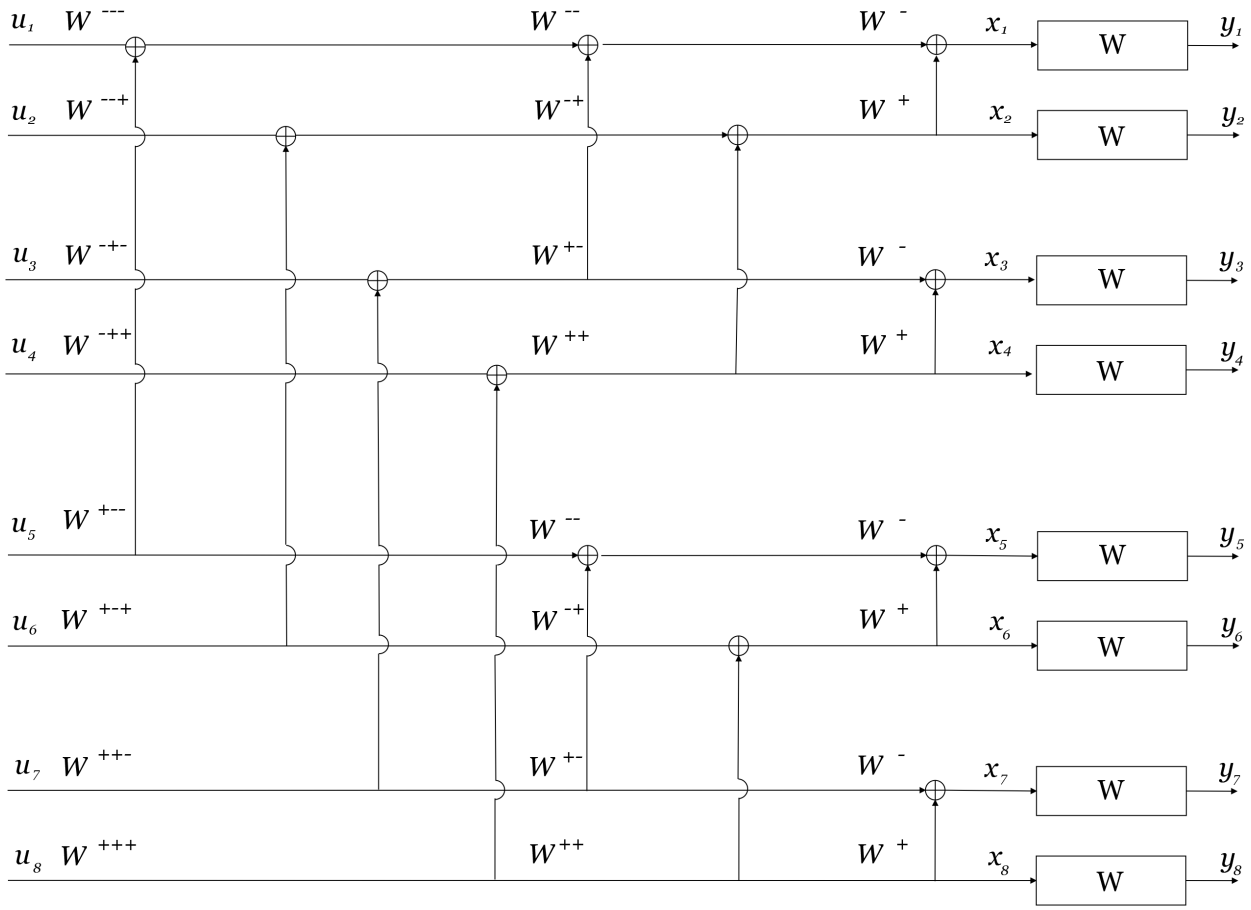
Fonte: próprio autor.

Verificamos que os canais $I(W^{+-})$ e $I(W^{++})$ possuem as maiores capacidades, portanto a comunicação se dará através destes, enquanto que $I(W^{--})$ e $I(W^{-+})$ serão congelados. Para $N = 8$, a capacidade será dividida em 8 canais, e de maneira similar à anterior, podemos calcular as mesmas:

$$
\begin{aligned}
&I(W^{---}) = I(W^{--})^2 = (0,1296)^2 = 0,01679616,\\
&I(W^{--+}) = 2I(W^{--}) - I(W^{--})^2 = 2(0,1296) - (0,1296)^2 = 0,24240384,\\
&I(W^{-+-}) = I(W^{-+})^2 = (0,5904)^2 = 0,34857216,\\
&I(W^{-++}) = 2I(W^{-+}) - I(W^{-+})^2 = 2(0,5904) - (0,5904)^2 = 0,83222784,\\
&I(W^{+--}) = I(W^{+-})^2 = (0,7056)^2 = 0,49787136,\\
&I(W^{+-+}) = 2I(W^{+-}) - I(W^{+-})^2 = 2(0,7056) - (0,7056)^2 = 0,91332864,\\
&I(W^{++-}) = I(W^{++})^2 = (0,9744)^2 = 0,94945536,\\
&I(W^{+++}) = 2I(W^{++}) - I(W^{++})^2 = 2(0,9744) - (0,9744)^2 = 0,99934464.
\end{aligned}
$$

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,

Figura 17 – Exemplo de canal W_8.



Fonte: próprio autor.

portanto congelamos os de menores capacidade e escolhemos os canais com as maiores capacidades para a transmissão.

Através do exemplo anterior, podemos verificar que à medida que aumentamos o número de canais, as capacidades dos 2^N canais tende a 0 ou a 1. Nos canais bons, a capacidade tende a 1 e nos canais ruins a capacidade tende a 0.

4.2 CODIFICAÇÃO

Nesta seção apresentamos os dois tipos de codificação polar. Primeiramente, na Subseção 4.2.1, apresentamos a estratégia de codificação não-sistemática dos códigos polares, e posteriormente, na Subseção 4.2.2, a estratégia de codificação sistemática dos códigos polares, em que é possível identificar a mensagem original.

4.2.1 Codificação Não-Sistemática

A codificação polar não-sistemática segue a codificação obtida pela técnica de polarização de canal apresentada na Seção 4.1 e é determinada através de:

$$\mathbf{x}_1^N = \mathbf{u}_1^N G_N, \tag{4.28}$$

sendo

- $\mathbf{x}_1^N = (x_1, x_2, \ldots, x_N) \in X^N$ é um vetor palavra-código;
- $\mathbf{u}_1^N = (u_1, u_2, \ldots, u_N) \in X^N$ é um vetor de informação;
- G_N é a matriz geradora de dimensões $N \times N$,

em que $N = 2^n$, para $n \geq 0$.

Para o caso em que $N = 2$, vimos que a matriz geradora de dimensões 2×2 é expressa como:

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = F. \tag{4.29}$$

Essa matriz primitiva também é conhecida como kernel de Arikan, ou simplesmente kernel. Podemos definir uma matriz geradora de dimensões arbitrárias através do produto tensorial de matrizes, também conhecido como produto de Kronecker, de forma que

$$G_N = B_N F^{\otimes n}, \tag{4.30}$$

B_N é a matriz permutação de inversão de bit dada por

$$B_N = R_N(B_{N/2} \otimes I_2) \tag{4.31}$$

e $F^{\otimes n}$ é a potência de Kronecker definida por

$$F^{\otimes n} = F \otimes F \otimes \ldots \ (n \ vezes). \tag{4.32}$$

Atualmente, a fim de simplificar a implementação de códigos polares, é mais comum a utilização de

$$G_N = F^{\otimes n} = \begin{bmatrix} F^{\otimes n-1} & 0 \\ F^{\otimes n-1} & F^{\otimes n-1} \end{bmatrix}, \tag{4.33}$$

sendo que por definição $F^{\otimes 1} = F$ e $F^{\otimes 0} = 1$. Para essa implementação, a equação de codificação torna-se:

$$\mathbf{x}_1^N = \tilde{\mathbf{u}}_1^N F^{\otimes n}, \tag{4.34}$$

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

Exemplo 4.2.1 *Desejamos obter $F^{\otimes n}$ para $N = 1, 2, 4$ e 8. Podemos utilizar a equação (4.32) para obter a potência de Kronecker para cada um dos casos:*

- $N = 1$*:*

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

- $N = 2$*:*

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

- $N = 4$*:*

$$F^{\otimes 2} = \begin{bmatrix} F^{\otimes 1} & 0 \\ F^{\otimes 1} & F^{\otimes 1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

- $N = 8$*:*

$$F^{\otimes 3} = \begin{bmatrix} F^{\otimes 2} & 0 \\ F^{\otimes 2} & F^{\otimes 2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

O vetor palavra-código resultante de $\mathbf{x}$ para um codificador não-sistemático pode ser representado como uma co-classe de G_N:

$$\mathbf{x} = \mathbf{u}G_N = \mathbf{u}_A G_A \oplus \mathbf{u}_{A^c} G_{A^c}, \tag{4.35}$$

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, ($u_{A^c} : u_j \in A^c$).
- $u_{A^c} = u - u_A$.

Definição 4.2.1 *Códigos polares não-sistemáticos são caracterizados através dos parâmetros* (N, k, A, μ_{A^c})*, 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}*.*

Exemplo 4.2.2 *Desejamos codificar a mensagem* $u = (10110010)$ *através de um codificador polar não-sistemático com parâmetros (8,4,{4,6,7,8},(0,0,0,0)). Inicialmente, identificamos os parâmetros do código:*

- $N = 8$*, é o comprimento do código,*
- $k = 4$*, é a dimensão do código,*
- $A = \{4, 6, 7, 8\}$.

Sabemos, por (4.30), que $G_N = B_N F^{\otimes n}$, como $N = 8$, logo, temos que $G_8 = B_8 F^{\otimes 3}$. Podemos calcular B_8 recursivamente através de (4.31):

$$\begin{cases} B_8 = R_8(B_4 \otimes I_2), \\ B_4 = R_4(B_2 \otimes I_2), \\ B_2 = R_2(B_1 \otimes I_2). \end{cases}$$

Determinemos a matriz permutação B_2:

$$B_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \left(1 \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Substituímos então B_2 na equação para o cálculo de B_4,

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

Podemos agora substituir B_4 na equação para o cálculo de B_8:

$$\begin{aligned} B_8 &= R_8(R_4 \otimes I_2) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Com B_8, podemos utilizar a equação 4.30 para obtermos a matriz geradora G_8:

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

Finalmente, através de 4.28, temos que:

$$
\begin{aligned}
\mathbf{x}_1^8 &= \mathbf{u}_1^8 G_8 \\
&= [11110001] \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 1&0&0&0&1&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&0&0&0&0&0&0 \\ 1&1&0&0&1&1&0&0 \\ 1&1&1&1&0&0&0&0 \\ 1&1&1&1&1&1&1&1 \end{bmatrix} \\
&= [11111101].
\end{aligned}
$$

O mesmo resultado pode ser obtido com a utilização de (4.34):

$$\begin{aligned}\mathbf{x}_1^8 &= \tilde{\mathbf{u}}_1^8 F^{\otimes 3} \\ &= [10101011] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\ &= [11111101].\end{aligned}$$

4.2.2 Codificação Sistemática

A codificação sistemática de códigos polares foi proposta por Arikan em 2011, visto que um dos obstáculos para o uso das implementações em hardware dos codificadores polares não-sistemáticos é a alta demanda de recursos computacionais para se obter a mensagem enviada. Apesar da codificação sistemática apresentar maior complexidade que a codificação não-sistemática, há vantagens em relação ao desempenho de BER.

