

FELIPE GABRIEL DE MELLO ELIAS

**CONSTRUÇÃO DE CÓDIGOS LDPC OTIMIZADOS PARA  
CANAIS COM DESVANECIMENTO NAKAGAMI-*M***

CURITIBA

2011

## TERMO DE APROVAÇÃO

Felipe Gabriel de Mello Elias

### Construção de Códigos LDPC Otimizados para Canais com Desvanecimento Nakagami-m

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Paraná.



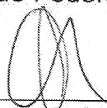
Prof. Dr. Evelio Martín García Fernández – Orientador  
Universidade Federal do Paraná



Prof. Dr. Richard Demo Souza – Co-Orientador  
Universidade Tecnológica Federal do Paraná



Prof. Dr. Alexandre Rasi Aoki – Convidado  
Universidade Federal do Paraná



Prof. Dr. Carlos Marcelo Pedroso – Convidado  
Universidade Federal do Paraná



Prof. Dr. Marcelo Eduardo Pellenz – Convidado  
Pontifícia Universidade Católica do Paraná

Curitiba, 28 de Abril de 2011

FELIPE GABRIEL DE MELLO ELIAS

**CONSTRUÇÃO DE CÓDIGOS LDPC OTIMIZADOS PARA  
CANAIS COM DESVANECIMENTO NAKAGAMI-*M***

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre. Programa de  
Pós-Graduação em Engenharia Elétrica, Setor  
de Tecnologia, Universidade Federal do Paraná.

Orientador: Prof. Dr. Evelio Martín Gar-  
cía Fernández

Coorientador: Prof. Dr. Richard Demo Souza

CURITIBA

2011

## AGRADECIMENTOS

Agradeço a Deus pelo dom da vida e por todas as bênçãos recebidas;

Agradeço a minha esposa Ana Paula pela compreensão nos momentos de ausência, pelo amor demonstrado em tantos momentos, pelos incentivos, pela paciência e pelas orações, esta conquista é nossa!

Agradeço a minha filha Penélope pela inspiração e pelo carinho e também por ceder muitos momentos para que este trabalho fosse concluído.

Agradeço aos meus familiares que me apoiaram, incentivaram e foram sempre presentes.

Agradeço ao meu orientador professor Doutor Evelio Martin Garcia Fernandez pela disposição e compreensão demonstradas em muitas horas de orientação, estudo e conversas.

Agradeço ao meu co-orientador professor Doutor Richard Demo Souza pelo ensino recebido.

Agradeço ao corpo docente do PPGEE da UFPR, incluindo o professor Doutor Marcelo Rosa, pela dedicação e pela qualidade de ensino.

Agradeço aos meus colegas Wellington Desan, Alisson Fondazzi e Glauber Brante que de alguma forma me ajudaram nesta conquista.

## RESUMO

Este trabalho realiza o estudo e a construção de códigos LDPC para o canal com desvanecimento Nakagami- $m$ . Este modelo de canal de desvanecimento vêm ganhando destaque nos últimos anos e possui uma grande importância nos sistemas de comunicação sem fio. Através do algoritmo *Density Evolution* associado ao algoritmo de busca *Differential Evolution* e do equacionamento das funções densidade de probabilidade para o canal Nakagami- $m$  foram encontrados códigos LDPC otimizados para o canal em questão. Simulações realizadas sobre o canal Nakagami- $m$  mostram que o código encontrado possui um desempenho superior a outros códigos construídos para outros modelos de desvanecimento.

**Palavras-chave:** Projeto de códigos LDPC. Algoritmo *Density Evolution*. Canal com Desvanecimento Nakagami- $m$ .

## ABSTRACT

This paper proposes the analysis and the project of LDPC codes for Nakagami- $m$  fading channels. This fading channel model have gained prominence in recent years and has a great importance in wireless communication systems. Using *Density Evolution* algorithm associated with *Differential Evolution* search algorithm and the equation of probability density functions for Nakagami- $m$  channel, optimized LDPC codes were found. Simulations on the Nakagami- $m$  channel show that the code found has a superior performance as compared with codes constructed for other fading channel models.

**Keywords:** LDPC Code Project. Density Evolution Algorithm. Nakagami- $m$  Fading Channel.

## LISTA DE FIGURAS

2.1	Grafo correspondente à matriz $H_1$ . . . . .	9
2.2	Grafo correspondente à matriz $H_1$ , com um ciclo de tamanho 6 destacado em pontilhado e <i>girth</i> de tamanho 4 destacado em negrito. . . . .	10
2.3	Grafo correspondente à matriz $H_2$ . . . . .	11
2.4	Cálculo de valor para nó de variável . . . . .	16
2.5	Cálculo de valor para nó de verificação . . . . .	16
3.1	Função densidade de probabilidade de Rayleigh para $\sigma_{Ray} = 1$ . . . . .	20
3.2	Função densidade de probabilidade de Rician para diferentes valores de $K$ . . . . .	21
3.3	Função densidade de probabilidade Nakagami- $m$ para diferentes valores de $m$ e $2\sigma^2 = 1$ . . . . .	24
3.4	Função densidade de probabilidade $p_h(x)$ e função densidade de probabilidade $p_N(x)$ para $m = 0,8$ . . . . .	26
5.1	Fluxograma de etapas do projeto para a criação de um código LDPC otimizado para canal Nakagami- $m$ . . . . .	41
5.2	Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Nakagami- $m = 0,8$ . . . . .	45
5.3	Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Nakagami- $m = 2,1$ . . . . .	46
5.4	Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Nakagami- $m = 4$ . . . . .	47
5.5	Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Rayleigh. . . . .	48
5.6	Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal AWGN. . . . .	49
6.1	Grafo correspondente à matriz $H_3$ . . . . .	59

## LISTA DE TABELAS

5.1	Funções de distribuição de grau para canais Nakagami- $m$ com $m = 0,8$ , $m = 2,1$ e $m = 4$ . . . . .	44
5.2	Funções de distribuição de grau para canais Rayleigh e AWGN . . . . .	45

## CONTEÚDO

<b>RESUMO</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>LISTA DE FIGURAS</b>	<b>iv</b>
<b>LISTA DE TABELAS</b>	<b>v</b>
<b>1 INTRODUÇÃO</b>	<b>2</b>
1.1 Contexto . . . . .	2
1.2 Objetivos . . . . .	3
1.3 Justificativa . . . . .	3
1.4 Metodologia de Construção . . . . .	5
1.5 Resultados . . . . .	6
1.6 Estrutura da Dissertação . . . . .	6
<b>2 CÓDIGOS LDPC</b>	<b>8</b>
2.1 Introdução . . . . .	8
2.2 Codificação . . . . .	13
2.3 Decodificação . . . . .	14
2.3.1 Decodificação <i>Belief-Propagation</i> . . . . .	14
2.4 Algoritmo PEG ( <i>Progressive Edge Growth</i> ) . . . . .	17
<b>3 DESVANECIMENTO EM CANAIS DE COMUNICAÇÃO E ASPEC-</b>	
<b>TOS GERAIS DO DESVANECIMENTO NAKAGAMI-<math>M</math></b>	<b>19</b>
3.1 Introdução . . . . .	19
3.2 Desvanecimento Rayleigh . . . . .	20
3.3 Desvanecimento Rician . . . . .	21
3.4 Desvanecimento Nakagami- $m$ . . . . .	22

	1
3.5	Geração de variáveis Nakagami- $m$ . . . . . 24
<b>4</b>	<b>TÉCNICA DE BUSCA DE CÓDIGOS</b> <span style="float: right;"><b>27</b></span>
4.1	Algoritmo DE . . . . . 27
4.1.1	Algoritmo de busca <i>Differential Evolution</i> . . . . . 30
4.2	DE para canal AWGN e canais com desvanecimento . . . . . 33
4.2.1	Análise para canal AWGN . . . . . 33
4.2.2	Análise para canal Rayleigh . . . . . 34
4.2.3	Análise para canal Nakagami- $m$ . . . . . 36
4.2.4	Simetria e estabilidade para canal Nakagami- $m$ . . . . . 37
<b>5</b>	<b>IMPLEMENTAÇÃO E RESULTADOS OBTIDOS</b> <span style="float: right;"><b>40</b></span>
5.1	Implementação . . . . . 40
5.2	Resultados Obtidos . . . . . 42
5.2.1	Resultados para $m = 0,8$ . . . . . 43
5.2.2	Resultados para $m = 2,1$ . . . . . 44
5.2.3	Resultados para $m = 4$ . . . . . 47
5.2.4	Outros Resultados . . . . . 48
<b>6</b>	<b>CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS</b> <span style="float: right;"><b>50</b></span>
	<b>BIBLIOGRAFIA</b> <span style="float: right;"><b>56</b></span>
	<b>APÊNDICE A</b> <span style="float: right;"><b>57</b></span>
	<b>APÊNDICE B</b> <span style="float: right;"><b>59</b></span>

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 Contexto

Os fundamentos da teoria da informação foram introduzidos por Shannon [1], o qual provou que é possível transmitir informação de maneira confiável através de um canal ruidoso se a taxa de transmissão é menor do que a capacidade do canal. Com o intuito de conseguir alcançar o limiar proposto por Shannon muitos pesquisadores têm trabalhado no aperfeiçoamento de técnicas de códigos corretores de erros (códigos de canal). Os primeiros resultados neste campo foram obtidos logo após a tese de Shannon ser publicada, como por exemplo os códigos corretores de erros de Hamming e de Reed-Solomon [2], os quais ainda são usados em aplicações como a correção de erros em CDs e nos sistemas de radiodifusão de televisão digital.

Alguns códigos de correção de erros modernos têm chamado a atenção de pesquisadores; estes possuem maior desempenho de correção se comparados a códigos clássicos, sendo seu diferencial o mecanismo de decodificação iterativo baseado nas probabilidades das mensagens recebidas. Códigos Turbo ou códigos *Low-Density Parity-Check* (em português, Verificador de Paridade de Baixa Densidade, ou simplesmente LDPC) possuem excelente desempenho [3], chegando muito próximo do limite de Shannon [4].

Os códigos de bloco clássicos (como códigos de Hamming ou de Reed-Solomon) são construídos sobre álgebras específicas e realizam sua decodificação através do cálculo de máxima verossimilhança entre a palavra recebida e as possíveis palavras código geradas. Diferentemente, os códigos LDPC são códigos de blocos lineares cuja decodificação é realizada através de um cálculo iterativo de probabilidades a partir das mensagens recebidas. Tais códigos LDPC foram descobertos na década de 60 por Gallager [5], estes eram não factíveis de implementação computacional na época e acabaram sendo esquecidos por anos. No início da década de 80 foi desenvolvida a teoria dos grafos de Tanner [6] que

consiste na sistematização da decodificação destes códigos e então os mesmos voltaram a ter destaque na década de 90 sendo redescobertos por diferentes pesquisadores, como MacKay e Neal [3] e Luby *et al* [7]. Desde então vêm sendo utilizados em muitas aplicações que necessitam de um esquema de correção de erros de alto desempenho. Entre algumas aplicações que utilizam códigos LDPC estão: o padrão europeu de sistema de televisão digital via satélite (DVB-S2) [8], o padrão 10Gb/s Ethernet(IEEE 802.3an) [9], os padrões WiFi (IEEE 802.11n) [10] e WiMAX (IEEE 802.16e) [11], além de diversos tipos de aplicações em armazenamento digital e redes ópticas [2].

## 1.2 Objetivos

O presente trabalho propõe a construção de códigos LDPC específicos para o canal de transmissão sem fio com desvanecimento Nakagami-m. Para a validação deste objetivo os códigos encontrados serão simulados e seu desempenho será analisado.

Para encontrar códigos otimizados para este canal serão utilizadas duas técnicas específicas: *Density Evolution* [12] (em português, Evolução de Densidade, ou simplesmente DE) e *Differential Evolution* [13] (em português, Evolução Diferencial). Estas técnicas associadas e projetadas para o canal Nakagami-m resultam em um conjunto de códigos LDPC otimizados para o canal.

## 1.3 Justificativa

O projeto de sistemas de comunicação, incluindo a construção de códigos corretores de erros otimizados, deve levar em consideração o modelo mais adequado ao sistema real. Utilizando a modelagem correta do canal de comunicação é possível obter ganhos de correção de erros, o que implica em menor potência transmitida e conseqüentemente um ganho de economia no projeto.

A maior parte dos trabalhos publicados na área de projeto de códigos LDPC é para o canal com ruído branco gaussiano aditivo (AWGN) [4] [12] [14] [15]. O canal AWGN é um modelo simples de canal de comunicações que leva em consideração apenas a adição do

ruído branco que possui densidade espectral constante e uma distribuição com amplitude gaussiana. O ruído Gaussiano provém da agitação térmica de átomos em condutores e é um ótimo modelo para comunicações via satélite e espaciais, entretanto não é o modelo adequado para transmissões sem fio em meios urbanos por causa dos fenômenos de multipercurso e obstrução de sinal por exemplo.

Por sua vez, o projeto de códigos LDPC para uso em comunicações sem fio tem que levar em conta o fenômeno do desvanecimento. O desvanecimento é um fenômeno que ocorre em canais de comunicação sem fio onde a visada direta entre antenas do transmissor e receptor é total ou parcialmente obstruída. Apenas uma pequena quantidade de trabalhos publicados levam em conta o projeto de códigos LDPC para canais com desvanecimento [16] [17].

Em cenários de comunicação sem fio onde a linha de visada direta entre transmissor e receptor está totalmente obstruída, o desvanecimento Rayleigh é utilizado para modelagem estatística do comportamento do canal ao invés do modelo AWGN [18]. Um exemplo de cenário onde o desvanecimento Rayleigh é uma boa opção de modelagem é o de ambientes densamente urbanos. O projeto específico de códigos LDPC para este canal foi muito explorado em [16], onde foi mostrado de maneira detalhada a metodologia de construção. Ali também foi mostrado que códigos projetados especificamente para o canal Rayleigh possuem um ganho de cerca de 1 dB, em termos de relação Sinal-Ruído (*Signal-to-Noise Ratio* ou SNR) com taxa de erro de bit (*Bit Error Rate* ou BER) de  $10^{-5}$  se comparados a códigos não otimizados (por exemplo, códigos projetados para canal AWGN).

Entretanto, muitos cenários de comunicação sem fio possuem uma linha de visada direta entre transmissor e receptor apenas parcialmente obstruída, como redes em malha e redes de sensores sem fio. Nestes casos o modelo de desvanecimento Rayleigh não é adequado. Uma opção é o modelo de desvanecimento Rician [19]; o qual é um modelo mais geral, e se torna igual a distribuição Rayleigh se a componente de visada direta é nula. Na literatura também foi encontrado o estudo e o projeto de códigos LDPC para este tipo de canal [17]. Neste trabalho o desempenho do código otimizado possui BER de

$10^{-5}$  para SNR de  $2dB$ .

Por sua vez, o modelo de canal com desvanecimento Nakagami- $m$  também é uma generalização e engloba outros modelos de desvanecimento como Rayleigh e Rician. Resultados recentes mostram que sistemas reais são melhor aproximados pelo modelo de desvanecimento Nakagami- $m$  [20], como por exemplo: comunicações via rádio em ambientes *indoor* e *outdoor* como mostrado no trabalho de Yacoub [21]; sistemas GPS para canais satélite-*indoor* e satélite-*outdoor* [22]; e também sistemas celulares em alguns ambientes urbanos CDMA [23] e 3G [24]. Foram encontradas na literatura poucas referências relacionando códigos LDPC e canal com desvanecimento Nakagami- $m$  [25] [26], sendo que não foi encontrada nenhuma referência de projeto de códigos LDPC para o canal Nakagami- $m$  utilizando o algoritmo DE, metodologia proposta neste trabalho.

O ganho obtido é muito alto para um sistema de comunicações sem fio. Se for levado em consideração que o projeto deverá ser realizado apenas uma vez e depois disso o código será utilizado por muitos usuários, a economia obtida é multiplicada de acordo com o número de usuários do sistema. Portanto a economia total do sistema será muito elevada, justificando desta maneira o trabalho em questão.

## 1.4 Metodologia de Construção

A metodologia de construção dos códigos é baseada no uso da técnica DE associada à técnica de busca *Differential Evolution*. Usando tais técnicas foram encontrados *ensembles* (em português, conjuntos) de códigos LDPC otimizados para o canal em questão.

O método DE é utilizado para encontrar a mínima SNR na qual uma mensagem transmitida ainda é possível de ser corrigida para um *ensemble* de códigos LDPC. O nível mínimo de SNR encontrado é chamado de limiar, sendo que o projetista do código poderá encontrar um *ensemble* com o melhor limiar; ou seja encontrar o melhor conjunto de códigos para um determinado canal [2].

Contudo, o método DE retorna apenas o limiar para um único *ensemble*. Para que o projetista possa encontrar o *ensemble* ótimo é necessário efetuar a busca entre os infinitos *ensembles* existentes, o que é praticamente inviável. Portanto é necessário a utilização

de uma técnica de busca que seja rápida e considere um espaço finito de amostras para análise. Em geral, um algoritmo de busca inteligente é associado ao método DE para encontrar o melhor *ensemble*, em uma população de diversos *ensembles* analisados em paralelo. O algoritmo de busca mais utilizado é chamado *Differential Evolution*, o qual será também explorado neste trabalho.

## 1.5 Resultados

Alguns códigos foram projetados para diferentes cenários modelados pela distribuição Nakagami- $m$  (diferentes tipos de cenários de acordo com a severidade do desvanecimento).

As simulações realizadas levaram em consideração apenas a modulação BPSK (*binary phase-shift keying*), uma modulação simples utilizada apenas para alcançar o objetivo do projeto que é a análise do desempenho do decodificador de erros.

Os resultados de simulação mostram que para todos os casos os códigos encontrados possuem desempenho superior ao de outros códigos construídos para outros modelos de desvanecimento (quando usados em um cenário Nakagami- $m$ ). Conforme será mostrado mais adiante através das otimizações para o canal Nakagami- $m$  é possível obter um ganho de pelo menos 0,2 a 0,3 dB de SNR. Conforme mencionado anteriormente esse ganho será multiplicado pelo número de usuários do sistema elevando a economia total.

## 1.6 Estrutura da Dissertação

A presente dissertação está estruturada da seguinte maneira: no Capítulo 2 serão apresentadas as principais características e propriedades dos códigos LDPC como métodos de construção, codificação e decodificação e o algoritmo *Progressive Edge Growth* que serve para gerar uma matriz de verificação de paridade otimizada de acordo com o *ensemble* escolhido. No Capítulo 3 será abordado o tema desvanecimento, onde serão mostrados os principais modelos e será mostrada a técnica para geração de variáveis com distribuição Nakagami- $m$ . Em seguida, no Capítulo 4 será discutido o algoritmo *Density Evolution* assim também como o algoritmo de busca *Differential Evolution*. No Capítulo 5 serão

apresentadas mais considerações sobre a metodologia envolvida no projeto de códigos e serão mostrados os resultados obtidos. Por fim, no Capítulo 6 serão apresentadas as considerações finais além de sugestões de trabalhos futuros.

## CAPÍTULO 2

### CÓDIGOS LDPC

#### 2.1 Introdução

Códigos LDPC realizam a codificação de uma mensagem de tamanho  $K$  bits a partir da adição de  $N - K$  bits de redundância, gerando assim uma palavra código de  $N$  bits. A decodificação é realizada utilizando uma matriz  $H$  chamada matriz de paridade, a codificação é realizada através da matriz geradora,  $G$ , (construída a partir de  $H$ ).

A matriz  $H$  de um código LDPC possui poucos elementos diferentes de zero, sendo portanto de baixa densidade. Além disso, códigos LDPC em geral são códigos binários construídos sobre o corpo de Galois  $\text{GF}(2)$ .

A matriz  $H$  possui  $N$  colunas e  $N - K$  linhas onde cada linha pode ser vista como uma equação de verificação de paridade do código. A taxa  $R$  de um código LDPC é dada por:

$$R = \frac{K}{N}. \quad (2.1)$$

Estes códigos podem ser divididos em dois tipos de acordo com a regularidade dos elementos das matrizes: códigos regulares e códigos irregulares. São chamados códigos LDPC regulares se existe uma quantidade fixa  $d_v$  de elementos diferentes de zero em cada coluna da matriz  $H$  e uma quantidade fixa  $d_c$  de elementos também diferentes de zero em cada uma das linhas da matriz de paridade. Neste caso o código é denotado como regular  $(d_v, d_c)$ . Por exemplo, a matriz:

$$H_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

é regular pois possui sempre quatro “uns” para cada uma das linhas e uma quantidade de dois “uns” para cada uma das colunas. Este código é chamado de LDPC regular (2,4). A matriz  $H_1$  possui 8 colunas e 4 linhas, portanto  $N = 8$  e  $m = 4$ .

Códigos LDPC também podem ser representados por grafos de Tanner [6] que são grafos bipartidos possuindo dois conjuntos de nós. O grafo possui  $N$  nós de variáveis (ou nós de bits) denotados  $v$  para as palavras codificadas e  $m$  nós de verificação (ou nós de cheque de paridade) denotados  $c$  para as equações de paridade. Uma aresta liga um nó de variável a um nó de verificação se e somente se existe um bit “um” na linha da matriz correspondente. No caso anterior, a matriz  $H_1$  possui o grafo de Tanner da Figura 2.1.

Analisando a correspondência entre a matriz  $H_1$  e o grafo de Tanner mostrado na figura 2.1 é possível observar que para a primeira linha existem quatro “uns” de acordo com a segunda, a quarta, a quinta e a oitava colunas respectivamente. Isto é compatível com o grafo cujo primeiro nó de verificação  $c_0$  (corresponde a primeira linha da matriz  $H_1$ ) está conectado através de arestas aos nós de variáveis  $v_1, v_3, v_4$  e  $v_7$  (corresponde aos bits “uns” da segunda, da quarta, da quinta e da oitava colunas respectivamente).

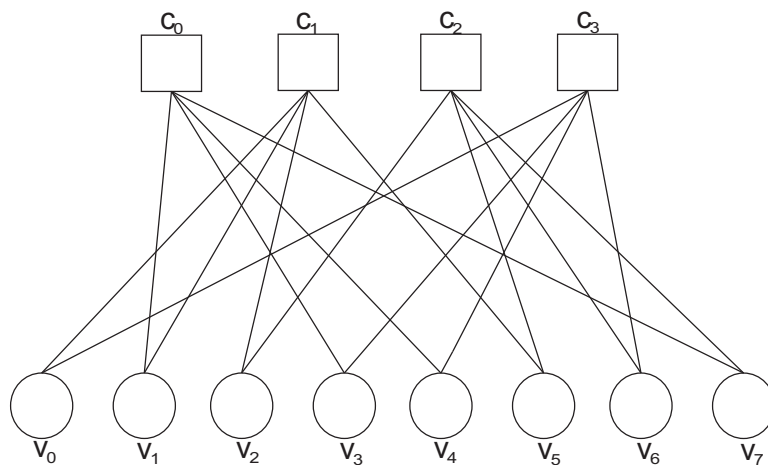


Figura 2.1: Grafo correspondente à matriz  $H_1$ .

Um código LDPC regular é caracterizado por um grafo de Tanner onde todos os nós de mesmo tipo possuem o mesmo número de arestas conectadas, como no exemplo acima. É possível perceber que cada um dos nós de variáveis  $v$  ( $v_0, v_1, \dots, v_7$ ) é conectado a somente dois nós de verificação  $c$ , através de duas arestas distintas. Cada um dos nós de verificação  $c$  ( $c_0, \dots, c_3$ ) é conectado a somente quatro nós de variáveis  $v$ , através de quatro arestas distintas.

O grafo de Tanner possui algumas propriedades específicas muito importantes para o desempenho de correção de erros do código:

- **Ciclo:** é uma sequência de nós e arestas conectados no qual o primeiro nó é também o nó final e não existe repetição de nó dentro da sequência.
- **Tamanho de um Ciclo:** é determinado pelo número de arestas que o ciclo contém;
- **Girth:** é o tamanho do menor ciclo do grafo.

O grafo da Figura 2.2, é o respectivo grafo da matriz  $H_1$  com um ciclo de tamanho seis destacado em pontilhado e *girth* de tamanho quatro destacado em negrito.

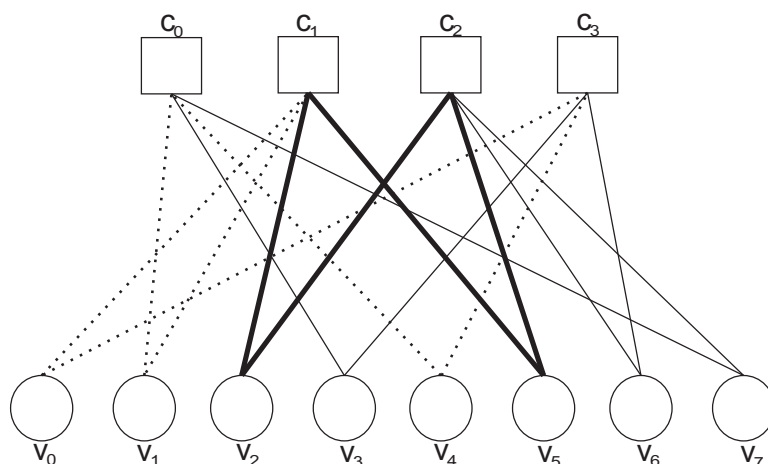


Figura 2.2: Grafo correspondente à matriz  $H_1$ , com um ciclo de tamanho 6 destacado em pontilhado e *girth* de tamanho 4 destacado em negrito.

Os códigos regulares possuem uma quantidade de elementos “uns” igual entre suas colunas e uma quantidade igual de elementos “uns” entre suas linhas. Por sua vez, códigos LDPC são irregulares se o número de elementos diferentes de zero em cada coluna não é fixo e se o número de elementos diferentes de zero em cada linha também não é fixo. Por exemplo, a matriz:

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

possui três elementos diferentes de zero na primeira coluna, dois elementos diferentes de zero na segunda, na terceira e na quarta colunas e três elementos diferentes de zero nas três últimas colunas.