A principal propriedade que diferencia uma palavra-código proveniente de um codificador polar sistemático de um codificador polar não-sistemático é que nela os bits não congelados do vetor de mensagem aparecem de maneira explícita na palavra-código.

Começaremos nossa análise através da Equação 4.35. Para isso, dividiremos a palavra-código em $\mathbf{x} = x_B + x_{B^c}$, sendo que B é um sub-conjunto arbitrário de $\{1, \ldots, N\}$. Assim, temos que

$$\mathbf{x}_B = \mathbf{u}_A G_{AB} \oplus \mathbf{u}_{A^c} G_{A^c B}, \tag{4.36}$$

e

$$\mathbf{x}_{B^c} = \mathbf{u}_A G_{AB^c} \oplus \mathbf{u}_{A^c} G_{A^c B^c}, \tag{4.37}$$

em que G_{AB} é uma submatriz de G, que consiste do vetor de elementos $G_{i,j}$, $i \in A$, $j \in B$; as outras submatrizes são definidas do mesmo modo. Nosso objetivo é encontrar codificadores sistemáticos nos quais $\mathbf{x}_B$ desempenha o mesmo papel de $\mathbf{u}_A$ nos codificadores não-sistemáticos, mantendo-se $\mathbf{u}_{A^c}$ fixo.

Definição 4.2.2 *Para qualquer código polar definido por um codificador polar não-sistemático com parâmetros $(A, \mathbf{u}_{A^c})$, há um codificador sistemático com parâmetros $(B, \mathbf{u}_{A^c})$, se, e somente se, A e B têm o mesmo número de elementos e G_{AB} é uma matriz inversível.*

Se as condições apresentadas na Definição 4.2.2 são satisfeitas, o mapeamento de $\mathbf{x}_B \mapsto \mathbf{x} = (\mathbf{x}_B, \mathbf{x}_{B^c})$ pode ser realizado através do cálculo de

$$\mathbf{u}_A = (\mathbf{x}_B - \mathbf{u}_{A^c} G_{A^c B})(G_{AB})^{-1}. \tag{4.38}$$

Assim, o resultado encontrado em (4.38) pode ser utilizado para o cálculo de (4.37). Desenvolveremos a Equação 4.33, de forma a evidenciar G_N:

$$G_N = F^{\otimes n} = \begin{bmatrix} F^{\otimes n-1} & 0 \\ F^{\otimes n-1} & F^{\otimes n-1} \end{bmatrix} = \begin{bmatrix} G_{N/2} & 0_{N/2} \\ G_{N/2} & G_{N/2} \end{bmatrix}. \tag{4.39}$$

Verificamos que a matriz geradora descrita em (4.39) é uma matriz triangular inferior, na qual a diagonal principal é composta de uns, também podemos mencionar que uma matriz inversível e que ${G_N}^{-1} = G_N$. Similarmente, 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. Observemos que a codificação sistemática pode ser realizada com complexidade $O(N \log N)$, através da solução da equação

$$\mathbf{x} = \mathbf{u} G_N, \tag{4.40}$$

dado $(\mathbf{u}_{A^c}, \mathbf{x}_A)$. Separemos a palavra-código sistemática em duas partes, $\mathbf{x}^{(1)}$, metade inicial da palavra, e $\mathbf{x}^{(2)}$, metade final da palavra, ou seja:

$$\mathbf{x} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right), \tag{4.41}$$

de forma que,

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

Podemos estender esse raciocínio a $\mathbf{u}$, e assim reescrever (4.40) como:

$$\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = \left(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}\right) \begin{bmatrix} G_{N/2} & 0_{N/2} \\ G_{N/2} & G_{N/2} \end{bmatrix}. \tag{4.42}$$

Precisamos então encontrar a solução para:

$$\mathbf{x}^{(2)} = \mathbf{u}^{(2)} G_{N/2}, \tag{4.43}$$

dado $\left(\mathbf{u}^{(2)}_{A_2^c}, \mathbf{x}^{(2)}_{A_2}\right)$, e $A_2 = \{i \in \{1, \ldots, N/2\} : i + N/2 \in A\}$. Supondo que (4.43) foi resolvida, então podemos escrever que

$$\mathbf{x}^{(1)} = \mathbf{u}^{(1)} G_{N/2} + \mathbf{x}^{(2)}, \tag{4.44}$$

e posteriormente, podemos rearranjar a equação anterior, de forma a evidenciar $\mathbf{x}^{(1)}$ e $\mathbf{x}^{(2)}$.

$$\left(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\right) = \mathbf{u}^{(1)} G_{N/2}. \tag{4.45}$$

Para resolvermos (4.45), temos $\mathbf{u}_{A_1^c}^{(1)}$ e $\left(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\right)_{A_1}$, sendo que $A_1 = \{i \in \{1, \ldots, N/2\} : i \in A\}$. A complexidade para a resolução total do problema pode ser descrita através de:

$$\mathcal{X}_N \leq 2\mathcal{X}_{N/2} + \alpha N, \tag{4.46}$$

onde,

- $\mathcal{X}_{N/2}$ representa o caso de maior complexidade para a resolução de (4.43) dentre todas as possibilidades de escolha para A_2.
- αN, para alguma constante α, representa o trabalho necessário para a realização das operações de subtração entre $\mathbf{x}_{A_2}^{(2)}$ e $\mathbf{x}_{A_1}^{(1)}$, substituição do resultado em (4.45), e então, a adição de $\mathbf{x}_2$ a $\left(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\right)$.

As inequações recursivas descritas por (4.46) estabelecem o limite de complexidade:

$$\mathcal{X}_N \leq \alpha N \log_2 N. \tag{4.47}$$

Exemplo 4.2.3 *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 $\mathbf{u} = (00010001)$. *Inicialmente, precisamos definir, para este caso, as matrizes apresentadas nas Equações 4.36 e 4.37,* G_{AB}, G_{AB^c}, G_{A^cB} *e* $G_{A^cB^c}$, *a partir de* G_8. *Temos que:*

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

$$(G_8)_{AB^c} = \begin{bmatrix} G_{8_{(1,4)}} & G_{8_{(1,6)}} & G_{8_{(1,7)}} & G_{8_{(1,8)}} \\ G_{8_{(2,4)}} & G_{8_{(2,6)}} & G_{8_{(2,7)}} & G_{8_{(2,8)}} \\ G_{8_{(3,4)}} & G_{8_{(3,6)}} & G_{8_{(3,7)}} & G_{8_{(3,8)}} \\ G_{8_{(5,4)}} & G_{8_{(5,6)}} & G_{8_{(5,7)}} & G_{8_{(5,8)}}. \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$(G_8)_{A^cB} = \begin{bmatrix} G_{8_{(4,1)}} & G_{8_{(4,2)}} & G_{8_{(4,3)}} & G_{8_{(4,5)}} \\ G_{8_{(6,1)}} & G_{8_{(6,2)}} & G_{8_{(6,3)}} & G_{8_{(6,5)}} \\ G_{8_{(7,1)}} & G_{8_{(7,2)}} & G_{8_{(7,3)}} & G_{8_{(7,5)}} \\ G_{8_{(8,1)}} & G_{8_{(8,2)}} & G_{8_{(8,3)}} & G_{8_{(8,5)}}. \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$(G_8)_{A^cB^c} = \begin{bmatrix} G_{8_{(1,1)}} & G_{8_{(1,2)}} & G_{8_{(1,3)}} & G_{8_{(1,5)}} \\ G_{8_{(2,1)}} & G_{8_{(2,2)}} & G_{8_{(2,3)}} & G_{8_{(2,5)}} \\ G_{8_{(3,1)}} & G_{8_{(3,2)}} & G_{8_{(3,3)}} & G_{8_{(3,5)}} \\ G_{8_{(5,1)}} & G_{8_{(5,2)}} & G_{8_{(5,3)}} & G_{8_{(5,5)}}. \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

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* $\mathbf{u} = 00010001$*, definimos* $\mathbf{x}_B = (u_4\ u_6\ u_7\ u_8) = (1001)$*. 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 (4.38):*

$$\begin{aligned}
\mathbf{u}_A &= (\mathbf{x}_B - \mathbf{u}_{A^c} G_{A^c B})(G_{AB})^{-1} \\
&= \left((1001) - (0000) \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \\
&= (1001) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = (0111).
\end{aligned}$$

Substituímos então $\mathbf{u}_A$ *em (4.37):*

$$\begin{aligned}
\mathbf{x}_{B^c} &= \mathbf{u}_A G_{AB^c} \oplus \mathbf{u}_{A^c} G_{A^c B^c} \\
&= (1001) \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \oplus (0000) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = (1001).
\end{aligned}$$

Em posse de $\mathbf{x}_B = (1001)$ *e* $\mathbf{x}_{B^c} = (1001)$*, obtemos a palavra-código sistemática* $\mathbf{x} = (10011001)$.

Outro método possível para a codificação polar sistemática é através do uso de dois codificadores polares não-sistemáticos em série. Para este codificador polar sistemático, inicialmente possuímos a mensagem $\mathbf{u} = (u_1, u_2, \ldots, u_N)$, de comprimento N, que é a entrada do primeiro subcircuito. Os bits congelados dessa mensagem definem as linhas congeladas do circuito completo.

O primeiro subcircuito realiza a operação de multiplicação da mensagem $\mathbf{u}$ pela matriz $F^{\otimes n}$, conforme apresenta a equação 4.48, o vetor $\mathbf{x}'$ é a saída do primeiro circuito.

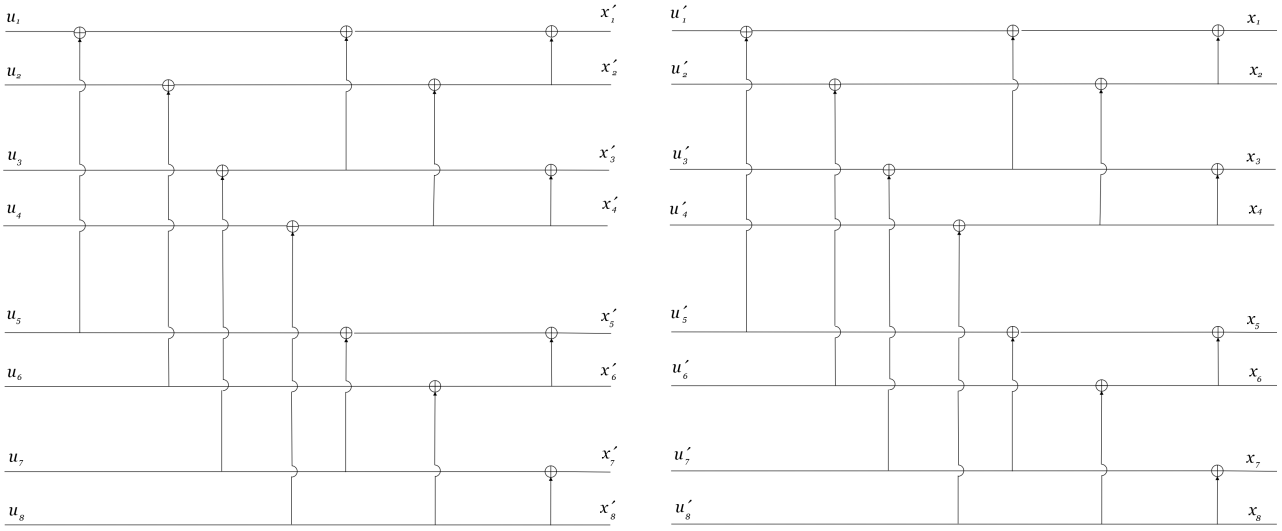
$$\mathbf{u} F^{\otimes n} = \mathbf{x}'. \tag{4.48}$$

Após isso, os bits de $\mathbf{x}'$ relativos às entradas congeladas, ou seja, que não pertencem à A, são igualados a zero, e esse novo vetor, denominado $\mathbf{u}'$, é a entrada do segundo subcircuito. Podemos então obter a palavra-código sistemática através de

$$\mathbf{u}' F^{\otimes n} = \mathbf{x}. \tag{4.49}$$

O circuito codificador sistemático da Figura 18, ilustra a realização dessas operações para o caso no qual $N = 8$.