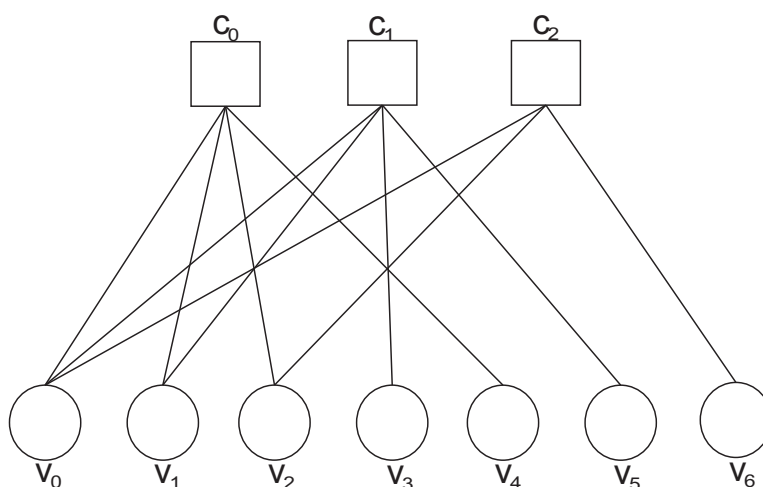


Figura 2.3: Grafo correspondente à matriz  $H_2$ .

A figura 2.3 mostra o grafo correspondente à matriz  $H_2$ , podendo-se analisar a correspondência entre a estrutura da matriz  $H_2$  e a forma gráfica da Figura 2.3. É possível notar que três arestas estão conectadas ao nó de variável  $v_0$  de acordo com a primeira coluna da matriz  $H_2$  que contém três elementos diferentes de zero. Também estão conectadas duas arestas em cada um dos nós  $v_1, v_2$  e  $v_3$  de acordo com a segunda, a terceira e a quarta coluna, que contém apenas dois elementos diferentes de zero. Estão conectados também apenas uma aresta para cada um dos nós de variáveis  $v_4, v_5$  e  $v_6$  em conformidade com as três últimas colunas da matriz  $H_2$  que contém apenas um elemento diferente de zero.

Ainda analisando a relação entre a matriz  $H_2$  e o grafo da figura 2.3 é possível observar a quantidade de elementos diferentes de zero entre as linhas da matriz  $H_2$  e nos nós de verificação. Cada linha possui uma quantidade de “uns” correspondente ao número de

arestas que emanam de cada nó de verificação. Neste caso as duas primeiras linhas possuem a mesma quantidade, quatro elementos diferentes de zero, e quatro arestas saem de cada nó de verificação  $c_0$  e  $c_1$ . A última linha possui três elementos diferentes de zero sendo que três arestas saem do nó  $c_2$ .

Uma maneira importante para definir um código ou mais códigos LDPC irregulares é através das chamadas funções de distribuição de grau. Para códigos LDPC irregulares, a fração de colunas com peso (quantidade de elementos diferentes de zero na coluna)  $i$  é denotada por  $\lambda_i$  e a fração de linhas com peso  $i$  é  $\rho_i$ . Os maiores pesos para as colunas e as linhas são denotados  $d_{v_{max}}$  ou  $\lambda_{max}$  e  $d_{c_{max}}$  ou  $\rho_{max}$ , respectivamente. A soma total das frações é equivalente a 1 de acordo com as equações:

$$\sum_{i=2}^{d_{v_{max}}} \lambda_i = 1, \quad (2.2)$$

$$\sum_{i=2}^{d_{c_{max}}} \rho_i = 1. \quad (2.3)$$

Por sua vez, as funções:

$$\lambda(x) = \sum_{i=2}^{d_{v_{max}}} \lambda_i x^{i-1}, \quad (2.4)$$

$$\rho(x) = \sum_{i=2}^{d_{c_{max}}} \rho_i x^{i-1}, \quad (2.5)$$

são chamadas de par de funções de distribuição de graus do código.

As matrizes  $H_1$  e  $H_2$  são matrizes que servem apenas para fins de exemplo e não são utilizadas na prática para codificação/decodificação. Matrizes reais são em geral muito

grandes da ordem de centenas ou milhares de elementos por linha e por coluna. Estas matrizes não possuem elemento  $\lambda_1$  como nas matrizes dadas nos dois exemplos, por isso os somatórios das equações (2.2), (2.3), (2.4), (2.5) são contados a partir do índice  $i = 2$ .

O exemplo da matriz irregular  $H_2$  possui as seguintes distribuições:

$$\lambda(x) = 0,571 + 0,286x + 0,143x^2, \quad (2.6)$$

$$\rho(x) = 0,333x^2 + 0,667x^3. \quad (2.7)$$

De acordo com a equação (2.6) 57,1% das colunas possuem apenas um bit “um” , outros 28,6% das colunas possuem dois bits “um” e 14,3% das colunas possuem três bits “um” . De acordo com a equação (2.7) 33,3% das linhas possuem três elementos diferentes de zero e 66,7% possuem quatro elementos diferentes de zero.

O par de funções de distribuição de graus do código define um conjunto (também conhecido como *emsemble*) de diversos códigos LDPC que possuem o mesmo comportamento em termos de desempenho de decodificação.

## 2.2 Codificação

A codificação LDPC é realizada a partir de

$$\mathbf{u}G = \mathbf{c}, \quad (2.8)$$

onde o vetor  $\mathbf{u}$  representa a mensagem a ser transmitida,  $G$  é a matriz geradora definida a partir da matriz  $H$  e o vetor  $\mathbf{m}$  é a palavra código obtida. A matriz  $H$  pode ser ajustada através do processo de eliminação de Gauss-Jordan [2] até ser colocada na forma:

$$H = [A \ I_{N-K}], \quad (2.9)$$

onde  $A$  é uma matriz de tamanho  $(N - K) \times K$  e  $I$  é uma matriz identidade de tamanho  $N - K$ . Por sua vez:

$$G = [I_K \ A^T]. \quad (2.10)$$

Onde  $A^T$  é matriz transposta de  $A$  e  $I$  é a matriz identidade, agora de tamanho  $K$ . Um exemplo de codificação é mostrado no Apêndice A.

## 2.3 Decodificação

A decodificação de códigos LDPC é realizada por uma classe de algoritmos iterativos chamados *message-passing* (em português, passagem de mensagem). Seu princípio é a troca de mensagens, durante certo número de iterações, dos nós de variáveis para nós de verificação e dos nós de verificação para nós de variáveis. Existem duas subclasses de algoritmos de decodificação *message-passing*, a primeira chamada de *Bit-Flipping* e a segunda chamada de *Belief-Propagation* (em português, propagação de crença). Esta segunda técnica é a mais utilizada e possui um desempenho superior [27] [2].

### 2.3.1 Decodificação *Belief-Propagation*

O algoritmo de decodificação *Belief-Propagation* é também chamado de Soma-Produto. Ele é baseado no princípio que se um bit incorreto da palavra código for recebido, estará contido em um grande número de equações de verificação de paridade sem bits incorretos, e cada uma destas poderá calcular o valor correto para o determinado bit errado. A esparsidade da matriz  $H$  ajuda a dispersar os bits das equações de verificação de paridade fazendo com que duas equações sejam muito pouco prováveis de conterem os mesmos bits da palavra código.

Um ponto importante na decodificação consiste em analisar se a palavra código em questão pertence ao conjunto de palavras possíveis de serem geradas pelo código. Para verificar se a palavra código recebida  $\mathbf{y}$  é uma palavra código válida para o código em questão é realizado o cálculo da síndrome  $\mathbf{s}$ , onde:

$$\mathbf{s} = H\mathbf{y}^T. \quad (2.11)$$

Se a palavra código  $\mathbf{y}$  pertence ao código em questão o valor da síndrome deve ser o vetor  $\mathbf{s}$  com todos os elementos zero. Se pelo menos um elemento do vetor síndrome for diferente de zero a palavra código não está correta e existem erros que necessitam de correção.

Este algoritmo utiliza decisão suave e as mensagens trocadas representam probabilidades, os valores representam o grau de confiança de um valor de bit. O algoritmo também aceita como entrada valores probabilísticos para cada bit recebido, estes valores são chamados de probabilidades *a priori*. Para uma variável binária  $x$  é necessário analisar apenas um valor de probabilidade pois  $p(x = 1) = 1 - p(x = 0)$ .

A relação logarítmica usada para representar os valores das variáveis é dada por:

$$L(x) = \ln \frac{p(x = 0)}{p(x = 1)}. \quad (2.12)$$

O valor do sinal de  $L(x)$  representa a decisão abrupta sobre  $x$  e a magnitude  $|L(x)|$  representa a confiança da variável.

Para cada iteração da decodificação são realizados dois processos:

- Cálculo de nó de variável: Para um nó de variável  $i$  é computada a mensagem  $E_{j,i}$  que é a informação extrínseca recebida do nó de verificação  $j$  baseado no que  $j$  recebeu dos demais nós de variáveis.

De acordo com a Figura 2.4 [27] as arestas que vão dos nós de variáveis para o nó de verificação  $j$  estão evidenciadas, com exceção da aresta em pontilhado, pois não deve ser enviada informação do nó  $i$  para o nó  $j$  para não haver redundância de informações. A mensagem extrínseca  $E_{j,i}$  (também evidenciada na Figura 2.4) que sai do nó de verificação  $j$  para o nó de variável  $i$ , é a opinião do nó  $j$  sobre a probabilidade do valor do bit  $c_i = 1$  da mensagem original codificada, baseado na informação disponível a partir do que o nó

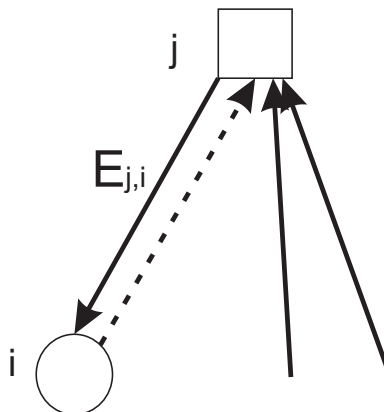


Figura 2.4: Cálculo de valor para nó de variável

$j$  recebeu dos outros nós de variável. A mensagem  $E_{j,i}$  é calculada [2] como:

$$E_{j,i} = 2 \tanh^{-1} \prod_{i' \neq i} \tanh(M_{j,i'}/2). \quad (2.13)$$

Onde  $M_{j,i'}$  é a mensagem enviada do nó  $i'$  para o nó  $j$ . Para que a mensagem transmitida realmente seja extrínseca é necessário retirar a informação redundante  $M_{j,i}$  que tenha sido transmitida do nó de bit  $i$  para o nó de verificação  $j$  durante a iteração anterior, por isso  $i' \neq i$ .

- Cálculo de nó de verificação: Para um nó de verificação  $j$  é computada a informação extrínseca recebida do nó de variável  $i$  baseado no que  $i$  recebeu dos demais nós de verificação, mais a mensagem proveniente do canal.

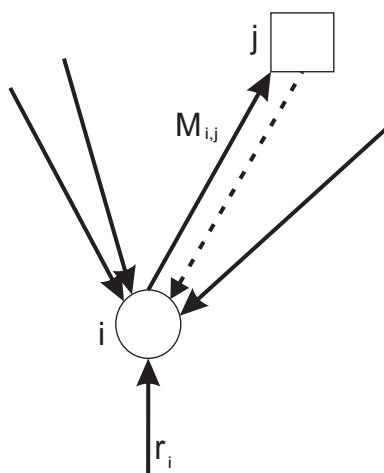


Figura 2.5: Cálculo de valor para nó de verificação

De acordo com a Figura 2.5 [27], é possível observar as mensagens enviadas de todos os nós de verificação com exceção da mensagem enviada de  $i$  para  $j$  (em pontilhado).

A mensagem  $M_{i,j}$  sai do nó de variável  $i$  para o nó de verificação  $j$  e é calculada [2] como:

$$M_{i,j} = \sum_{j' \neq j} E_{j',i} + r_i, \quad (2.14)$$

onde  $r_i$  é a informação *a priori* do canal. Para evitar de enviar novamente para cada nó de variável informação que este já possui, a mensagem enviada do nó de variável  $i$  para o nó de verificação  $j$ , representada na equação (2.14), não possui o componente  $E_{j,i}$  que acabou de ser recebido do nó  $j$ , por isso  $j' \neq j$ . Por sua vez,  $r_i$  é definido como:

$$r_i = \ln \frac{p(m_i = 0|y_i)}{p(m_i = 1|y_i)}. \quad (2.15)$$

O algoritmo *Belief-Propagation* processa um número estabelecido de iterações com troca de mensagens entre nós de variável e de verificação. Depois deste número máximo (deve ser bem projetado para o canal em questão) de iterações o algoritmo converge e não há ganho de correção de erros. O algoritmo também pode parar se uma palavra código válida é encontrada através da verificação de síndrome (2.11)[27].

Um exemplo de decodificação é mostrado no Apêndice B.

## 2.4 Algoritmo PEG (*Progressive Edge Growth*)

Para a construção da matriz verificadora de paridade  $H$  dos códigos introduzidos nesta dissertação foi utilizado o algoritmo PEG (em inglês, *Progressive Edge Growth*) proposto originalmente em [15]. Para um determinado par de distribuição de graus o algoritmo de construção PEG gera grafos de Tanner com desempenho de correção superior a grafos gerados por outros métodos [15] para o mesmo par de funções  $(\lambda, \rho)$  que geram um *ensemble*.

Na entrada do algoritmo PEG é inserido o par de funções de distribuição de grau

encontrados através do algoritmo *Density Evolution*. O algoritmo realiza a construção de uma matriz  $H$  cujos pesos das colunas e linhas obedecem às distribuições escolhidas. O algoritmo é inicializado com um valor de nós de variáveis, um número específico de nós de verificação e com a sequência de graus de nós de variáveis  $\lambda$ . Através destes parâmetros o objetivo do algoritmo é a construção de um grafo inserindo neste uma aresta por vez, e cada aresta é adicionada de forma que ela maximize o *girth* local, que como foi citado anteriormente é o tamanho do menor ciclo do grafo.

Dado que os nós de variáveis com menor grau são mais susceptíveis a erros (pois recebem menor quantidade de informação dos nós conectados), a inserção de arestas começa com os nós de variáveis de menor grau e progride para os nós de variáveis de maior grau. O algoritmo não vai para o próximo nó de variável enquanto todas as arestas do nó atual não foram inseridas.

A primeira aresta que sai do primeiro nó de variável é conectada a um nó de verificação com menor grau possível no estado atual do grafo. As outras inserções de arestas a um nó de variável são feitas para que o *girth* local para aquele nó de variável seja o maior possível para o grafo. Desta maneira evitando *girth* de tamanho quatro ou seis pois para o algoritmo de decodificação *Belief-Propagation*, os ciclos em um grafo de Tanner levam à correlações nas probabilidades marginais passadas adiante pelo decodificador. Quanto menor o ciclo, menor será o número de iterações que são livres de correlações de acordo com [2]. Ciclos pequenos levam a um perda de desempenho do decodificador.

Como o grafo é construído por etapas, é analisado um nó de variável por vez e é adicionada uma aresta por vez. O estado atual do grafo é tal que um ou mais nós de verificação não poderão ser alcançados pelo nó de variável atual atravessando todas as arestas conectadas até então; a aresta deve ser conectada a um nó de verificação inalcançável, dessa forma nenhum ciclo é criado. Caso contrário, se todos os nós de verificação são alcançáveis pelo nó de variável atual através de algum número de arestas, a nova aresta deve ser conectada a um nó de verificação que resulte no maior *girth* do ponto de vista do nó de variável atual.

# CAPÍTULO 3

## DESVANECIMENTO EM CANAIS DE COMUNICAÇÃO E ASPECTOS GERAIS DO DESVANECIMENTO NAKAGAMI- $M$

### 3.1 Introdução

Em um meio de transmissão sem fio as ondas de rádio partem de uma antena transmissora até uma antena receptora. Durante a transmissão as ondas sofrem os efeitos de diferentes tipos de fenômenos como: absorção, reflexão, refração, difração e espalhamento.

No caso de sistemas sem fio como o sistema celular, em geral não existe uma linha de visada direta entre as antenas transmissora e receptora pois a altura da antena do dispositivo geralmente é muito menor do que as estruturas que se encontram ao seu redor. Deste modo o sinal transmitido chega ao destino através de diferentes percursos. Este fenômeno é chamado de multi-percurso e pode exercer influência construtiva ou destrutiva sobre a onda de rádio afetando a recepção final. Este fenômeno causa mudanças na amplitude e na fase do sinal no receptor provocando o desvanecimento do sinal recebido [19].

Os tipos de desvanecimento podem ser classificados em pequena ou larga escala, dependendo da velocidade da flutuação da potência do sinal recebido. Neste trabalho será analisado apenas o tipo de desvanecimento de pequena escala. Dada a sua característica aleatória, para a modelagem de diferentes canais de comunicação sem fio que sofrem desvanecimento são utilizadas funções de densidade de probabilidades. Existem diversos modelos de desvanecimento em pequena escala, os mais usados em cenários de comunicação sem fio são: desvanecimento Rayleigh, desvanecimento Rician e desvanecimento Nakagami- $m$ .

## 3.2 Desvanecimento Rayleigh

A distribuição Rayleigh têm sido amplamente utilizada para a modelagem do desvanecimento em cenários de comunicação sem fio. Este modelo é aplicado em cenários onde não há linha de visada direta entre transmissor e receptor [18]. A distribuição de Rayleigh possui função de densidade de probabilidade (pdf) definida por:

$$p_R(a) = \frac{a}{\sigma_{Ray}^2} \exp\left(-\frac{a^2}{2\sigma_{Ray}^2}\right), \quad (3.1)$$

onde  $a$  é o fator de desvanecimento Rayleigh (no caso  $a \geq 0$ ),  $\sigma_{Ray}$  é o desvio padrão da pdf e o valor  $2\sigma_{Ray}^2$  é igual a  $E[a^2]$ . A figura 3.1 mostra a função densidade de probabilidade Rayleigh para  $\sigma_{Ray} = 1$ .

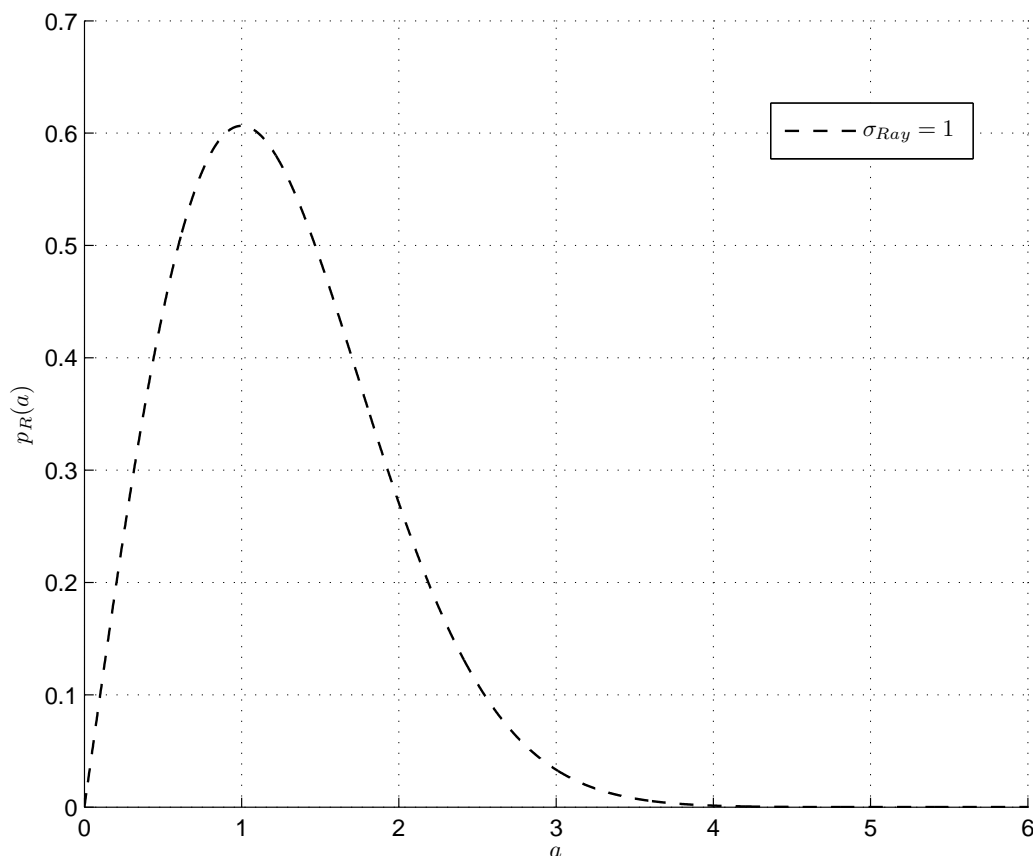


Figura 3.1: Função densidade de probabilidade de Rayleigh para  $\sigma_{Ray} = 1$ .

### 3.3 Desvanecimento Rician

O desvanecimento Rician é geralmente usado para modelar cenários onde existe uma parcela de linha de visada direta entre transmissor e receptor [28]. A função de distribuição de Rician é dada por:

$$p_{Ri}(a) = \frac{a}{\sigma_{Ri}^2} \exp\left(-\frac{a^2 + \gamma^2}{2\sigma_{Ri}^2}\right) I_0\left(\frac{a\gamma}{\sigma_{Ri}^2}\right), \quad (3.2)$$

onde  $\gamma^2$  é a energia da componente da linha de visada direta, o fator  $K = \frac{\gamma^2}{2\sigma_{Ri}^2}$  é a relação entre a energia da visada direta e a energia dos componentes das ondas dispersas,  $I_0(\cdot)$  é a função de Bessel [29] modificada de ordem zero,  $a$  é o fator de desvanecimento Rician e  $\sigma_{Ri}$  é o desvio padrão da função densidade de probabilidade.

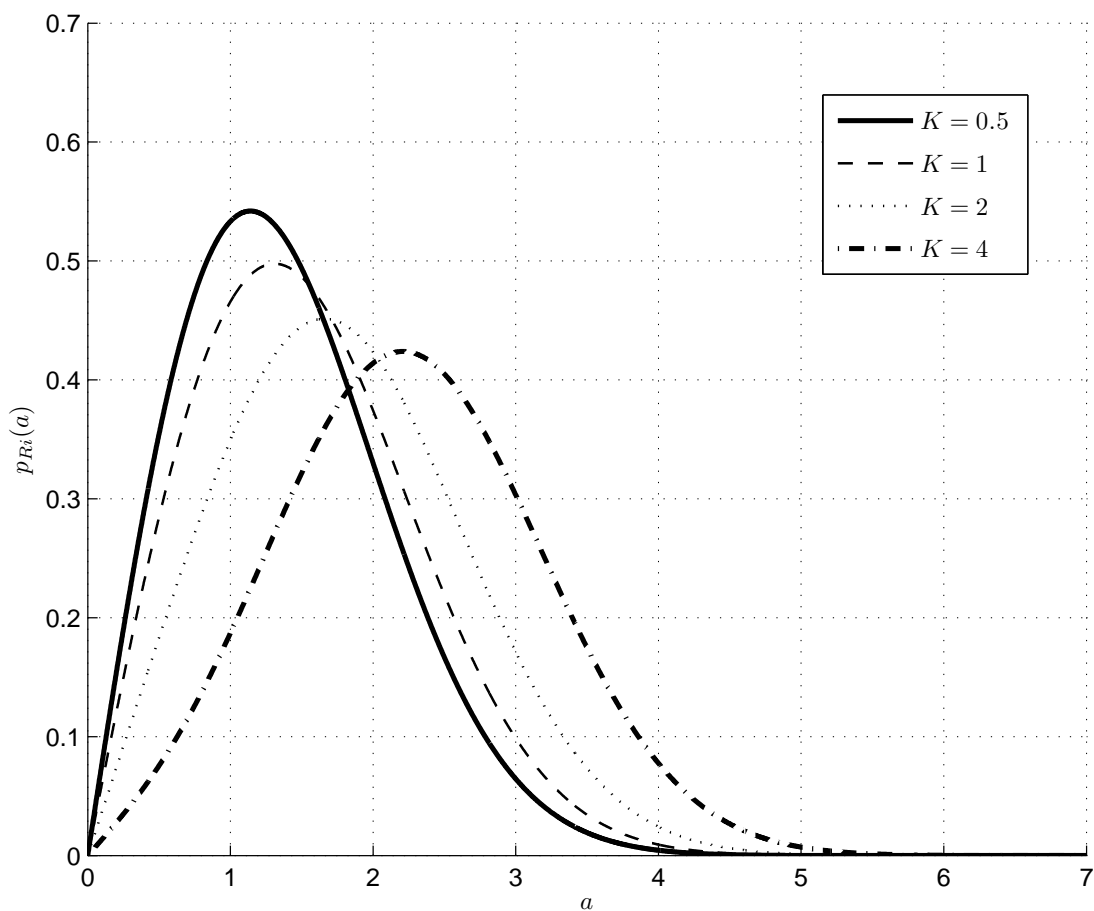


Figura 3.2: Função densidade de probabilidade de Rician para diferentes valores de  $K$ .

A figura 3.2 mostra a função densidade de probabilidade Rician para diferentes valores de  $K$ , sendo que  $2\sigma_{Ri}^2 = 1$ . Na Figura 3.2 é possível observar que quanto maior o valor de  $K$  a distribuição tende a ficar semelhante a distribuição Gaussiana pois a componente da linha de visada direta é maior que a componente das ondas dispersas. Para menores valores de  $K$  a distribuição tende a ficar mais parecida com a distribuição Rayleigh, pois a componente das ondas dispersas (denominador da fração da definição de  $K$ ) são maiores que a componente da visada direta (numerador da fração da definição de  $K$ ) prejudicando de maneira mais severa o sinal, pois existe um maior percentual ondas dispersas na transmissão.

### 3.4 Desvanecimento Nakagami- $m$

A distribuição Nakagami- $m$  têm ganhado um crescente destaque na área de modelagem de sistemas sem fio. Esta distribuição foi primeiramente apresentada em [20], na década de 70. A distribuição Nakagami- $m$  é um modelo mais geral de distribuição e engloba diversos cenários, entre eles estão as distribuições de Rayleigh e Rician. Dados empíricos descritos a seguir mostram que diferentes sistemas de comunicação são melhor modelados utilizando desvanecimento Nakagami- $m$ :

1) A soma de vários sinais com desvanecimento Rayleigh independentes e identicamente distribuídos formam um sinal com distribuição do tipo Nakagami- $m$ ; este tipo de interferência é verificado em modelos de sistemas celular [23];

2) Resultados recentes mostram que dados empíricos são melhor modelados segundo o desvanecimento Nakagami- $m$  para diversos tipos de comunicações via rádio em ambientes *indoor* e *outdoor* [21]. Neste caso as medidas reais foram melhor modeladas utilizando a distribuição Nakagami- $m$  para dois valores de  $m$  ( $m = 0,8$  *outdoor* e  $m = 2,1$  *indoor*). As modelagens utilizando as distribuições Rayleigh e Rician não foram apropriadas para os casos em questão;

3) Estudos realizados a partir de medidas reais comprovam que dados recebidos de satélites GPS em canais satélite-*indoor* e satélite-*outdoor* são melhores modelados segundo a distribuição Nakagami- $m$  [22];

4) Sistemas CDMA possuem melhor ajuste de acordo com a distribuição Nakagami- $m$  segundo estudos realizados em [23]. Outra análise realizada em [24] para sistemas 3G WCDMA em ambientes urbanos mostrou resultados semelhantes.