Figura 18 – Codificador sistemático para $N = 8$.



Fonte: próprio autor.

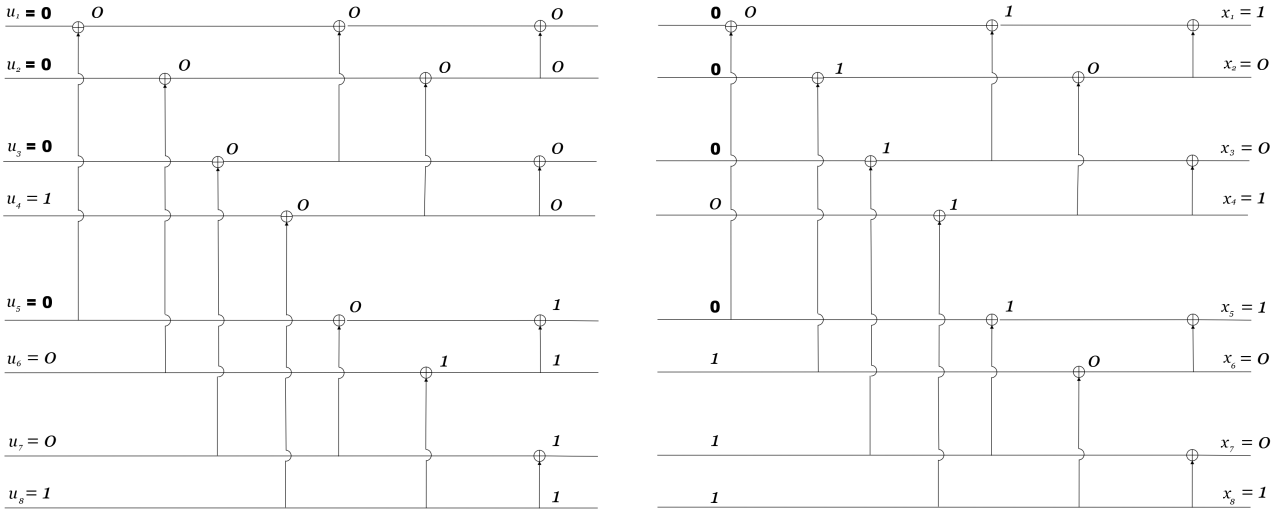
Exemplo 4.2.4 *Utilizando o circuito codificador sistemático da Figura 18, desejamos codificar a mensagem* $\mathbf{u} = (00010001)$*, com o intuito de obter como palavra-código o mesmo resultado do Exemplo 4.2.2, por isso, consideraremos os mesmos parâmetros. Como primeiro passo, identificamos que as linhas não-congeladas do codificador são as linhas 4, 6, 7 e 8, dessa forma, temos que as linhas 1, 2, 3 e 5 são congeladas. Como entrada do primeiro subcircuito da Figura 19 (extremidade esquerda) temos a mensagem* $\mathbf{u} = (00010001)$*, sendo que os bits representados em negrito são as linhas congeladas. Realizando as operações de soma módulo-2, no sentido da esquerda para direita, obtemos* $\mathbf{x}' = (00001111)$*, como saída do primeiro circuito. Através da operação de congelamento dos bits de* $\mathbf{x}'$ *referentes às linhas congeladas 1, 2, 3 e 5, obtemos o vetor* $\mathbf{u}' = (00000111)$*, que é a entrada do segundo subcircuito; semelhantemente ao primeiro subcircuito, realizamos as operações e obtemos como saída do segundo subcircuito a palavra-código sistemática* $\mathbf{x} = (10011001)$*, que é a mesma palavra-código do exemplo anterior. Podemos verificar que os bits da mensagem* $\mathbf{u}$ *referentes às linhas não congeladas aparecem de maneira explícita em* $\mathbf{x}$*:*

$$\mathbf{u} = (000\mathbf{1}0\mathbf{001}),$$
$$\mathbf{x} = (100\mathbf{11}\mathbf{001}).$$

4.3 DECODIFICAÇÃO

Nesta seção apresentamos a decodificação de códigos polares. Inicialmente, na Subseção 4.3.1, apresentaremos um algoritmo de decodificação para os códigos polares não-sistemáticos através de cancelamento sucessivo, SC (*Successive Cancellation*). Já na Subseção 4.3.2 apresentaremos o

Figura 19 – Exemplo de codificação sistemática.



Fonte: próprio autor.

decodificador por cancelamento sucessivo em lista, SCL (*Successive Cancellation List*) para códigos polares sistemáticos e em 4.3.2.1 será apresentado o uso dos CRCs atrelados a códigos polares sistemáticos.

4.3.1 Decodificação Não-Sistemática

Consideremos que $\mathbf{x}_1^N$ é a palavra-código de um codificador polar e que $\mathbf{y}_1^N$ é a mensagem recebida após a transmissão através de um canal W_N. A tarefa do decodificador por cancelamento sucessivo SC que será apresentado a seguir é determinar $\hat{\mathbf{u}}_1^N$, como a mensagem original $\mathbf{u}_1^N$, a partir de $(\mathbf{y}_1^N, \mathbf{u}_{A^c})$. Devido ao conhecimento prévio das linhas congeladas, podemos evitar erros de decodificação com mais facilidade nas mesmas, dessa forma, a verdadeira tarefa se encontra em determinar $\hat{\mathbf{u}}_A$ igual a $\mathbf{u}_A$. Para isso, teremos que:

$$\hat{\mathbf{u}}_i \triangleq \begin{cases} \mathbf{u}_i, \ se\ i \in A^c, \\ L_N^{(i)}(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1})\ se\ i \in A, \end{cases} \tag{4.50}$$

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

$$L_N^{(i)}(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}) = \begin{cases} 0, \ se\ \frac{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=0)}{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=1)} \geq 1. \\ 1, \ se\ \frac{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=0)}{W_N^i(\mathbf{y}_1^N, \hat{\mathbf{u}}_1^{i-1}|u_i=1)} < 1. \end{cases} \tag{4.51}$$

Podemos determinar LR com o uso das relações de recursão:

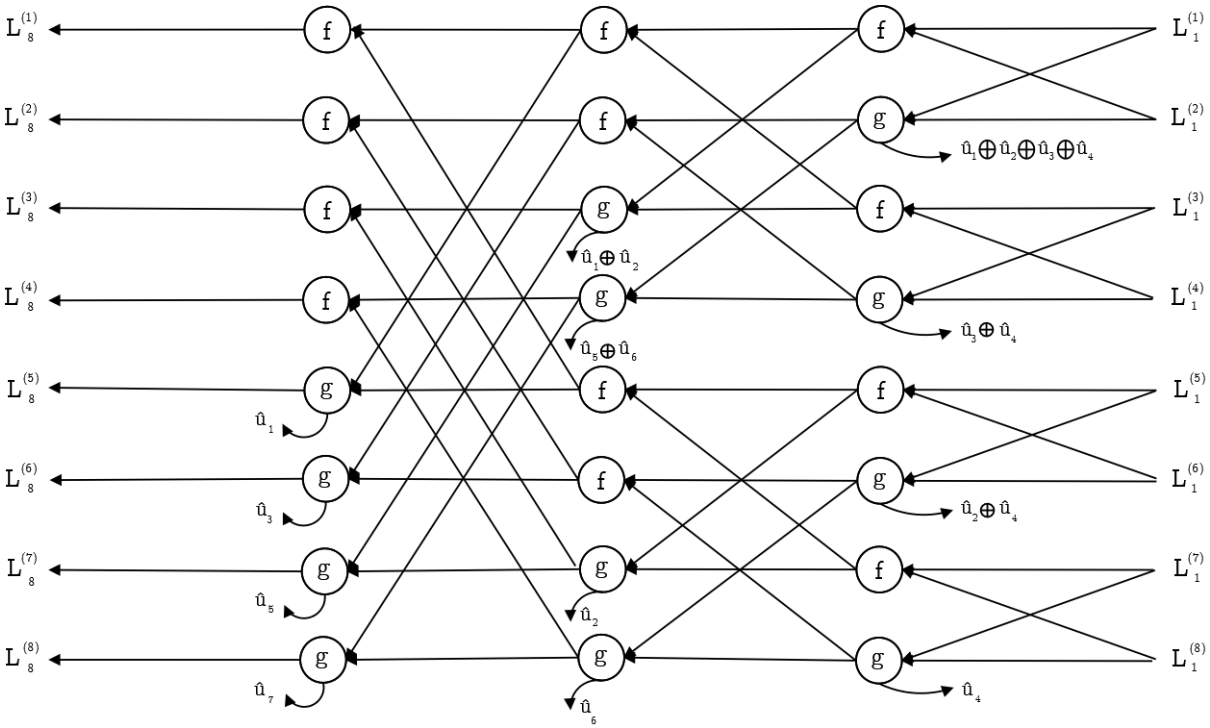
$$f(a,b) = \frac{(ab)+1}{a+b}, \tag{4.52}$$

$$g(a,b,\hat{u}_i) = a^{1-2\hat{u}_{i-1}}b. \tag{4.53}$$

Dessa forma, a partir das Equações 4.51, podemos construir uma rede de fluxo de razão de

verossimilhança, ou seja, um decodificador por cancelamento sucessivo, onde atualizamos LR ao longo das rotas, esse decodificador é apresentado na Figura 20. Arikan introduziu uma ordem de decodificação na qual a complexidade computacional provou-se ser $O(N\log(N))$. Esse tipo de codificação pode ser classificado como decodificação de *belief propagation* em tal gráfico de fator.

Figura 20 – Decodificador por cancelamento sucessivo para um bloco de tamanho $N = 8$.



Fonte: próprio autor.