Como vários cenários são na realidade melhor ajustados com a modelagem Nakagami- $m$ , as distribuições Rayleigh/Rician utilizadas de maneira equivocadas podem causar prejuízo no projeto de muitos sistemas de comunicações sem fio, aumentando o custo total do projeto. O projeto de códigos de correção de erros otimizados para determinado canal pode trazer melhora de desempenho, o que implica em menor potência de transmissão e economia de recursos.

Poucas referências foram encontradas na literatura relacionando códigos LDPC com o canal Nakagami- $m$ . O trabalho [25] mostra a utilização de um tipo específico de códigos LDPC associado à técnica HARQ e simulado em um ambiente Nakagami- $m$ . O trabalho [26] utiliza técnicas de combinação de diversidade associada a uma técnica de construção de códigos LDPC chamada de *Gaussian Approximation* ou GA, para encontrar códigos LDPC para canais com desvanecimento Nakagami- $m$ , o trabalho obteve bons resultados.

Como não foi encontrado na literatura nenhum projeto de códigos LDPC para o canal Nakagami- $m$  utilizando a técnica DE, este trabalho propõe a construção de códigos LDPC para este canal.

A função densidade de probabilidade da distribuição Nakagami- $m$  é dada por:

$$p_N(a) = \frac{2}{\Gamma(m)} \left( \frac{m}{2\sigma_{Naka}^2} \right)^m a^{2m-1} \exp\left(-\frac{ma^2}{2\sigma_{Naka}^2}\right), \quad (3.3)$$

onde  $\Gamma(m)$  representa a função Gamma [30],  $m$  é o fator de forma da função Nakagami- $m$ ,  $a$  é o fator de desvanecimento Nakagami- $m$  e o valor  $2\sigma_{Naka}^2$  é igual a  $E[a^2]$ . O fator  $m$  (fator de forma) indica a severidade do desvanecimento por múltiplos percursos (quanto menor o valor de  $m$ , mais severo o desvanecimento). O modelo de distribuição Nakagami- $m$  é uma generalização dos modelos descritos acima, a distribuição Rayleigh é um caso especial da distribuição Nakagami- $m$  para  $m = 1$ . Se  $m = 2$ , é equivalente ao caso Rayleigh com duas antenas receptoras e MRC (*Maximal Ratio Combining*, uma técnica de

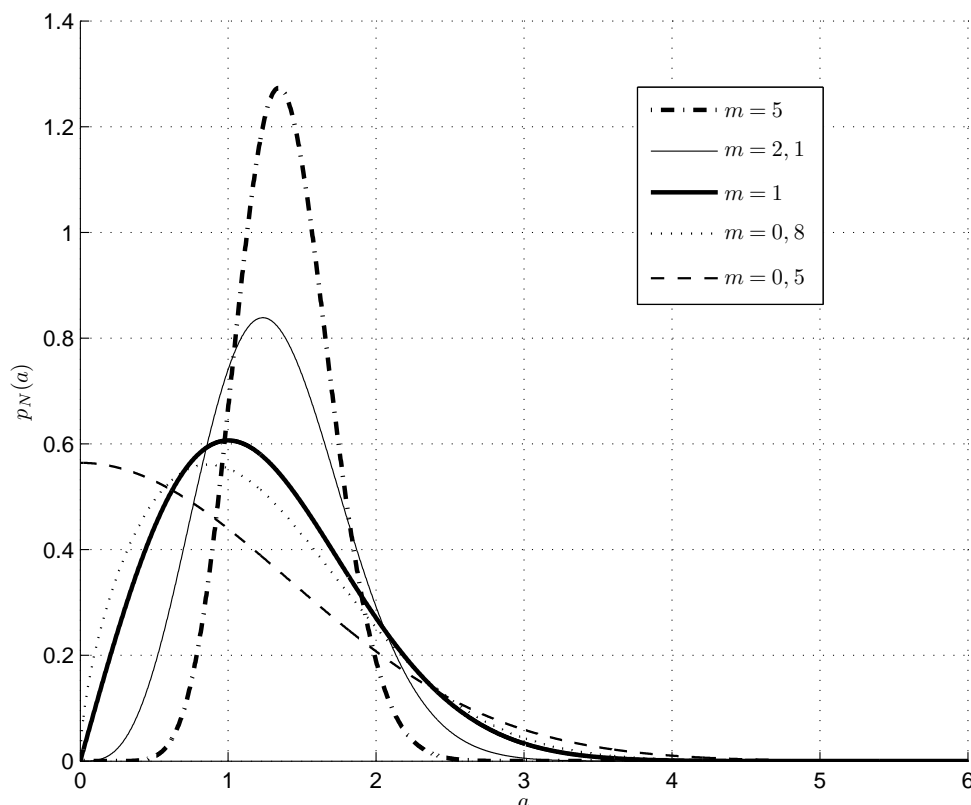


Figura 3.3: Função densidade de probabilidade Nakagami- $m$  para diferentes valores de  $m$  e  $2\sigma^2 = 1$ .

combinação de diversidade). A distribuição Rician é um caso particular da distribuição Nakagami- $m$  para valores grandes de  $m$ . A figura 3.3 mostra a função densidade de probabilidade Nakagami- $m$  para diferentes valores de  $m$  e  $2\sigma_{Naka}^2 = 1$ . De acordo com a Figura 3.3 é possível observar que quanto maior o valor de  $m$  maior a semelhança com a distribuição Rician. É possível notar que para  $m = 1$  a distribuição Nakagami- $m$  é idêntica a distribuição Rayleigh.

### 3.5 Geração de variáveis Nakagami- $m$

Para a simulação do canal com desvanecimento Nakagami- $m$  é necessário utilizar um método para a geração de variáveis aleatórias com distribuição Nakagami- $m$ . Existem diferentes métodos na literatura para a geração de variáveis aleatórias com distribuição Nakagami- $m$ , entre eles estão o método de rejeição proposto por Matthaiou e Laurenson [31], o método de Cao e Beaulieu [32], o método de Santos e Yacoub [33] e o método de

Zhang et al [34].

O método escolhido para a geração de variáveis aleatórias com distribuição Nakagami- $m$  para serem utilizadas na aplicação do desvanecimento do canal é o método da rejeição [31]. Este método é muito utilizado em situações onde a geração de variáveis aleatórias é uma tarefa complexa dada uma determinada pdf. A partir de uma outra pdf parecida com a original e de mais fácil implementação, variáveis aleatórias são geradas e selecionadas de acordo com um algoritmo de seleção. Este método consiste em criar uma função  $p_h(x)$  que satisfaz a desigualdade  $p_h(x) \geq p_N(x)$ , onde  $p_N(x)$  é definida em (3.3). A função  $p_h$  é:

$$p_h(x) = \frac{A\sqrt{\Omega}}{\Omega - B\sqrt{\Omega} + x^2} \geq p_N(x), \quad (3.4)$$

onde  $\Omega = E[x^2]$  e os valores de  $A$  e  $B$  são dados respectivamente por,

$$A = (2 - B) \frac{2m^m}{\Gamma(m)} e^{-m} \quad (3.5)$$

e

$$B = 2\sqrt{\frac{2m - 1}{2m}}. \quad (3.6)$$

Uma amostra aleatória de  $x$  é gerada através da equação:

$$x = \frac{\sqrt{\Omega}}{2} \left( \sqrt{4 - B^2} \tan \left( \frac{t\sqrt{4 - B^2}}{2A} \right) + B \right), \quad (3.7)$$

onde  $t$  é uma variável aleatória uniformemente distribuída no intervalo

$$\left[ \frac{2A \tan^{-1} \left( \frac{-B}{\sqrt{4 - B^2}} \right)}{\sqrt{4 - B^2}}, \frac{A\pi}{\sqrt{4 - B^2}} \right]. \quad (3.8)$$

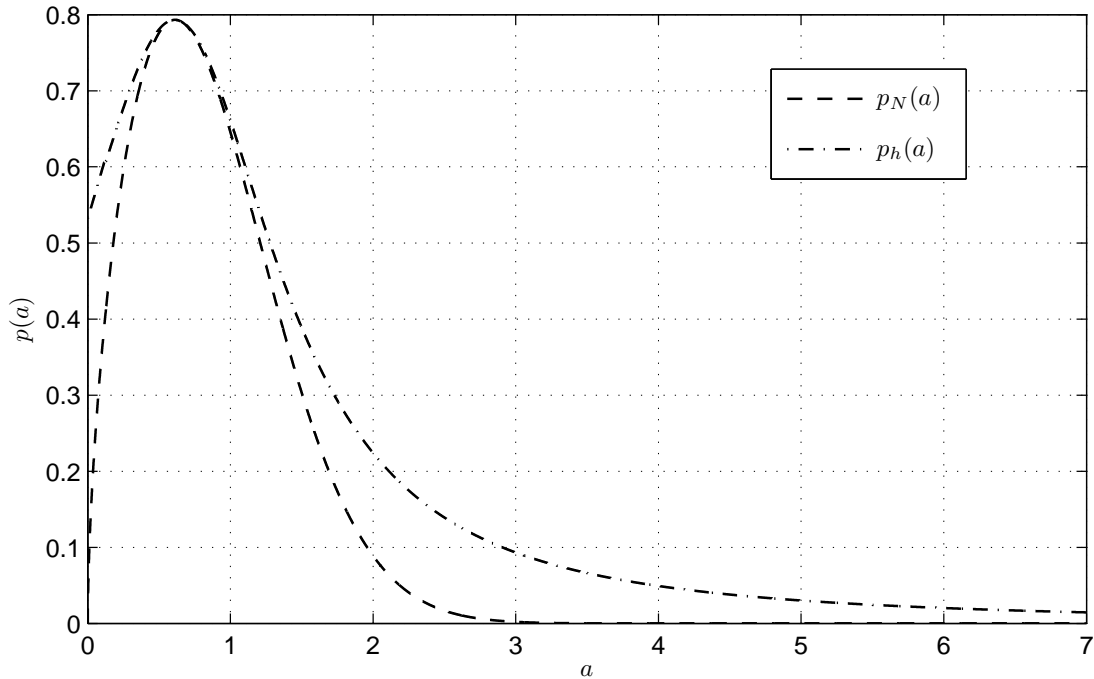


Figura 3.4: Função densidade de probabilidade  $p_h(x)$  e função densidade de probabilidade  $p_N(x)$  para  $m = 0,8$ .

As curvas correspondentes às funções  $p_h(x)$  e  $p_N(x)$  são comparadas na Figura 3.4 considerando  $m = 0,8$ . É possível observar que a função  $p_h(x)$  encobre a função  $p_N(x)$ .

Uma amostra  $x$  gerada é aceita ou rejeitada através do seguinte algoritmo “aceita-rejeita”[35]:

- 1) Gerar a variável aleatória  $x$  a partir da pdf  $p_h$ .
- 2) Gerar a variável aleatória  $u$  a partir da distribuição uniforme sobre o intervalo unitário.
- 3) Se  $u \leq \frac{p_N(x)}{p_h(x)}$ , então:
  - 3.a) tomar  $x$  como o valor desejado;
  - 3.b) caso contrário voltar à etapa 1).

A geração de variáveis Nakagami- $m$  para a simulação do canal terá como base o algoritmo de rejeição apresentado neste capítulo. Com a modelagem do canal Nakagami- $m$  também apresentada neste capítulo será possível utilizar a técnica DE para obter os códigos otimizados.

## CAPÍTULO 4

### TÉCNICA DE BUSCA DE CÓDIGOS

#### 4.1 Algoritmo DE

O algoritmo DE [12] determina qual o maior nível de ruído de canal no qual mensagens recebidas são possíveis de serem decodificadas livres de erros através do algoritmo *Belief Propagation*. Utilizando este algoritmo é possível encontrar um *ensemble* definido por um par de funções de distribuições de graus  $(\lambda, \rho)$  otimizado para um canal específico. O algoritmo utiliza o princípio do decodificador *Belief-Propagation* e seu nome refere-se à evolução das pdfs das mensagens que são trocadas entre os nós de maneira iterativa durante um número máximo de iterações  $l_{max}$ .

A cada iteração o algoritmo DE realiza duas etapas de cálculos: a primeira é o cálculo da pdf para a mensagem  $M_{i,j}$  (enviada de um nó de variável para um nó de verificação) denotada por  $p_l^{(v)}$ , onde o índice  $l$  denota o número da iteração; a segunda etapa de cálculo é a pdf para a mensagem  $E_{i,j}$  (enviada de um nó de verificação para um nó de variável) denotada por  $p_l^{(e)}$ .

Para o cálculo de cada pdf é assumido que todas as mensagens são independentes entre si, isto é, o canal é sem memória e nenhuma das probabilidades *a-priori* são correlacionadas. Também é presuposto que não existam ciclos no grafo de Tanner. É realizada apenas a análise sobre uma mensagem pois todas as pdfs das diferentes mensagens serão iguais entre si [2].