Exemplo 4.3.1 *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 4.52, 4.53, 4.50 e 4.51 e decodificar a mensagem recebida através do método de cancelamento sucessivo. A Figura 21 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* $L_8^{(1)}$*:*

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

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* $L_8^{(3)}$*:*

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

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* $L_8^{(7)}$*:*

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

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* $L_8^{(2)}$*:*

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

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* $\hat{u}_7 = 1$*.*

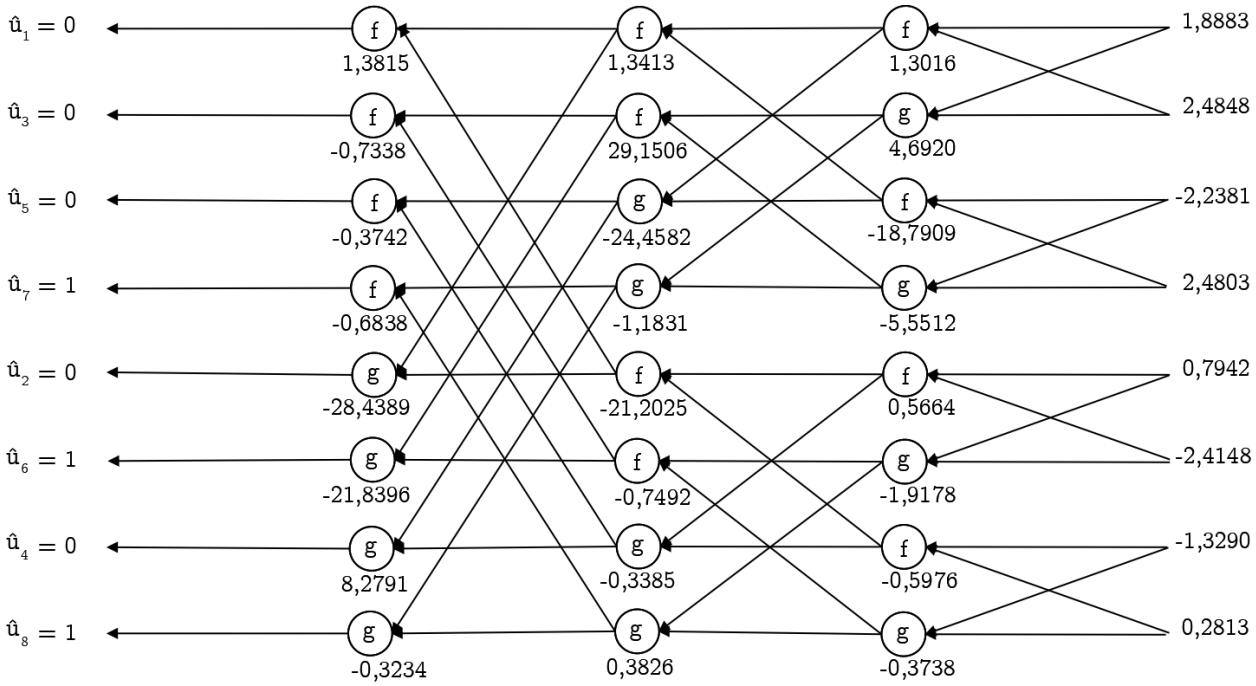
- ***Etapa 8****: determinamos o nó remanescente de terceiro estágio.*
  - *Cálculo de* $L_8^{(8)}$*:*

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

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* $\hat{\mathbf{u}} = (00010101)$*.*

Figura 21 – Exemplo de decodificação não-sistemática por cancelamento sucessivo.

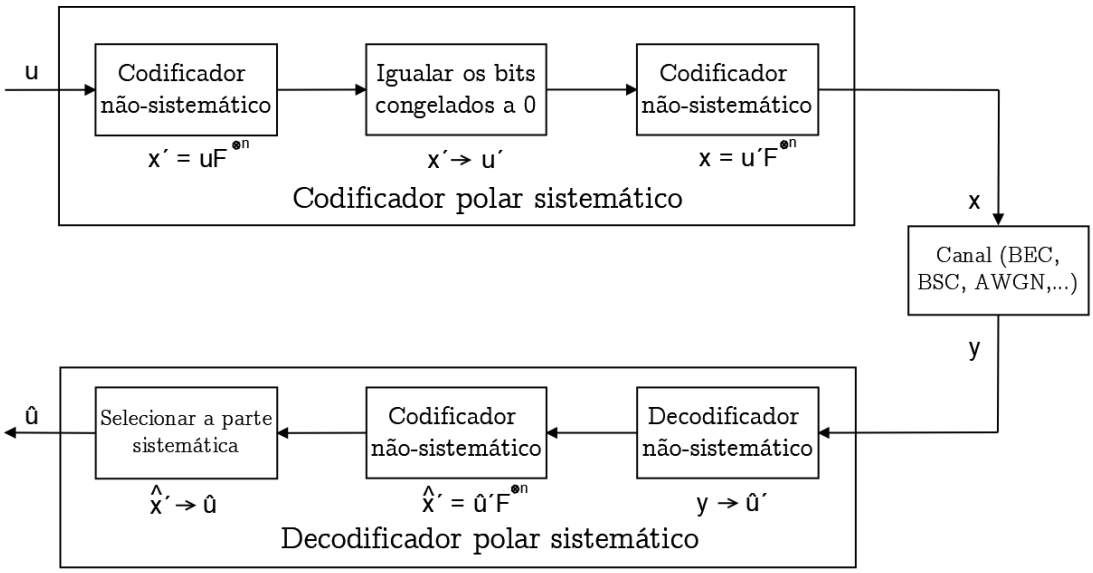


Fonte: próprio autor.

4.3.2 Decodificação Sistemática

Para o processo de decodificação dos códigos polares sistemáticos, o decodificador por cancelamento sucessivo em lista, SCL (*Successive Cancellation List*) é apresentado em [Arli 2020]. Este método consiste na utilização de um decodificador composto de três sub-estágios. O primeiro sub-estágio é um decodificador polar não-sistemático, o segundo sub-estágio é um codificador polar não-sistemático e o terceiro sub-estágio é um seletor, conforme podemos observar na Figura 22, que descreve o processo completo de codificação e decodificação de códigos polares sistemáticos.

Figura 22 – Codificação e decodificação de códigos polares sistemáticos.



Fonte: próprio autor.

Uma mensagem é codificada por um codificador sistemático, posteriormente, a palavra-código é transmitida através de um canal ruidoso, como um BEC, BSC ou AWGN, e então a saída desse canal, $\mathbf{y}$, é recepcionada pelo decodificador sistemático.

O decodificador polar sistemático tem como objetivo e função gerar uma mensagem estimada $\hat{\mathbf{u}}$ que corresponda à mensagem original $\mathbf{u}$. Entretanto, em vez de realizar uma decodificação polar sistemática direta, o decodificador realiza em seu primeiro estágio uma decodificação não-sistemática projetada para um código não-sistemático $\tilde{C}$. Esse código $\tilde{C}$ possui o mesmo conjunto de palavras-código que o código sistemático C, mas 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 por cancelamento sucessivo em lista, SCL. Diferentemente do decodificador SC, que mantém somente um caminho L (ou lista) para a decodificação, 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). À medida que 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 e Tse 2012, Tal e Vardy 2011].

Em seu segundo estágio, o decodificador possui um codificador não-sistemático para o código não-sistemático $\tilde{C}$, que codifica a mensagem estimada não-sistemática $\hat{\mathbf{u}}^{'}$ para obter uma palavra-código não-sistemática $\hat{\mathbf{x}}^{'}$. Em seu estágio final, o decodificador sistemático possui um seletor que emite como saída uma estimação $\hat{\mathbf{u}}$ da mensagem original, fazendo $\hat{\mathbf{u}} = \hat{\mathbf{x}}_B$. Vamos explicar melhor a aplicação deste decodificador através do exemplo a seguir; devido à complexidade dos cálculos aritméticos para demonstração do uso do decodificador por cancelamento sucessivo em lista para valores de L maiores que 1, demonstraremos os cálculos para L igual a 1, e nas simulações computacionais apresentaremos os resultados para L maiores que 1.

Exemplo 4.3.2 *Consideremos que a mensagem* $\mathbf{u} = [00010101]$ *foi codificada através de um codificador polar sistemático e que a palavra-código resultante é* $\mathbf{x} = [01010101]$. *Sabendo que essa palavra-código foi transmitida através de um canal AWGN no qual* $E_b/N_0 = 1\ dB$ *e que o vetor recebido foi*

$$\mathbf{y} = (1,2204; 1,4903; 2,1278; -2,5714; 0,3169; 0,2567; -0,7311; -0,2785).$$

Utilizando a estratégia de decodificação proposta na Figura 22, desejamos decodificar a mensagem recebida, de forma que a mensagem estimada $\hat{\mathbf{u}}$*, seja igual a mensagem original,* $\mathbf{u}$*. Iniciamos o processo de decodificação polar sistemática pelo primeiro estágio do decodificador, o qual se trata de um decodificador polar não-sistemático através da decodificação por SCL com L igual a 1.*

- *Etapa 1: determinamos os nós de primeiro estágio possíveis,* $L_2^{(1)}$*,* $L_2^{(3)}$*,* $L_2^{(5)}$*,* $L_2^{(7)}$*, posteriormente, os nós de segundo estágio possíveis,* $L_4^{(1)}$*,* $L_4^{(5)}$*, e finalmente, um dos nós de terceiro estágio,* $L_8^{(1)}$*.*

$$L_2^{(1)} \to f\left(L_1^{(1)}, L_1^{(2)}\right) = f(1,2204; 1,4903) = \frac{(1,2204 \cdot 1,4903) + 1}{1,2204 + 1,4903} = 1,0399.$$

$$L_2^{(3)} \to f\left(L_1^{(3)}, L_1^{(4)}\right) = f(2,1278; -2,5714) = \frac{(2,1278 \cdot -2,5714) + 1}{2,1278 + (-2,5714)} = 10,0799.$$

$$L_2^{(5)} \to f\left(L_1^{(5)}, L_1^{(6)}\right) = f(0,3169; 0,2567) = \frac{(0,3169 \cdot 0,2567) + 1}{(0,3169 + 0,2567)} = 1,8852.$$

$$L_2^{(7)} \to f\left(L_1^{(7)}, L_1^{(8)}\right) = f(-0,7311+(-0,2785)) = \frac{(-0,7311 \cdot (-0,2785)) + 1}{-0,7311 + (-0,2785)} = -1,1922.$$

$$L_4^{(1)} \to f\left(L_2^{(1)}, L_2^{(3)}\right) = f(1,0399; 10,0799)$$
$$= \frac{(1,0399 \cdot 10,0799) + 1}{1,0399 + 10,0799} = 1,0326.$$

$$L_4^{(5)} \to f\left(L_2^{(5)}, L_2^{(7)}\right) = f(1,8852; -1,1922)$$
$$= \frac{(1,8852 \cdot (-1,1922)) + 1}{(1,8852 + (-1,1922))} = -1,8000.$$

$$L_8^{(1)} \to f\left(L_4^{(1)}, L_4^{(5)}\right) = f(1,0326; -1,8000)$$
$$= \frac{(1,0326 \cdot (-1,8000)) + 1}{1,0326(-1,8000)} = 1,1118.$$

Como $L_8^{(1)} = 1,1118 > 1$*, logo* $\hat{\mathbf{u}}_\mathbf{1} = 0$*.*

- *Etapa 2: com informações obtidas na etapa 1, podemos determinar outro nó de terceiro estágio,* $L_8^{(5)}$*.*

$$L_8^{(5)} \to g\left(L_4^{(1)}, L_4^{(5)}, \hat{u_1}\right) = 1,0326^{(1-2(0))}(-1,8000) = -1,8586.$$

Como $L_8^{(5)} = -1,8586 > 1$*, logo* $\hat{\mathbf{u}}_\mathbf{5} = 0$*.*

- *Etapa 3: com informações obtidas na Etapa 1, podemos determinar outro nós de segundo estágio,* $L_4^{(3)}$*,* $L_4^{(7)}$*, e de terceiro estágio,* $L_8^{(3)}$*.*

$$L_4^{(3)} \to g\left(L_2^{(1)}, L_2^{(3)}, \hat{u_2}\right) = 1,0399^{(1-2(0))}(10,0799) = 10,4817.$$

$$L_4^{(7)} \to g\left(L_2^{(5)}, L_2^{(7)}, \hat{u_2}\right) = 1,8852^{(1-2(0))}(-1,1922) = -2,2475.$$

$$L_8^{(3)} \to f\left(L_4^{(3)}, L_4^{(7)}\right) = f(10,4817; -2,2475)$$
$$= \frac{(10,4817 \cdot (-2,2475)) + 1}{(10,4817 + (-2,2475))} = -2,7395.$$

Como $L_8^{(3)} = 2,7395 > 1$*, logo* $\hat{\mathbf{u}}_\mathbf{3} = 0$.