O cálculo de  $p_l^{(v)}$  é baseado em (2.14), mas ao invés de utilizar como mensagem a relação logarítmica entre bits de informação é realizada a análise da pdf do canal. De acordo com [35], [2] e [27] o somatório na segunda parte de (2.14) pode ser obtido através da convolução de todas as pdfs das mensagens provenientes dos nós de verificação e da pdf da mensagem original do canal, denotada por  $p_0$  [27]. Para cada canal analisado esta probabilidade é diferente. A equação:

$$p_i^{(v)} = p_0 * (p_{i-1}^{(c)})^{*(d-1)}, \quad (4.1)$$

é a pdf para a mensagem  $M_{j,i}$  [27] como foi mencionado anteriormente e este resultado pode ser estendido para o caso de códigos LDPC irregulares sofrendo apenas pequenas alterações. O operador  $*$  significa convolução e o operador  $*$  é a potência convolução (*convolution power*), ou seja um número convoluído com o mesmo  $n$ -vezes. Neste caso é utilizada a função  $\lambda(x)$  definida em (2.4) para reformular a equação (4.1) como [2]:

$$p_i^{(v)} = p_0 * \sum_{d=1}^{d_v} \lambda_d (p_{i-1}^{(c)})^{*(d-1)}. \quad (4.2)$$

Na sequência é realizada a análise de  $p_i^{(c)}$  baseado em (2.13) que é a função densidade de probabilidade da mensagem  $m^{(c)}$  enviada por um nó de verificação para um nó de variável. A função densidade de probabilidade do nó de verificação na iteração  $l$  [27] é dada por:

$$p_i^{(c)} = 2 \tanh^{-1} \left[ \sum_{d=1}^{d_c} \rho_d \left( \tanh \left( \frac{p_{i-1}^{(v)}}{2} \right) \right)^{*(d-1)} \right]. \quad (4.3)$$

De acordo com [12][14] a probabilidade de que determinado bit esteja errado é denotada por  $P_e^l$ , que é a fração de mensagens incorretas depois de  $l$  iterações e também pode ser compreendida como a probabilidade da pdf  $p_i^{(v)}$  ser menor que zero, de acordo com:

$$P_e^l = \int_{-\infty}^0 p_i^{(v)}(x) dx. \quad (4.4)$$

Para um dado valor de SNR é possível executar o algoritmo DE até encontrar o valor de probabilidade de erro  $P_e^l = 0$  ou até ele atingir uma probabilidade finita de erro mesmo que as iterações aumentem.

Como foi citado anteriormente SNR é a relação Sinal-Ruído, onde  $SNR = E_b/N_0$ , sendo que  $E_b$  é a energia por bit e  $N_0$  é a densidade espectral de potência de ruído. Uma definição importante que será muito utilizada na sequência é o valor  $\sigma_R$ , desvio padrão do ruído, que é definido como  $\sigma_R = \sqrt{(\frac{1}{2R} \frac{E_b}{N_0})}$ . O valor de limiar é denotado por  $E_b/N_0^*$  em dB e significa o maior valor entre todos os níveis de ruído analisados onde  $\lim_{l \rightarrow \infty} P_e^l = 0$ , de acordo com [16].

A técnica DE necessita que duas propriedades sejam atendidas para seu devido funcionamento: Simetria e Estabilidade.

- Simetria : Se o canal é simétrico então as relações logarítmicas da saída do decodificador serão simétricas também. Através de distribuições simétricas é possível mostrar que o desempenho do decodificador não depende da palavra transmitida e portanto pode-se utilizar a palavra toda-zero para a entrada do algoritmo. Além disso as densidades das mensagens sempre convergem quando o número de iterações tende ao infinito se a propriedade de simetria é provada e se o grafo de Tanner não possuir ciclos. Para ser atendida a propriedade de simetria deve-se satisfazer [14]:

$$p(x) = p(-x) \exp(x), \quad (4.5)$$

onde  $p(x)$  é uma pdf.

- Estabilidade: Um código aleatório construído a partir das equações de distribuição de um determinado *ensemble* de códigos terá um desempenho muito próximo da capacidade de canal calculada para o *ensemble*. A prova desta propriedade e da convergência dos códigos que cumprem com ela é mostrada em [14], onde é mostrado que a fração de mensagens incorretas converge para zero se:

$$\lambda'(0)\rho'(1) < \exp(s), \quad (4.6)$$

onde  $\lambda'(0) = \lambda_2$  e  $\rho'(1) = 1$  conforme mostrado em [2] [14], portanto:

$$\lambda_2 < \exp(s). \quad (4.7)$$

A parte da direita de (4.7) também é conhecida por  $\lambda_2^*$ , o valor limite de garantia da condição de estabilidade. Para os códigos projetados neste trabalho sempre foi escolhido  $\lambda_2$  que cumprisse com a condição de estabilidade  $\lambda_2 < \lambda_2^*$ . O parâmetro  $s$  é definido em [14] como:

$$s = -\ln \left( 2 \int_0^\infty p(x) e^{-\frac{x}{2}} dx \right). \quad (4.8)$$

Desta maneira a propriedade da estabilidade define uma importante limitação na construção de códigos LDPC pois limita o valor de  $\lambda_2$ , este resultado serve como base para encontrar os demais valores de  $\lambda$ .

#### 4.1.1 Algoritmo de busca *Differential Evolution*

O algoritmo DE retorna apenas o limiar para um único *ensemble*. Para encontrar o *ensemble* ótimo é preciso realizar a busca entre os infinitos *ensembles*, o que torna a pesquisa praticamente inviável. Portanto é utilizada uma técnica de busca rápida e que considere um espaço finito de amostras. Através de um algoritmo de busca inteligente é possível encontrar quais os melhores pares de distribuição  $(\lambda, \rho)$  no sentido de proporcionar o melhor desempenho de correção para um determinado canal. O algoritmo de busca mais utilizado em associação com o algoritmo DE é chamado *Differential Evolution* [13].

O algoritmo de busca define uma população de variáveis, no caso cada variável é um vetor contendo o par de distribuição  $(\lambda, \rho)$ . Para cada variável é executado o algoritmo DE e segundo as definições do algoritmo é realizado um processo de mutação e seleção. Este processo é executado por diversas iterações até ser encontrado um vetor otimizado para o canal em questão.

O algoritmo é muito custoso computacionalmente e na literatura são encontrados es-

tratégias para um processo mais eficiente [14][16] descrito a seguir. Em geral são definidas algumas restrições para o par de distribuição de graus, a primeira é a restrição do elemento de grau máximo  $\lambda_{max}$  e  $\rho_{max}$ . Outra restrição é feita sobre a quantidade de elementos das funções de distribuição de graus, apenas alguns são escolhidos e são chamados de elementos livres do par de distribuição. Para o presente trabalho foram escolhidos os elementos livres  $\lambda_2, \lambda_3, \lambda_{max}$  e outros poucos elementos  $\lambda$  de acordo com a literatura [14] [16]; foram escolhidos também  $\rho_{max}$  e apenas mais um elemento  $\rho$ . Aplicando estas restrições no algoritmo de busca, o espaço de busca é reduzido e conseqüentemente o tempo de análise de busca. Estas restrições não causam impactos significativos na qualidade e desempenho dos códigos obtidos, de acordo com [14].

Para que seja possível trabalhar com os valores dos graus de distribuição dos grafos é necessário remover as dependências entre os componentes do par de distribuição de graus, de acordo com [16] as equações  $(\lambda, \rho)$  satisfazem as restrições:

$$\sum_i \frac{\rho_i}{i} = K \sum_i \frac{\lambda_i}{i}, \quad (4.9)$$

onde  $K = 1 - R$  e  $R$  é a taxa do código, além disso:

$$\lambda_2 = 1 - \sum_{i=3}^{dv_{max}} \lambda_i, \quad (4.10)$$

$$\rho_2 = 1 - \sum_{i=3}^{dc_{max}} \rho_i. \quad (4.11)$$

Utilizando (4.9), (4.10) e (4.11) é possível resolver para  $\lambda_{max}$ :

$$\lambda_{max} = \frac{\frac{1-K}{2} + \sum_{i=3}^{dc_{max}} \rho_i(1/i - 1/2) - K \sum_{i=3}^{dv_{max}} \lambda_i(1/i - 1/2)}{K \left( \frac{1}{dv_{lmax}} - \frac{1}{2} \right)}. \quad (4.12)$$

O algoritmo a seguir foi utilizado em [16] e em [13] e foi escolhido para o presente trabalho pela sua simplicidade. Ele é detalhado em quatro etapas:

**1) Inicialização:** É definida uma população de  $NP$  vetores, onde  $NP = 10L$  e  $L$  é o comprimento de cada vetor sendo que  $L = dv_{max} + dc_{max} - 5$  é a quantidade de elementos livres do par de distribuição  $(\lambda, \rho)$ . Cada vetor é constituído por elementos variáveis, desta forma:

$$\mathbf{t}_{i,\mathcal{G}} = \lambda_3, \dots, \lambda_{dv_{max}-1}, \rho_3, \dots, \rho_{dc_{max}}, \quad (4.13)$$

onde o índice  $\mathcal{G}$  significa o número da geração do algoritmo (para a primeira geração  $\mathcal{G} = 0$ ) e  $i$  é o índice do vetor entre a população,  $i = 1, \dots, NP - 1$ .

O algoritmo é iniciado com um nível de SNR inicial. Na primeira geração os vetores possuem valores aleatórios. Primeiramente é realizado o cálculo para satisfazer a propriedade de estabilidade, de acordo com (4.7). Achando o valor de  $\lambda_2$  para cada um dos vetores é possível determinar os demais elementos  $\lambda$  e  $\rho$  de cada vetor seguindo as restrições impostas de acordo com as equações (4.10), (4.11) e (4.12). Desta maneira é estabelecida uma população de vários vetores que obedecem às restrições mencionadas e cada um deles será analisado em paralelo. Para cada vetor  $\mathbf{t}_{i,\mathcal{G}}$  o algoritmo DE é rodado para uma certa quantidade de iterações, por exemplo 1000 [16] [14]. O valor de erro calculado por (4.4) é armazenado ( $\mathbf{t}_{E_i,\mathcal{G}}$ ) e o vetor que possui menor erro é denotado como  $\mathbf{t}_{best,\mathcal{G}}$ .

**2) Mutação:** Para a geração  $\mathcal{G} + 1$  são criados novos vetores (de acordo com o procedimento descrito em [13] [36]) cujo padrão de mutação segue:

$$\mathbf{t}'_{i,\mathcal{G}+1} = \mathbf{t}_{best,\mathcal{G}} + 0,5(\mathbf{t}_{u_1,\mathcal{G}} - \mathbf{t}_{u_2,\mathcal{G}} + \mathbf{t}_{u_3,\mathcal{G}} - \mathbf{t}_{u_4,\mathcal{G}}), \quad (4.14)$$

onde os valores de  $u_1, u_2, u_3$  e  $u_4$  são escolhidos aleatoriamente no intervalo  $[0, NP - 1]$ . Para cada novo vetor o algoritmo DE é rodado novamente (durante o número definido de iterações) e o valor de erro é armazenado em  $\mathbf{t}'_{E_i,\mathcal{G}+1}$ .

**3) Seleção:** São comparados os valores ( $\mathbf{t}_{E_i,\mathcal{G}}$ ) e  $\mathbf{t}'_{E_i,\mathcal{G}+1}$ . Se ( $\mathbf{t}_{E_i,\mathcal{G}}$ ) é maior que  $\mathbf{t}'_{E_i,\mathcal{G}+1}$  então o novo vetor  $\mathbf{t}_{i,\mathcal{G}+1}$  recebe o valor  $\mathbf{t}'_{i,\mathcal{G}+1}$ , caso contrário  $\mathbf{t}_{i,\mathcal{G}+1}$  recebe  $\mathbf{t}_{i,\mathcal{G}}$ .

**4) Critério de parada:** Se o erro residual ( $\mathbf{t}_{E_{best}, \mathcal{G}+1}$ ) de  $\mathbf{t}_{best, \mathcal{G}+1}$  não é zero o algoritmo volta ao passo 2, se o resultado for um valor muito pequeno ( $< 10^{-8}$ ) então aumentar o valor de SNR e voltar ao passo 1, caso contrário o algoritmo é encerrado e é encontrado o limiar para o código e o par de distribuições de graus  $\lambda(x)$  e  $\rho(x)$  para o canal em questão.

## 4.2 DE para canal AWGN e canais com desvanecimento

A seguir será mostrada a análise das pdfs das mensagens para três canais de comunicações diferentes: AWGN (conforme mostrado em [12]), Rayleigh (conforme mostrado em [16]) e Nakagami- $m$  utilizadas na execução do algoritmo DE. A análise inédita para canal com desvanecimento Nakagami- $m$  é evidenciada de forma detalhada. Além disso são provadas as condições de estabilidade e simetria para cada um dos canais.