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

$$\begin{aligned} L_8^{(7)} \rightarrow g\left(L_4^{(3)}, L_4^{(7)}, \hat{u}_3\right) &= g(10,4817; -2,2475; 0) \\ &= 10,4817^{(1-2(0))}(-2,2475) = -23,5573. \end{aligned}$$

Como $L_8^{(7)} = -23,5573 < 1$*, logo* $\hat{\mathbf{u}}_\mathbf{4} = 1$.

- *Etapa 5: utilizando-se das informações obtidas até o momento, podemos calcular os nós de primeiro estágio,* $L_2^{(2)}$*,* $L_2^{(4)}$*,* $L_2^{(6)}$*,* $L_2^{(8)}$*, outros nós de segundo estágio,* $L_4^{(2)}$ *e* $L_4^{(6)}$ *e do nó de terceiro estágio,* $L_8^{(2)}$.

$$\begin{aligned} L_2^{(2)} \rightarrow g\left(L_1^{(1)}, L_1^{(2)}, \hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3 \oplus \hat{u}_4\right) \\ = g(1,2204; 1,4903; 0 \oplus 0 \oplus 0 \oplus 0) = g(1,2204; 1,4903; 0) \\ = 1,2204^{(1-2(0))}(1,4903) = 1,8188. \end{aligned}$$

$$\begin{aligned} L_2^{(4)} \rightarrow g\left(L_1^{(3)}, L_1^{(4)}, \hat{u}_3 \oplus \hat{u}_4\right) \\ = g((2,1278; -2,5714; 0 \oplus 0) = g(2,1278; -2,5714; 0) \\ = 2,1278^{(1-2(0))}(-2,5714) = -5,4714. \end{aligned}$$

$$\begin{aligned} L_2^{(6)} \rightarrow g\left(L_1^{(5)}, L_1^{(6)}, \hat{u}_2 \oplus \hat{u}_4\right) \\ = g(0,3169; 0,2567; 0 \oplus 0) = g(0,3169; 0,2567; 0) \\ = 0,3169^{(1-2(0))}(0,2567) = 0,0813. \end{aligned}$$

$$\begin{aligned} L_2^{(8)} \rightarrow g\left(L_1^{(7)}, L_1^{(8)}, \hat{u}_4\right) &= g(-0,7311; -0,2785; 0) \\ &= -0,7311^{(1-2(0))}(-0,2785) = 0,2036. \end{aligned}$$

$$\begin{aligned} L_4^{(2)} \rightarrow f\left(L_2^{(2)}, L_2^{(4)}\right) &= f(1,8188; -5,4714) \\ &= \frac{(1,8188 \cdot -5,4714) + 1}{(1,8188 + (-5,4714))} = 2,4506. \end{aligned}$$

$$L_4^{(6)} \to f\left(L_2^{(6)}, L_2^{(8)}\right) = f(0,0813; 0,2036)$$
$$= \frac{(0,0813 \cdot 0,2036) + 1}{0,0813 + 0,2036} = 3,5674.$$

$$L_8^{(2)} \to f\left(L_4^{(2)}, L_4^{(6)}\right) = f(2,4506; 3,5674)$$
$$= \frac{(2,4506 \cdot 3,5674) + 1}{2,4506 + 3,5674} = 1,6189.$$

Como $L_8^{(2)} = 1,6189 > 1$*, logo* $\hat{\mathbf{u}}_{\mathbf{5}} = 0$*.*

- *Etapa 6: as informações obtidas na etapa 5 nos possibilitam a determinação do nó* $L_8^{(6)}$*, de terceiro estágio.*

$$L_8^{(6)} \to g\left(L_4^{(2)}, L_4^{(6)}, \hat{u}_5\right) = g(2,4506; 3,5674; 0)$$
$$= 2,4506^{(1-2(0))}(3,5674) = 8,7423.$$

Como $L_8^{(6)} = 8,7423 > 1$*, logo* $\hat{\mathbf{u}}_{\mathbf{6}} = 0$*.*

- *Etapa 7: podemos determinar, com os resultados obtidos na etapas anteriores, os nós remanescentes de segundo estágio,* $L_4^{(4)}$ *e* $L_4^{(8)}$*, e do nó* $L_8^{(4)}$*, de terceiro estágio.*

$$L_4^{(4)} \to g\left(L_2^{(2)}, L_2^{(4)}, \hat{u}_5 \oplus \hat{u}_6\right) = g(1,8188; -5,4714; 0 \oplus 1)$$
$$= g(1,8188; -5,4714; 1) = 1,8188^{(1-2(1))}(-5,4714) = -3,0083.$$

$$L_4^{(8)} \to g\left(L_2^{(6)}, L_2^{(8)}, \hat{u}_6\right) = g(0,0813; 0,2036; 1)$$
$$= 0,0813^{(1-2(1))}(0,2036) = 2,5030.$$

$$L_8^{(4)} \to f\left(L_4^{(4)}, L_4^{(8)}\right) = f(-3,0083; 2,5030) = \frac{(-3,0083 \cdot 2,5030) + 1}{-3,0083 + 2,5030} = 12,9208.$$

Como $L_8^{(4)} = 12,9208 > 1$*, logo* $\hat{\mathbf{u}}_{\mathbf{7}} = 0$*.*

- *Etapa 8: os resultados obtidos na etapa anterior nos permitem calcular o último nó de terceiro estágio remanescente,* $L_8^{(8)}$*.*

$$L_8^{(8)} \rightarrow g\left(L_4^{(4)}, L_4^{(8)}, \hat{u}_7\right) = g(-3,0083; 2,5030; 1)$$
$$= -3,0083^{(1-2(1))}(2,5030) = -0,8320.$$

Como $L_8^{(8)} = -0,8320 < 1$*, logo* $\hat{\mathbf{u}}_\mathbf{8} = 1$.

Assim, determinamos que a mensagem estimada não-sistemática é $\hat{\mathbf{u}}^{'} = [00000011]$*. No segundo estágio do decodificador, temos um codificador polar não-sistemático, o qual realiza as seguintes operações, baseado em (4.34):*

$$\hat{\mathbf{x}}^{'} = \hat{\mathbf{u}}^{'} F^{\otimes 3}$$
$$= [00000011] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$
$$= [01010101].$$

No terceiro estágio do decodificador, o seletor seleciona a parte sistemática de $\hat{\mathbf{x}}^{'}$*, que corresponde às posições {4,6,7,8}, a fim de se obter a estimação da mensagem original, portanto* $\hat{\mathbf{u}} = [00010101]$*. Se compararmos a mensagem estimada à mensagem original,* $\mathbf{u}$*, verificamos que não houve erro na transmissão.*

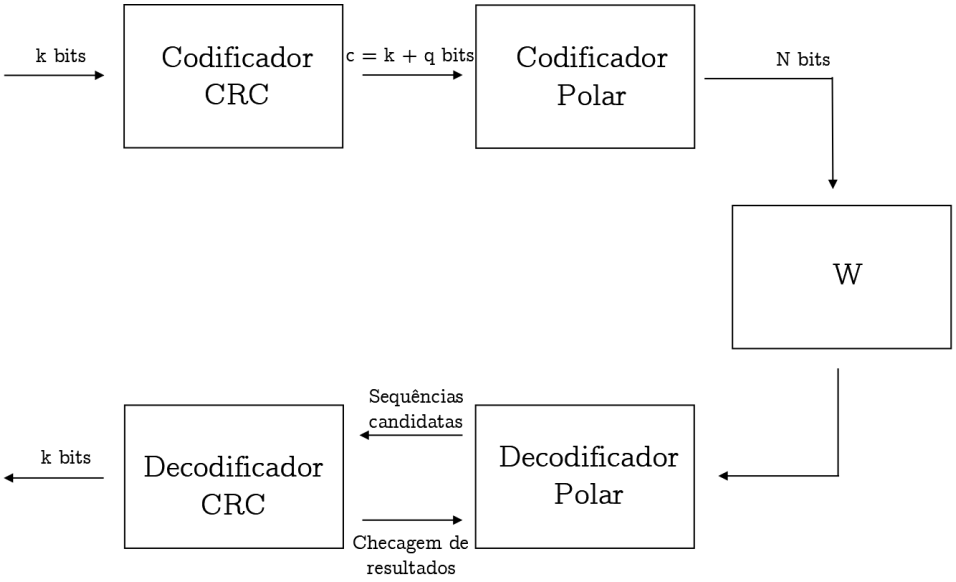
4.3.2.1 Uso de Códigos CRC na Decodificação de Códigos Polares Sistemáticos

Dentre as diversas aplicações dos códigos CRC, podemos citar seu uso atrelado a códigos polares com o objetivo de aumentar o desempenho do sistema. O decodificador por cancelamento sucessivo em lista, SCL, utilizados para decodificar códigos polares sistemáticos tem como uma de suas características que a palavra-código correta a ser decodificada está na lista de decodificação, porém não é necessariamente sempre a palavra-código mais provável. Dessa forma, uma das maneiras de se aumentar a probabilidade de que a palavra-código correta seja a escolhida da lista é através da combinação de um código CRC ao decodificador SCL.

Neste caso, o sistema de transmissão, apresentado na Figura 23, é composto por um codificador CRC, onde os k bits de informação são codificados em N bits, através da adição de $N - k$ bits de paridade (redundância). Estes bits redundantes serão adicionados à informação para permitir a identificação correta da mensagem, posteriormente no decodificador. Essa sequência é então codificada pelo codificador polar, para então ser modulada e transmitida através de um canal. Ressalta-se que,

devido à adição dos bits de redundância, os códigos polares sistemáticos com CRC passam a ser descritos pelos parâmetros $(N, k+q)$, em $q = N - k$ representam os q-CRC bits redundantes do código CRC. No receptor, a probabilidade é calculada a partir do sinal recebido e então é transmitida ao decodificador em lista com CRC, onde os caminhos L das palavras-código possíveis do decodificador em lista são testados pelo detector de CRC, a partir do caminho com maior probabilidade. Quando a palavra-código possível é determinada como correta pelo decodificador CRC, os k bits de informação correspondentes são considerados como os bits transmitidos e os demais bits são zerados.

Figura 23 – Concatenação de códigos polares e códigos CRC.



Fonte: próprio autor.

5 RESULTADOS E DISCUSSÃO

Detalharemos neste capítulo os resultados da implementação computacional e as simulações que foram realizadas através do uso de uma rotina de programação em linguagem *Python*. Com o intuito de obter-se resultados precisos, as simulações foram repetidas uma grande quantidade de vezes, conforme descreve o método de Monte Carlo, que é fundamentado na Lei dos Grandes Números e no Teorema Central do Limite [Jacoboni e Lugli 2011].

A partir de distribuições de probabilidade, as simulações de Monte Carlo podem efetuar amostragens aleatórias com a repetição do processo um elevado número de vezes, assim, é possível, através de métodos estatísticos, como a média, analisar a amostra e obter uma aproximação numérica do resultado [ROSS 2011].

Podemos estimar um evento de interesse A através de 5.1, sendo que a quantidade de simulações é $N^{'}$ e a quantidade de ocorrência de um determinado evento de interesse A é $N_A^{'}$.

$$P_r(A) = \lim_{N' \to \infty} \frac{N_A^{'}}{N^{'}}. \tag{5.1}$$

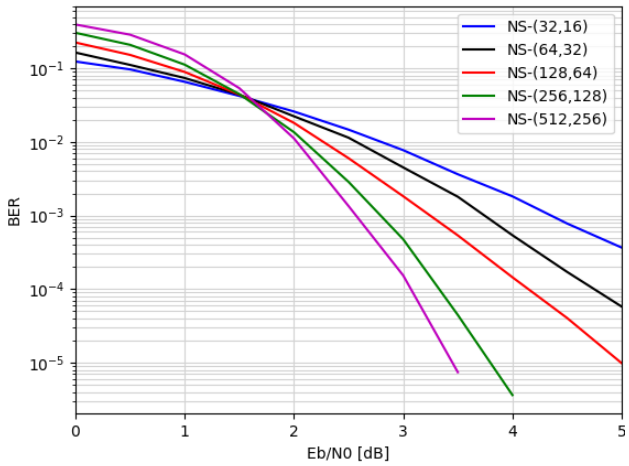
A aplicação deste método para as simulações realizadas neste trabalho se deu com a repetição dos processos 10^4 vezes. Implementou-se um algoritmo computacional em um dispositivo com processador Intel(R) Core(TM) i7-4710HQ CPU @2,50 GHz e 16,0 GB de RAM, que simula um sistema de transmissão no qual uma mensagem u de k bits aleatórios é gerada, e que posteriormente é codificada por um codificador polar, com taxa $R = k/N = 0,5$, considerando-se um intervalo de E_b/N_0 de 0 a 5 dB, e potência de ruído de 2 dBm. A palavra-código $\mathbf{x}$ é então modulada através de um modulador BPSK, e transmitida através de um canal AWGN, que introduz ruído. Recebe-se a mensagem, que é demodulada e decodificada em $\hat{\mathbf{u}}$ por um decodificador, sendo posteriormente comparada à mensagem $\mathbf{u}$ transmitida. Caso haja erros, iremos contabilizá-los em relação à BER, que apresenta a relação da quantidade de bits que tiveram erros pela quantidade de bits transmitidos, e na FER, que apresenta a relação da quantidade de frames com erros, pela quantidade de frames transmitidos. A FER é importante na análise do desempenho do sistema em relação a erros de blocos, ou erros em rajada. Ambos os parâmetros nos auxiliam a verificar o desempenho dos sistemas, e nos permitem avaliá-los em relação à sua confiabilidade.

Alguns requisitos devem ser seguidos nos cenários para os sistemas 5G, como BER menor ou igual a 10^{-6}, valores de latência menores que 10 ms, levando em consideração que a mesma depende das especificações dos dispositivos utilizados nesses sistemas, e a transmissão de pacotes curtos de até 512 bits [Popovski 2014, Lema et al. 2017, Popovski et al. 2019]. A seguir, iremos apresentar os resultados obtidos considerando a codificação polar não-sistemática e sistemática. Para a decodificação, iremos utilizar o decodificador por cancelamento sucessivo SC no caso não-sistemático e o decodificador com cancelamento sucessivo em lista SCL para o sistemático. No caso sistemático, ainda iremos adicionar um código CRC ao decodificador SCL de modo a verificar as alterações no desempenho do mesmo. Como estamos interessados nos cenários 5G, consideramos palavras de até 512 bits. Observamos que a latência não foi analisada neste trabalho, mas para o caso não-sistemático resultados para latência

foram apresentados em [TERÇAS 2019].

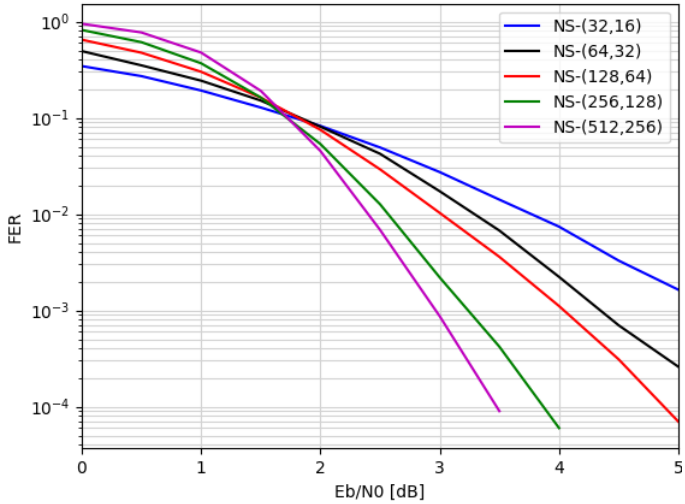
Para a codificação polar não-sistemática, considerou-se os parâmetros mencionados anteriormente e foram obtidas as curvas de BER em relação à SNR na Figura 24 e as curvas de FER em relação à SNR na Figura 25. Destaca-se que quando $E_b/N_0 \simeq 1,75\ dB$, todos os comprimentos de palavras possuem a mesma BER e FER. Observou-se que anteriormente a esse ponto, as curvas referentes às palavras-código com comprimento maior apresentaram um desempenho inferior, entretanto, após esse ponto, o comportamento é o oposto, ou seja, quanto maior o comprimento da palavra-código, melhor é seu desempenho, sendo que as curvas tendem a valores de BER e FER de 10^{-6} mais rapidamente, o que é desejável em cenários 5G.

Figura 24 – Curvas de BER em relação à SNR para códigos polares não-sistemáticos com $(N, k) = (32, 16; 64, 32; 128, 64; 256, 128; 512, 256)$.



Fonte: próprio autor.

Figura 25 – Curvas de FER em relação à SNR para códigos polares não-sistemáticos com $(N, k) = (32, 16; 64, 32; 128, 64; 256, 128; 512, 256)$.

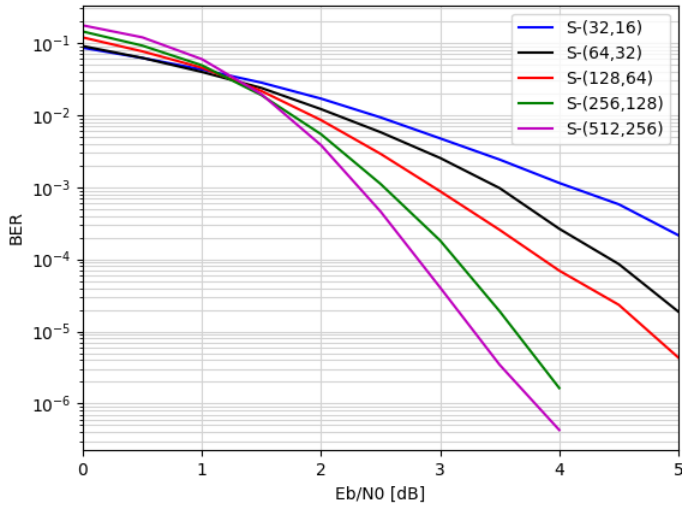


Fonte: próprio autor.

Inicialmente para os códigos polares sistemáticos, foram obtidas as curvas de BER em relação à SNR na Figura 26 e as curvas de FER em relação à SNR na Figura 27, com os mesmos parâmetros

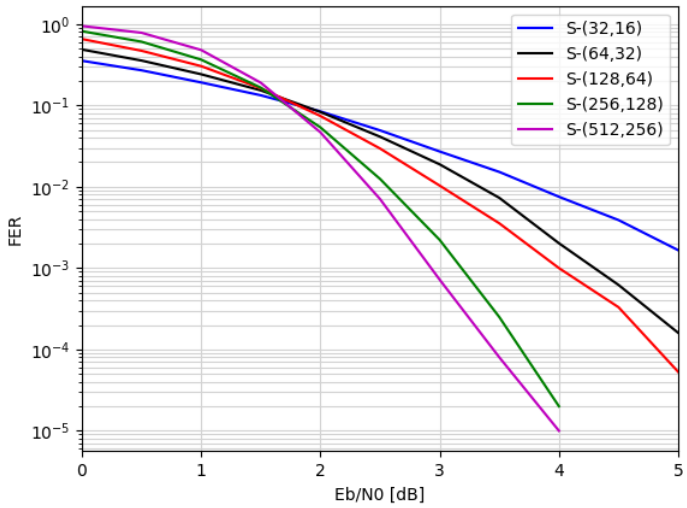
descritos anteriormente. Foi considerado o decodificador SCL para $L = 1$, que é o mesmo decodificador SC porém adicionando um seletor para identificar as posições sistemáticas.

Figura 26 – Curvas de BER em relação à SNR para códigos polares sistemáticos com $(N, k) = (32, 16; 64, 32; 128, 64; 256, 128; 512, 256)$.



Fonte: próprio autor.

Figura 27 – Curvas de FER em relação à SNR para códigos polares sistemáticos com $(N, k) = (32, 16; 64, 32; 128, 64; 256, 128; 512, 256)$.



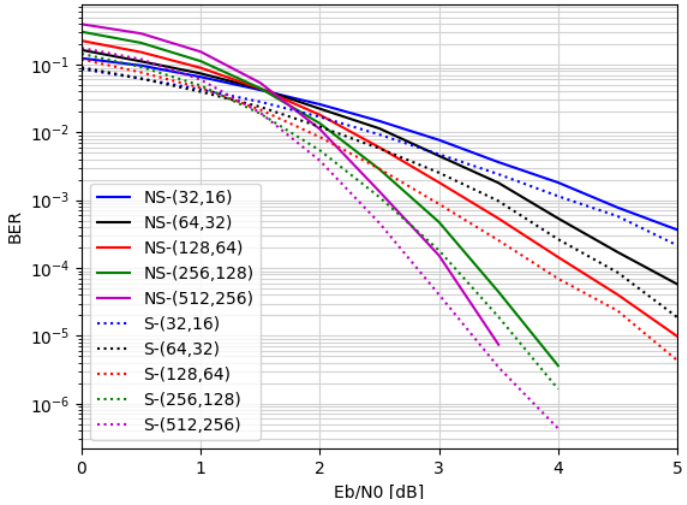
Fonte: próprio autor.

Quando comparamos o desempenho dos códigos polares não-sistemáticos com os códigos polares sistemáticos, podemos verificar na Figura 28 que os sistemáticos apresentam desempenho superior para todos os tamanhos de palavras, e que tendem a valores de BER menores que 10^{-6} mais rapidamente que os não-sistemáticos. Quando analisamos a FER na figura Figura 29, observamos que o desempenho é semelhante para as duas estratégias de codificação.