### 4.2.1 Análise para canal AWGN

O símbolo binário  $x_i$  que será transmitido é mapeado como  $w_i = (1 - 2x_i)$  gerando um símbolo pertencente ao conjunto  $\{1, -1\}$ . O símbolo  $y$  amostrado no receptor possui função densidade de probabilidade dada por:

$$p(y|w) = \frac{1}{\sqrt{2\pi\sigma_R^2}} \exp\left(-\frac{(y-w)^2}{2\sigma_R^2}\right). \quad (4.15)$$

Se  $P_r(x=0) = P_r(x=1) = \frac{1}{2}$  então a mensagem inicial observada pelo canal (equação (2.15)) pode ser expressa da seguinte forma:

$$r_0 = \log \frac{P(x_i=0|y_i)}{P(x_i=1|y_i)} = \frac{2}{\sigma^2} y_i. \quad (4.16)$$

Supondo sendo enviada a palavra toda zero, então todos os símbolos  $w_i = 1$  e portanto (4.15) pode ser re-escrita como:

$$p_0(r_0) = \frac{\sigma_R}{2\sqrt{2\pi}} \exp\left(-\frac{\left(r_0 - \frac{2}{\sigma_R^2}\right)^2}{2\left(\frac{4}{\sigma_R^2}\right)}\right), \quad (4.17)$$

que representa uma densidade gaussiana com média  $2/\sigma_R^2$  e variância  $4/\sigma_R^2$ . O valor  $p_0(r_0)$  é utilizado na entrada do algoritmo DE.

Para o canal AWGN a simetria é provada em [12]. Ali também é realizada a prova da condição de estabilidade para o canal AWGN, onde é mostrado que para este canal (4.7) e (4.8) resultam em:

$$\lambda_2 < e^{\frac{1}{2\sigma_R^2}}. \quad (4.18)$$

Com o valor de  $\lambda_2$  e as equações de restrições (4.10), (4.11) e (4.12) é possível gerar *ensembles* que sejam aleatórios mas respeitem as restrições, durante a execução do algoritmo *Differential Evolution*.

### 4.2.2 Análise para canal Rayleigh

A variável aleatória que representa a saída do canal com desvanecimento possui uma pdf condicional semelhante ao caso do canal AWGN, entretanto existe a presença do fator de desvanecimento  $a$ , de modo que (4.15) é alterada para:

$$p(y|w,a) = \frac{1}{\sqrt{2\pi\sigma_R^2}} \exp\left(-\frac{(y-wa)^2}{2\sigma_R^2}\right). \quad (4.19)$$

Para a análise do canal em questão,  $a$  é o fator de desvanecimento Rayleigh com  $E[a^2] = 1$ . Deste modo a pdf do canal Rayleigh é:

$$p_R(a) = 2a \exp -a^2. \quad (4.20)$$

Neste caso, como a palavra transmitida tem todos os bits iguais a zero:

$$p_0(r_0|a) = \frac{\sigma_R}{2a\sqrt{2\pi}} \exp \left( -\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{8a^2}{\sigma_R^2}\right)} \right). \quad (4.21)$$

Quando o valor médio de  $a$  é usado a função de densidade condicional de  $r_0$  pode ser escrita de acordo com:

$$p_0(r_0) = \int_0^\infty \frac{\sigma_R}{2a\sqrt{2\pi}} \exp \left( -\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{8a^2}{\sigma_R^2}\right)} \right) p_R(a) da, \quad (4.22)$$

substituindo (4.20) em (4.22) resulta em:

$$p_0(r_0) = \int_0^\infty \frac{\sigma_R}{\sqrt{2\pi}} \exp \left( -\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{8a^2}{\sigma_R^2}\right)} \right) \exp(-a^2) da, \quad (4.23)$$

rearranjando de acordo com [16]:

$$p_0(r_0) = \frac{\sigma_R}{\sqrt{2\pi}} \exp \left( -\frac{-r_0 \left(\sqrt{2\sigma_R^2 + 1} - 1\right)}{2} \right) \times \int_0^\infty \exp \left( -\frac{\left(\frac{\sigma_R^2}{2a} r_0 - a\sqrt{2\sigma_R^2 + 1}\right)^2}{2\sigma_R^2} \right) da. \quad (4.24)$$

A equação (4.24) é utilizada como entrada para o algoritmo DE específico para o canal Rayleigh.

Da mesma forma que para o canal AWGN, em [16] é realizada a prova da condição de simetria para o canal com desvanecimento Rayleigh.

Também em [16] é provada a condição de estabilidade para o canal com desvanecimento Rayleigh. Assim é mostrado que para este canal (4.7) e (4.8) resultam em:

$$\lambda_2 < 1 + \frac{1}{2\sigma_R^2}. \quad (4.25)$$

Da mesma forma que para o caso AWGN é possível gerar *ensembles* que sejam aleatórios mas repitem as restrições durante a execução do algoritmo *Differential Evolution*, através do valor de  $\lambda_2$  e as equações de restrições (4.10), (4.11) e (4.12).

### 4.2.3 Análise para canal Nakagami- $m$

Conforme mencionado anteriormente a análise das pdfs das mensagens para algoritmo DE no canal Nakagami- $m$  ainda não havia sido documentado na literatura. A seguir é realizada a análise da pdf que serve como entrada no algoritmo DE. Para o cálculo da pdf da mensagem inicial  $p_0(r_0)$  para o canal com desvanecimento Nakagami- $m$ , é realizado o mesmo procedimento realizado para os canais AWGN e Rayleigh, tal que:

$$p(y|w,a) = \frac{1}{\sqrt{2\pi\sigma_R^2}} \exp\left(-\frac{(y-wa)^2}{2\sigma_R^2}\right). \quad (4.26)$$

Neste caso  $a$  é o fator de desvanecimento Nakagami- $m$ , com  $E[a^2] = 1$ . A pdf para  $a$  é dada por:

$$p_N(a) = \frac{2}{\Gamma(m)} (m)^m a^{2m-1} \exp(-ma^2). \quad (4.27)$$

Também é suposta a transmissão da palavra toda-zero, portanto,  $w_i = 1$  em (4.22). Aplicando (4.27) em (4.22) obtemos:

$$p_0(r_0) = \int_0^\infty \frac{\sigma_R}{a\sqrt{2\pi}} \exp\left(-\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{8a^2}{\sigma_R^2}\right)}\right) \frac{1}{\Gamma(m)} (m)^m a^{2m-1} \exp(-ma^2) da, \quad (4.28)$$

onde (4.28) é a pdf utilizada no algoritmo DE para o canal Nakagami- $m$ .

#### 4.2.4 Simetria e estabilidade para canal Nakagami- $m$

Também são realizadas as provas de simetria e estabilidade, comprovando a validade da técnica para o canal Nakagami- $m$ . A condição de simetria é provada através de (4.5), portanto:

$$p_0(-r_0)exp(r_0) = \int_0^\infty \frac{\sigma_R}{a\sqrt{2\pi}} \exp\left(-\frac{\left(-r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{4a^2}{\sigma_R^2}\right)}\right) \times \frac{1}{\Gamma(m)} (m)^m a^{2m-1} \exp(-ma^2) \exp(r_0) da, \quad (4.29)$$

expandindo:

$$p_0(-r_0)exp(r_0) = \int_0^\infty \frac{\sigma_R}{a\sqrt{2\pi}} \exp\left(\frac{-r_0^2 - \frac{4a^4 r_0}{\sigma_R^2} - \frac{4a^4}{\sigma_R^2} + \frac{8a^4 r_0}{\sigma_R^2}}{\left(\frac{4a^2}{\sigma_R^2}\right)}\right) \times \frac{1}{\Gamma(m)} (m)^m a^{2m-1} \exp(-ma^2) da, \quad (4.30)$$

simplificando, chega-se a pdf da mensagem original para o canal Nakagami- $m$  comprovando a validade da condição de simetria:

$$p_0(-r_0)exp(r_0) = \int_0^\infty \frac{\sigma_R}{a\sqrt{2\pi}} \exp\left(-\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{4a^2}{\sigma_R^2}\right)}\right) \times \frac{1}{\Gamma(m)} (m)^m a^{2m-1} \exp(-ma^2) da. \quad (4.31)$$

A prova da condição de simetria ainda não havia sido publicada, este resultado inédito é extremamente importante pois mostra que existem *ensembles* otimizados para o canal

em questão.

A condição de estabilidade é obtida a partir de (4.7) e (4.8), tal que:

$$e^{-s} = 2 \int_0^\infty \int_0^\infty \frac{\sigma_R}{a\sqrt{2\pi}} \exp\left(-\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{4a^2}{\sigma_R^2}\right)}\right) \times \frac{1}{\Gamma(m)} (m)^m a^{2m-1} \exp(-ma^2) da e^{-\frac{r_0}{2}} dr_0. \quad (4.32)$$

Rearranjando (4.32):

$$e^{-s} = 2 \int_0^\infty \exp(-ma^2) \frac{(m)^m a^{2m-1}}{\Gamma(m)} \frac{\sigma_R}{\sqrt{2\pi}} da \int_0^\infty \exp\left(-\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{4a^2}{\sigma_R^2}\right)}\right) e^{-\frac{r_0}{2}} dr_0. \quad (4.33)$$

Resolvendo a primeira integral:

$$e^{-s} = \int_0^\infty \exp(-ma^2) \frac{(m)^m a^{2m-1}}{\Gamma(m)} \frac{\sigma_R}{\sqrt{2\pi}} da \int_0^\infty \exp\left(-\frac{\left(r_0 - \frac{2a^2}{\sigma_R^2}\right)^2}{\left(\frac{4a^2}{\sigma_R^2}\right)}\right) e^{-\frac{r_0}{2}} dr_0. \quad (4.34)$$

Simplificando chega-se finalmente em:

$$e^{-s} = \frac{(m)^m}{\Gamma(m)} \int_0^\infty 2a^{2m-1} \exp\left(-\left(m + \frac{1}{2\sigma_R^2}\right)a^2\right) da. \quad (4.35)$$

Resolvendo a integral:

$$e^{-s} = \frac{m^m}{\left(\frac{1}{2\sigma_R^2} + m\right)^m}, \quad (4.36)$$

obtendo por fim a condição de estabilidade para o canal Nakagami- $m$ :

$$\lambda_2 < \frac{\left(\frac{1}{2\sigma_R^2} + m\right)^m}{m^m}. \quad (4.37)$$

A equação (4.37) ainda não havia sido publicada em nenhum trabalho. Essa equação inédita é muito importante para a construção de códigos para o canal Nakagami- $m$ . Com o valor de  $\lambda_2$  e as equações de restrições (4.10), (4.11) e (4.12) é possível gerar *ensembles* que sejam aleatórios mas repitem as restrições, durante a execução do algoritmo *Differential Evolution*.

## CAPÍTULO 5

### IMPLEMENTAÇÃO E RESULTADOS OBTIDOS

#### 5.1 Implementação

Neste trabalho foram realizados diversos projetos de códigos para o canal Nakagami- $m$ . Cada projeto é dividido em duas partes: uma etapa de busca e uma etapa de testes. Na etapa de busca os cálculos teóricos e a técnica DE foram utilizadas para encontrar as distribuições de graus otimizadas para o canal Nakagami- $m$ . Na etapa de testes foram realizadas simulações computacionais para a análise de desempenho dos códigos obtidos.

Cada uma das etapas do projeto de um código LDPC otimizado para o canal Nakagami- $m$  pode ser subdividido em mais partes, cada parte é mostrada através de um bloco no fluxograma da Figura 5.1. A primeira etapa é subdividida em três partes descritas a seguir:

##### *Etapa de Busca*

1) Realização da análise teórica das pdfs das mensagens e a prova das propriedades de simetria e estabilidade como mostrado na Seção 4.2. Com a equação da pdf da mensagem é possível executar o algoritmo DE e com a equação de  $\lambda_2$  é possível executar o algoritmo *Differential Evolution*. O software *Mathematica* foi utilizado no desenvolvimento das provas teóricas.

2) Com os dados obtidos anteriormente foi realizada a execução do algoritmo DE associado ao algoritmo de busca *Differential Evolution*. Desta forma é possível encontrar o valor de limiar para o canal e também um *ensemble* otimizado definido por um par de distribuição de graus. Os algoritmos são executados de acordo com as descrições do Capítulo 4. Neste trabalho foi utilizado o software MATLAB para o desenvolvimento destes dois algoritmos.

3) Através do par de distribuição de graus otimizado obtido para o canal foi realizada a próxima etapa do projeto que é a construção da matriz  $H$  utilizando o algoritmo PEG.