Agora, iremos adicionar um código CRC aos códigos polares sistemáticos. Adicionou-se um 8-CRC com 8 bits redundantes em um decodificador SCL. Esta adição altera os parâmetros desses códigos, inicialmente descritos por (N, k), passando a ser $(N, k + 8)$. Com o intuito de avaliar o

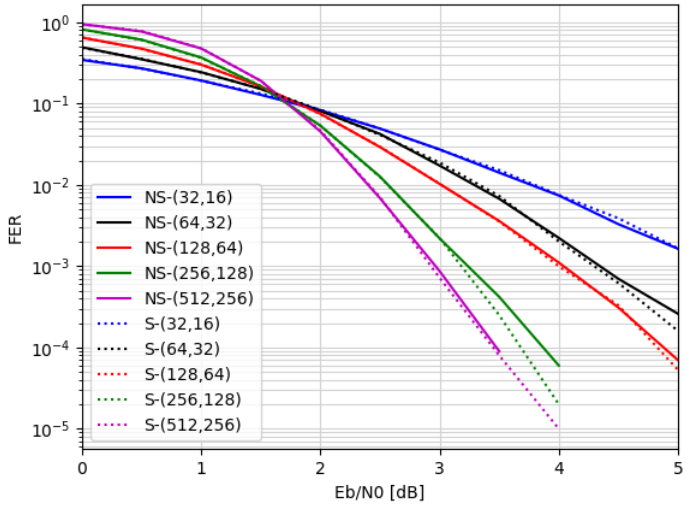
comportamento dos códigos polares sistemáticos com CRC para diferentes tamanhos de lista, iniciamos as simulações fixando $N = 16$ e variando o tamanho da lista em $L = 2^n$, com $n = 0, \ldots, 5$. As curvas para BER e FER são apresentadas na Figura 30 e na Figura 31, respectivamente. O comportamento observado nas curvas é que para valores maiores de L, a convergência para valores menores de BER e FER ocorre mais rapidamente, ou seja, o uso de uma lista maior, gera um desempenho melhor.

Figura 28 – Curvas de BER em relação à SNR para códigos polares não-sistemáticos e sistemáticos com $(N, k) = (32, 16; 64, 32; 128, 64; 256, 125; 512, 256)$.



Fonte: próprio autor.

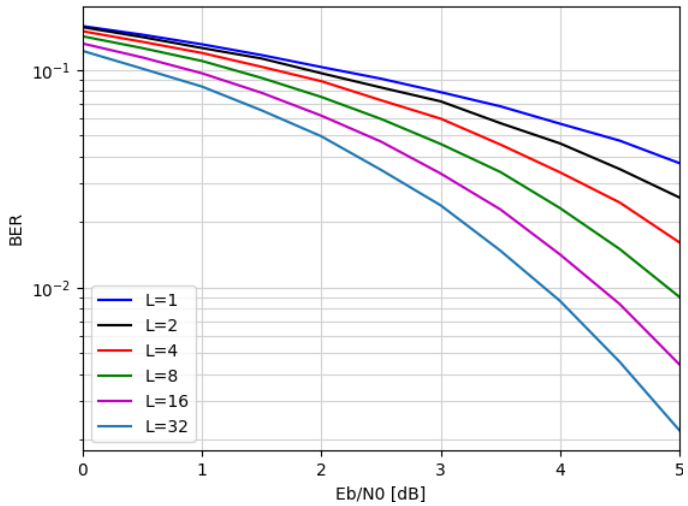
Figura 29 – Curvas de FER em relação à SNR para códigos polares não-sistemáticos e sistemáticos com $(N, k) = (32, 16; 64, 32; 128, 64; 256, 125; 512, 256)$.



Fonte: próprio autor.

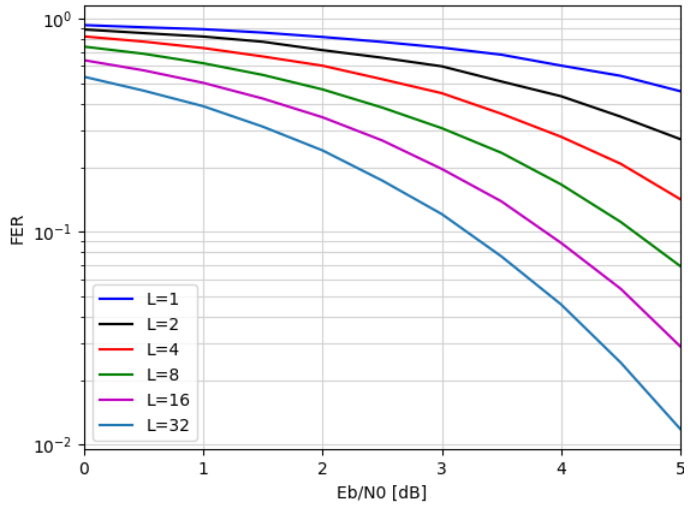
Como vimos, em cenário de quinta geração 5G, é desejável se utilizar pacotes curtos de até 512 bits. Porém, para o dispositivo utilizado nas simulações deste trabalho não foi possível considerar tamanhos de listas grandes a medida que o comprimento das palavras também crescia. Dessa forma, simulou-se um cenário no qual fixou-se o comprimento da lista do decodificador SCL em $L = 1$, o que seria equivalente ao decodificador por cancelamento sucessivo SC, e comparou com os resultados

Figura 30 – Curvas de BER em relação à SNR para códigos polares sistemáticos com parâmetros $(16, 16)$ para diferentes tamanhos de lista, L.



Fonte: próprio autor.

Figura 31 – Curvas de FER em relação à SNR para códigos polares sistemáticos com parâmetros $(16, 16)$ e diferentes tamanhos de lista, L.



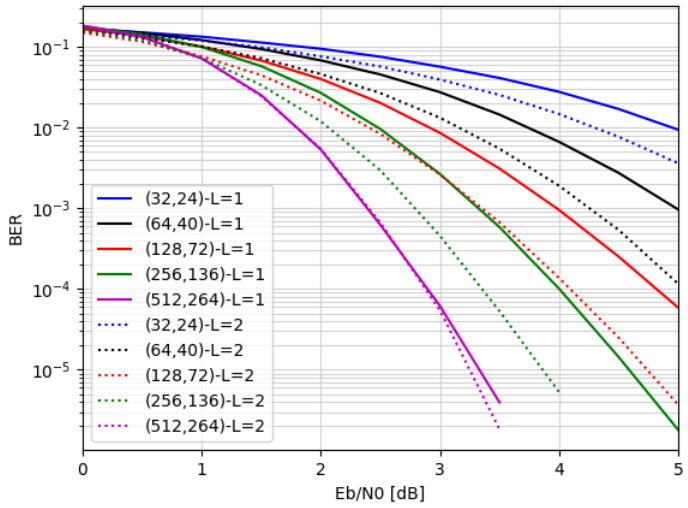
Fonte: próprio autor.

para $L = 2$, variando-se o tamanho das mensagens em $N = 2^n$ bits, com $n = 5, \ldots, 9$. Na Figura 32 validou-se que conforme aumenta-se o comprimento das palavras-código, as curvas tendem a valores de BER menores que 10^{-6} mais rapidamente. Do mesmo modo, aumentando o tamanho da lista para $L = 2$ em um mesmo comprimento de palavra-código, a BER também convergirá mais depressa para 10^{-6}. Um comportamento similar é observado para as curvas de FER apresentada na Figura 33.

Por fim, com o objetivo de comparar o desempenho dos códigos polares não-sistemáticos em relação aos códigos polares sistemáticos atrelados a um CRC e com tamanho de lista $L = 2$, simulou-se os mesmos fazendo-se o uso de parâmetro semelhantes. A Figura 34 apresenta as curvas de BER em relação à SNR para códigos polares não-sistemáticos, representadas pelas linhas tracejadas e pontilhadas, e para códigos polares sistemáticos, representadas pelas linhas contínuas. Verificou-se que, para as mensagens com parâmetros (128,72) e (512,264), os códigos sistemáticos convergem mais

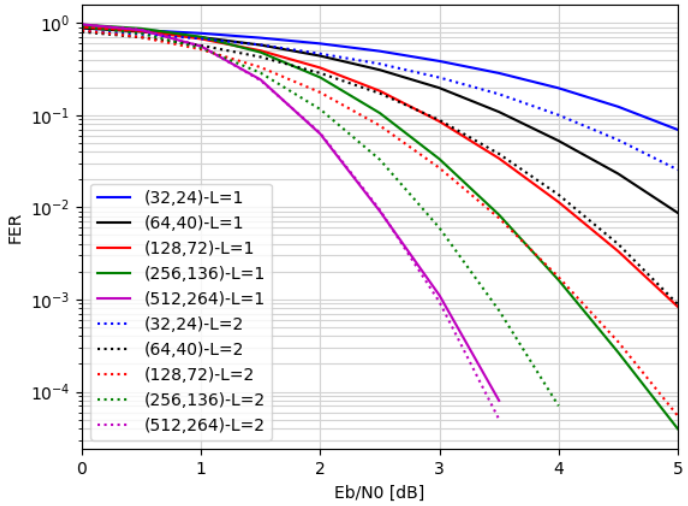
rapidamente para valores de BER menores que 10^{-6}, entretanto o mesmo não ocorreu para os demais comprimentos de palavras. Já quando a FER é analisada na Figura 35, com exceção da palavra com menor comprimento, a com parâmetros (32,24), todas as demais mensagens codificadas sistematicamente com o uso do CRC, apresentam desempenho superior que os das mensagens codificadas de forma não-sistemática, ou seja, as curvas tendem a valores de FER menores que 10^{-6} mais rapidamente que as referentes à codificação não-sistemática. Desta forma, validou-se que para alguns casos específicos, o uso do CRC atrelado aos códigos polares sistemáticos melhorou o desempenho em relação à taxa de erro de bit, entretanto, ao analisar um conjunto de bits, ou seja, analisando-se a taxa de erro de frame, verificou-se que, o uso do CRC melhorou consideravelmente o desempenho do sistema, tornando-o superior em todos os casos, com exceção de $N = 32$.

Figura 32 – Curvas de BER em relação à SNR para códigos polares sistemáticos com CRC e parâmetros $(N, k + 8) = (32, 24; 64, 40; 128, 72; 256, 136; 512, 264)$, $L = (1; 2)$.



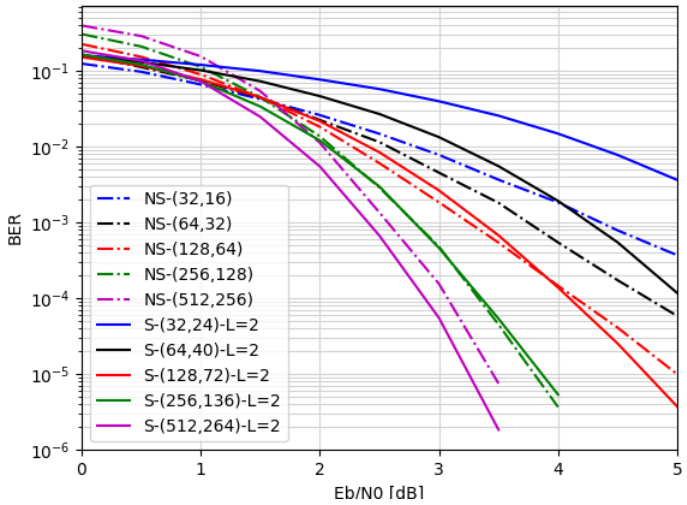
Fonte: próprio autor.

Figura 33 – Curvas de FER em relação à SNR para códigos polares sistemáticos com CRC e parâmetros $(N, k + 8) = (32, 24; 64, 40; 128, 72; 256, 136; 512, 264)$, $L = (1; 2)$.



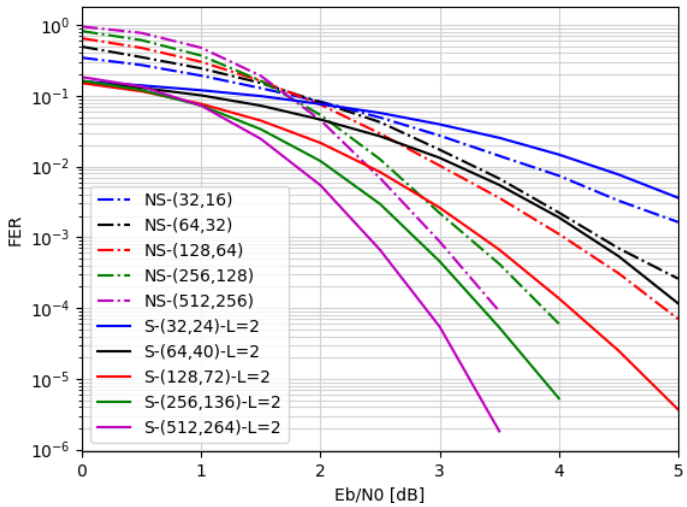
Fonte: próprio autor.

Figura 34 – Curvas de BER em relação à SNR para códigos polares não-sistemáticos e sistemáticos com CRC.



Fonte: próprio autor.

Figura 35 – Curvas de FER em relação à SNR para códigos polares não-sistemáticos e sistemáticos com CRC.



Fonte: próprio autor.

6 CONCLUSÕES

Neste trabalho, realizamos um estudo de uma técnica de codificação de canal designada como códigos polares, que foram apresentados inicialmente em sua forma não-sistemática, e posteriormente em sua forma sistemática. Esse tipo de codificação é uma das candidatas a serem utilizadas nas aplicações dos sistemas 5G.

Para isso apresentamos inicialmente uma revisão de teoria da informação e codificação. Após isso, os códigos cíclicos, que são códigos de bloco lineares nos quais as palavras-códigos são deslocadas ciclicamente foram apresentados a fim de introduzir os códigos CRC, que são códigos em que os bits de redundância são utilizados em algoritmos de decodificação para ajudar na melhorar a eficiência dos sistemas de comunicação.

Para a codificação, os códigos polares fazem o uso de uma técnica conhecida como polarização de canal, a qual classifica os canais como bons ou ruins de acordo com a sua capacidade. Os canais ruins são congelados, ou seja, não são utilizados para transmitir informação. A utilização dos canais bons aumenta o desempenho do sistema. Os códigos polares sistemáticos, diferentemente dos códigos polares não-sistemáticos, tem como característica principal apresentar, de maneira explícita na palavra-código, os bits não congelados, o que aumenta o desempenho do sistema.

A técnica de decodificação por cancelamento sucessivo foi apresentada para os códigos polares não-sistemáticos, que é um tipo de decodificação por máxima verossimilhança e que realiza a decodificação através do caminho reverso, ou seja, quando um bit é inferido, o próximo bit dependerá do anterior. Para a decodificação polar sistemática, apresentou-se a técnica que é composta por três sub-estágios, sendo que o primeiro é um decodificador polar não-sistemático por cancelamento sucessivo em lista, SCL, o segundo é um codificador polar não-sistemático e o terceiro é um seletor.

Para observar o comportamento dos códigos polares não-sistemáticos e sistemáticos, com e sem CRC, implementou-se uma rotina de programação em linguagem *Python*, da qual foi possível obter curvas de taxa de erro de bit e taxa de erro de frame em relação à razão sinal ruído. Para ambos os tipos de código verificou-se que o aumento de N leva à diminuição da BER, de forma que foi preciso uma SNR menor, ou seja, menos energia. Para os códigos sistemáticos, também verificou-se que o aumento do tamanho da lista L, do decodificador por cancelamento sucessivo em lista leva à uma diminuição ainda mais rápida.

Os dois tipos de codificação foram testados no cenário com pacotes curtos de até 512 bits, para os sistemas 5G, averiguando-se que ambos têm capacidade para atingir valores de BER e FER menores que 10^{-6} num canal AWGN. Analisando-se as curvas de BER, quando comparamos os códigos não-sistemáticos e os sistemáticos sem CRC, verificou-se explicitamente um melhor desempenho dos códigos sistemáticos em relação à BER. Também verificou-se que a adição da redundância de um código CRC às palavras-código sistemáticas levou a um desempenho superior para os casos em que $N = 128$ e $N = 512$, já ao analisar as curvas de FER, com exceção de $N = 32$, a adição do CRC levou a um desempenho superior para todos os demais tamanhos.

Sugere-se, para trabalhos futuros, o estudo do comportamento e a análise do desempenho dos

códigos polares sistemáticos atrelados ao uso de diferentes comprimentos de CRC, a utilização de outros tipos de decodificadores e também outros parâmetros desejáveis em cenários 5G, como a latência no caso da codificação sistemática.

REFERÊNCIAS

5GPPP. *View on 5G Architecture*. Segunda edição. [S.l.]: 5GPPP Architecture Working Group, 2017.

ALWIS, C. D. et al. Survey on 6g frontiers: Trends, applications, requirements, technologies and future research. **IEEE Open Journal of the Communications Society**, v. 2, p. 836–886, 2021.

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.

ARIKAN, E. Systematic polar coding. **IEEE Communications Letters**, v. 15, n. 8, p. 860–862, 2011.

ARLI, A. Ç. ***Polar Code Decoding with Soft Decisions Algorithms***. Tese (Doutorado) — The Graduate School of Natural and Applied Sciences of Çankaya University, January 2020. Disponível em: <https://acikbilim.yok.gov.tr/handle/20.500.12812/77329>.

BAICHEVA, T.; KAZAKOV, P. **CRC selection for decoding of CRC-polar concatenated codes**. 2019.

BERROU, C.; GLAVIEUX, A.; THITIMAJSHIMA, P. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In: **Proceedings of ICC '93 - IEEE International Conference on Communications**. [S.l.: s.n.], 1993. v. 2, p. 1064–1070 vol.2.

BIOGLIO, V.; CONDO, C.; LAND, I. Design of polar codes in 5g new radio. **IEEE Communications Surveys & Tutorials**, v. 23, n. 1, p. 29–40, 2021.

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

ERAZO, A. H. R. **Early Detection Using CRC Precoding and Polar Codes for Low Latency Communications**. Tese (Doutorado) — Espace ÉTS Mémoires et Thèses, 2017. Disponível em: <https://espace.etsmtl.ca/id/eprint/1905>.

FOROUZAN, B. A. **Data Communications and Networking**. [S.l.: s.n.], 2008. ISBN 978-0-07-296775-3.

GSA. **5G - market snapshot June 2022**. Global mobile Suppliers Association, 2022. Disponível em: <https://gsacom.com/paper/5g-market-snapshot-june-2022/>.

HAYKIN, S. ***Communications Systems***. Quarta edição. [S.l.]: John Wiley & Sons, Inc., 2001.

JACOBONI, C.; LUGLI, P. ***The Monte Carlo Method for Semiconductor Device Simulation***. [S.l.]: Springer Vienna, 2011. (*Computational Microelectronics*). ISBN 9783709174531.

LEMA, M. A. et al. **Business Case and Technology Analysis for 5G Low Latency Applications**. arXiv, 2017. Disponível em: <https://arxiv.org/abs/1703.09434>.

LI, B.; SHEN, H.; TSE, D. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check. **IEEE Communications Letters**, v. 16, n. 12, p. 2044–2047, 2012.

LIN, S.; COSTELLO, D. **Error Control Coding**. [S.l.: s.n.], 2004. ISBN 978-1-4613-6787-1.

LU, Y.; GOTO, S. A study on channel polarization and polar coding. In: **2011 9th IEEE International Conference on ASIC**. [S.l.: s.n.], 2011. p. 804–807.

MARTÃO, V. B. **Codificação de canal para redes 5G utilizando códigos polares**. Monografia — Universidade Estadual Paulista, 2018. Disponível em: <http://hdl.handle.net/11449/156785>.

MORELOS-ZARAGOZA, R. H. **The Art of Error Correcting Coding**. [S.l.: s.n.], 2006. 45 p. ISBN 0-470-01558-6.

MURATA, T.; OCHIAI, H. On design of crc codes for polar codes with successive cancellation list decoding. In: **2017 IEEE International Symposium on Information Theory (ISIT)**. [S.l.: s.n.], 2017. p. 1868–1872.

NIU, K.; CHEN, K. Crc-aided decoding of polar codes. **IEEE Communications Letters**, v. 16, n. 10, p. 1668–1671, 2012.

PATIL, M. V.; PAWAR, S.; SAQUIB, Z. Coding techniques for 5g networks: A review. In: **2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)**. [S.l.: s.n.], 2020. p. 208–213.

POPOVSKI, P. **Ultra-Reliable Communication in 5G Wireless Systems**. arXiv, 2014. Disponível em: <https://arxiv.org/abs/1410.4330>.

POPOVSKI, P. et al. Wireless access in ultra-reliable low-latency communication (urllc). **IEEE Transactions on Communications**, v. 67, n. 8, p. 5783–5801, 2019.

PRESMAN, N. **Methods in Polar Coding**. Tese (Doutorado), 08 2015.

ROSS, S. M. ***An Elementary Introduction to Mathematical Finance***. [S.l.]: Cambridge University Press, 2011. (). ISBN 9780521192538.

ROTH, R. ***Introduction to Coding Theory***. Primeira edição. [S.l.]: Cambridge University Press, 2006.

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

SAAD, W.; BENNIS, M.; CHEN, M. A vision of 6g wireless systems: Applications, trends, technologies, and open research problems. **IEEE Network**, v. 34, n. 3, p. 134–142, 2020.

SARKIS, G. et al. Flexible and low-complexity encoding and decoding of systematic polar codes. **IEEE Transactions on Communications**, v. 64, 07 2015.

SHAH, P. M.; VYAVAHARE, P. D.; JAIN, A. Modern error correcting codes for 4g and beyond: Turbo codes and ldpc codes. In: **2015 Radio and Antenna Days of the Indian Ocean (RADIO)**. [S.l.: s.n.], 2015. p. 1–2.

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

TAL, I.; VARDY, A. List decoding of polar codes. In: **2011 IEEE International Symposium on Information Theory Proceedings**. [S.l.: s.n.], 2011. p. 1–5.

TERÇAS, L. **Codificação de canal para comunicações ultra confiáveis em sistemas 5G**. Monografia — Universidade Estadual Paulista, 2019. Disponível em: <http://hdl.handle.net/11449/217184>.

VANGALA, H.; HONG, Y.; VITERBO, E. Efficient algorithms for systematic polar encoding. **IEEE Communications Letters**, IEEE, Institute of Electrical and Electronics Engineers, v. 20, n. 1, p. 17 – 20, 2016. ISSN 1089-7798.

WANG, X. et al. An optimized encoding algorithm for systematic polar codes. **EURASIP Journal on Wireless Communications and Networking**, v. 2019, 08 2019.

ŞAŞOğLU, E. Polarization and polar codes. **Foundations and Trends® in Communications and Information Theory**, v. 8, n. 4, p. 259–381, 2011.