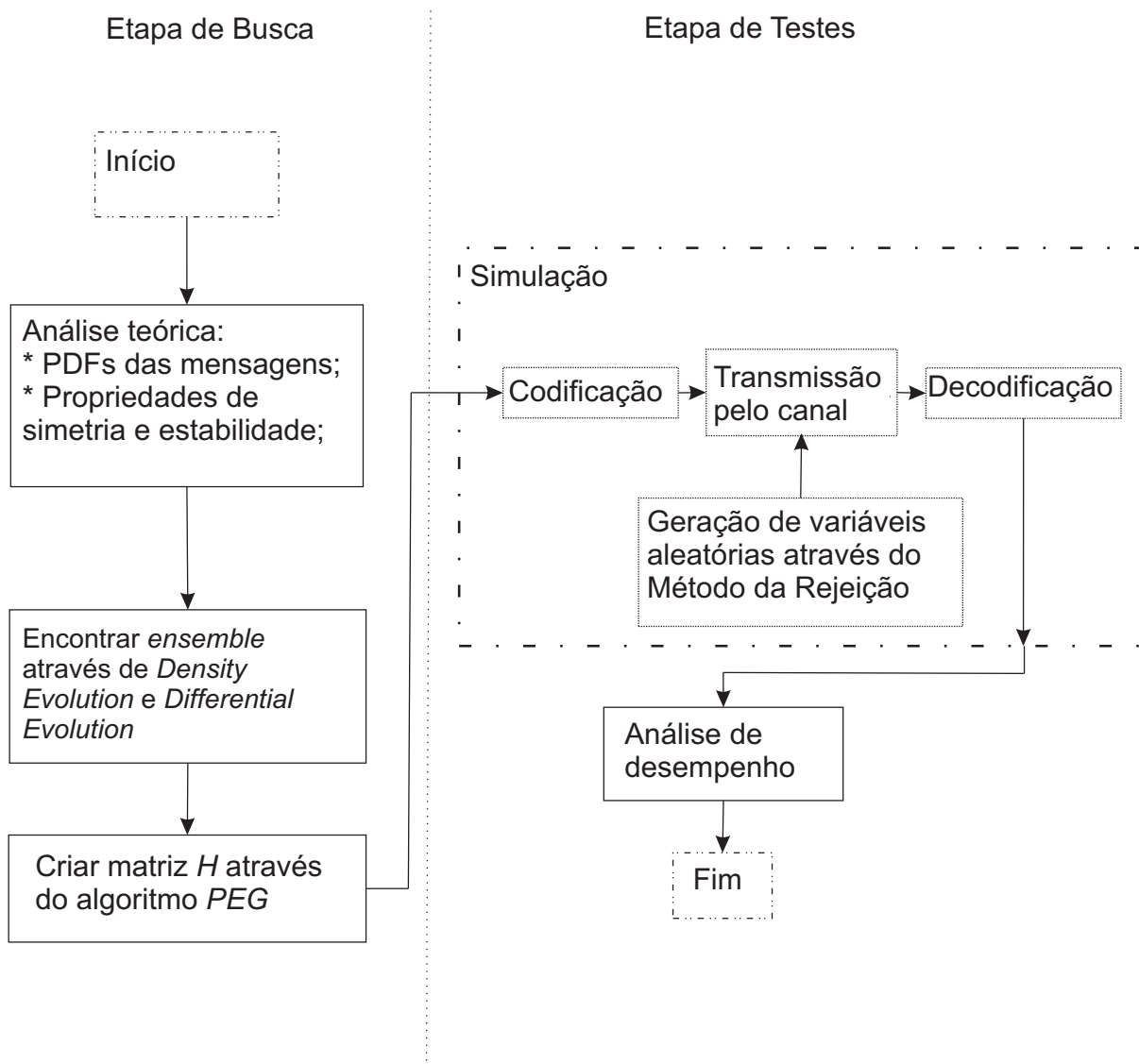


Figura 5.1: Fluxograma de etapas do projeto para a criação de um código LDPC otimizado para canal Nakagami- $m$

Neste trabalho foi utilizado o software para a construção PEG escrito em linguagem C e encontrado em [37].

### *Etapa de Testes*

Com a matriz otimizada  $H$  são realizadas simulações computacionais. A etapa de testes é subdividida em quatro etapas: codificação, transmissão, geração de variáveis aleatórias Nakagami- $m$  e decodificação.

1) Primeiramente é realizada a codificação de mensagens simuladas. Estas mensagens são codificadas de acordo com a Seção 2.2. A modulação utilizada é BPSK.

2) O desvanecimento é simulado de acordo com o Método da Rejeição, Seção 3.5.

O desvanecimento altera as mensagens codificadas simulando um canal de comunicação com desvanecimento Nakagami- $m$ , além disso é adicionado ruído térmico Gaussiano. A geração das variáveis aleatórias para a distribuição Nakagami- $m$  foi desenvolvida através de linguagem C e em MATLAB.

3) É realizada a demodulação e em seguida a decodificação das mensagens corrompidas utilizando o algoritmo *Belief-Propagation*, de acordo com a Seção 2.3.

4) Depois da decodificação das mensagens recebidas utilizando os softwares de codificação/decodificação foi realizada uma análise da taxa de erro de bit (em inglês, Bit Error Rate ou ainda BER) para cada simulação. A decodificação *Belief Propagation* contou com 400 iterações para cada palavra.

Neste projeto o software que simula a codificação, transmissão, decodificação e realiza o cálculo de BER, é encontrado em [38] e [39], e foi escrito em linguagem C. A menor BER foi definida como  $10^{-5}$  que é válido para simulações de comunicações sem fio Também foi criado um software de conversão do formato de matriz esparsa para o tipo de arquivo utilizado nos softwares citados.

## 5.2 Resultados Obtidos

Foram realizados três projetos de códigos LDPC com taxa  $R = \frac{1}{2}$  para canais com desvanecimento Nakagami- $m$  com  $m = \{0,8; 2,1; 4\}$ . Os valores de  $m = 0,8$  e  $m = 2,1$  foram utilizados pois foram valores que melhor se ajustaram em um exemplo de transmissão real, conforme descrito em [21].

Cada um dos projetos realizados foi comparado com outras simulações de outros projetos já encontrados na literatura [16], porém otimizados para canais Rayleigh e AWGN. Estes códigos projetados para canais Rayleigh e AWGN foram simulados em cenários Nakagami- $m$ , e comparados com os novos códigos projetados neste trabalho. Desta forma será possível verificar que os três novos códigos projetados para o canal Nakagami- $m$  possuem desempenho de correção superior aos projetos realizados para os canais Rayleigh e AWGN, quando usados em ambiente Nakagami- $m$ , como será mostrado a seguir.

A análise de cada código é realizada em termos de BER versus  $E_b/N_0$  (SNR). Para

cada valor de  $E_b/N_0$  foram realizadas simulações de transmissão, onde 10000 palavras de tamanho 1536 bits foram codificadas gerando mensagens de tamanho 3072, pois a taxa de cada código é  $R = 1/2$ . Este é o mesmo tamanho de bloco utilizado em [16] e [17], outros tamanho foram testados mas este tamanho foi escolhido para fins de comparações com os demais trabalhos já publicados.

Conforme definido na Seção 4.1.1 foram realizadas restrições sobre os elementos de grau máximo  $\lambda_{max}$  e  $\rho_{max}$ . Para o projeto de códigos neste trabalho foram definidos  $\lambda_{max} = 20$  e  $\rho_{max} = 6$ , conforme [16]. Outra restrição é feita sobre a quantidade de elementos das funções de distribuição de graus. Neste caso foram escolhidos os elementos  $\lambda_2, \lambda_3, \lambda_4, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_{19}$  e  $\lambda_{20}$ . Estas escolhas não foram feitas por acaso, nos trabalhos de [12], [14], [16] e [13] foram escolhidos estes valores, com os quais os melhores resultados foram obtidos para as otimizações. Os valores de  $\lambda_2$  para cada projeto são encontrados através dos cálculos de estabilidade, com o valor encontrado é possível utilizar as equações de restrições de acordo com a Seção 4.1.1 limitando os demais valores de  $\lambda$  e  $\rho$ . Conforme já foi mencionado existe a possibilidade do algoritmo PEG retornar diferentes matrizes  $H$  para um mesmo par de distribuição, entretanto todas possuem desempenho muito similares.

### 5.2.1 Resultados para $m = 0,8$

O primeiro projeto realizado foi a construção de um código otimizado para o canal Nakagami- $m$  com  $m = 0,8$ , de acordo com [21] um ambiente *outdoor* de comunicação sem-fio. Depois do procedimento realizado de acordo com as seções anteriores chegou-se na distribuição de graus mostrada na primeira coluna da Tabela 5.1.

A Figura 5.2 mostra o resultado da simulação realizada para o novo código em questão. Nesta figura é possível notar, por exemplo, que o código para canal Nakagami- $m$ , com  $m = 0,8$ , garante uma BER de aproximadamente  $4,5 \times 10^{-5}$  para 3 dB de SNR. Por sua vez, o código projetado para canal Rayleigh encontrado em [16] cuja distribuição se encontra na Tabela 5.2 possui desempenho de correção inferior ao código projetado para Nakagami- $m$ . O código projetado para canal Rayleigh garante uma BER de aproximadamente  $1,7 \times 10^{-4}$

Tabela 5.1: Funções de distribuição de grau para canais Nakagami- $m$  com  $m = 0,8$ ,  $m = 2,1$  e  $m = 4$

Elementos ( $\lambda, \rho$ )	$m = 0,8$	$m = 2,1$	$m = 4$
$\lambda_2$	0,2223	0,2110	0,2109
$\lambda_3$	0,1997	0,1951	0,2483
$\lambda_4$	0,1836	0,1671	0,0234
$\lambda_6$	0,0128	0,0334	0,1627
$\lambda_7$	0,0182	0,0054	0,0538
$\lambda_8$	0,0221	0,1401	0,0610
$\lambda_9$	0,0011	0,0023	0,0034
$\lambda_{19}$	0,1986	0,0729	0,0815
$\lambda_{20}$	0,1416	0,1729	0,1550
$\rho_8$	0,9434	0,9363	0,923
$\rho_9$	0,0566	0,0637	0,077

para 3dB de SNR. Na mesma figura é mostrada a BER obtida com um código projetado para o canal AWGN [37], cuja distribuição também é mostrada na Tabela 5.2. O código para canal AWGN tem desempenho de  $2,5 \times 10^{-3}$  para 3 dB de SNR.

A comparação entre códigos projetados para desvanecimento Nakagami- $m$ , Rayleigh e AWGN, em um canal Nakagami- $m$ , é útil para a comprovação do desempenho superior dos códigos projetados neste trabalho. Se um código é projetado apenas para canal com desvanecimento do tipo Rayleigh mas o ambiente de transmissão sem fio possui na verdade característica de desvanecimento do tipo Nakagami- $m$ , o código não será tão eficiente quanto um código projetado especificamente para aquele canal. A metodologia da comparação explora justamente este ganho de desempenho que um código projetado para Nakagami- $m$  tem quando o meio é bem modelado como Nakagami- $m$  também.

### 5.2.2 Resultados para $m = 2,1$

A Figura 5.3 mostra o resultado da simulação realizada para o novo código LDPC projetado para canal Nakagami- $m$  com  $m = 2,1$  (segunda coluna da Tabela 5.1). De acordo com [21] um ambiente *outdoor* de comunicação sem-fio. Uma comparação é feita, na mesma figura, com os mesmos códigos (para canais Rayleigh e AWGN) utilizados para comparação no primeiro caso, Tabela 5.2. Os três códigos foram simulados supondo transmissão

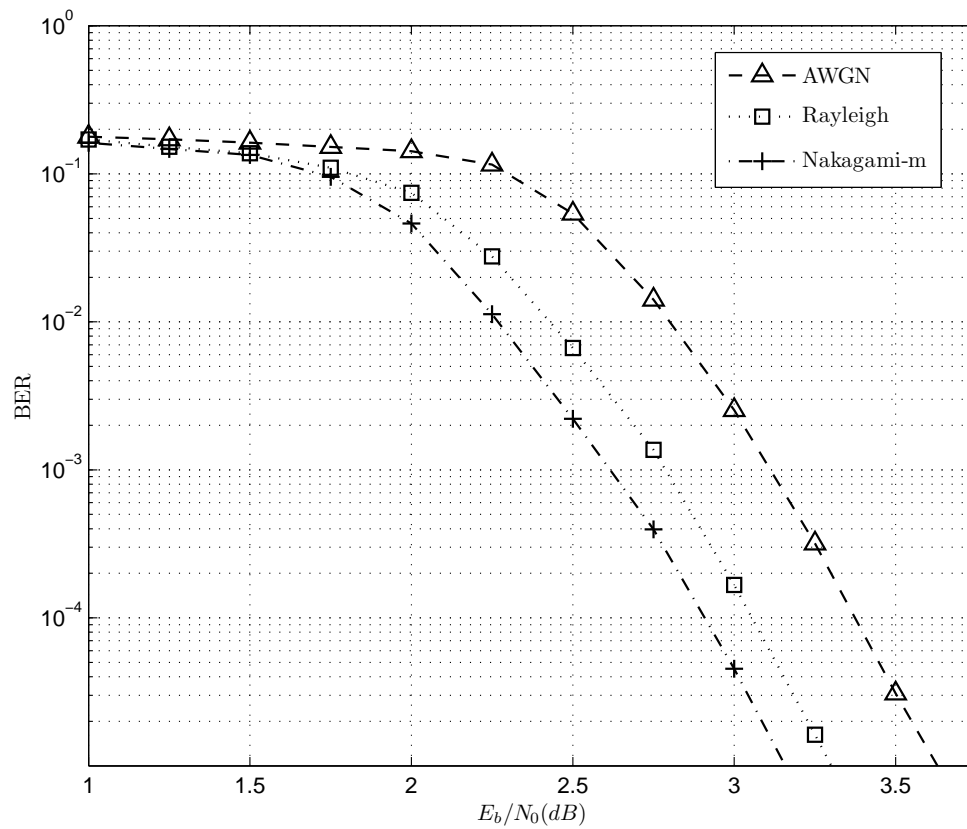


Figura 5.2: Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Nakagami- $m = 0,8$ .

Tabela 5.2: Funções de distribuição de grau para canais Rayleigh e AWGN

Elementos ( $\lambda, \rho$ )	Rayleigh	AWGN
$\lambda_2$	0,246544	0,457827
$\lambda_3$	0,230609	0,323775
$\lambda_4$	0,002045	0,0214226
$\lambda_6$	0,046487	0,0592851
$\lambda_7$	0,150161	0,0389015
$\lambda_8$	0,035344	0,0248109
$\lambda_9$	—	0,0884569
$\lambda_{19}$	0,004812	0,0176697
$\lambda_{20}$	0,283998	0,04746251
$\rho_7$	0,0361	0,000952
$\rho_8$	0,92876	0,951871
$\rho_9$	0,03514	0,047177

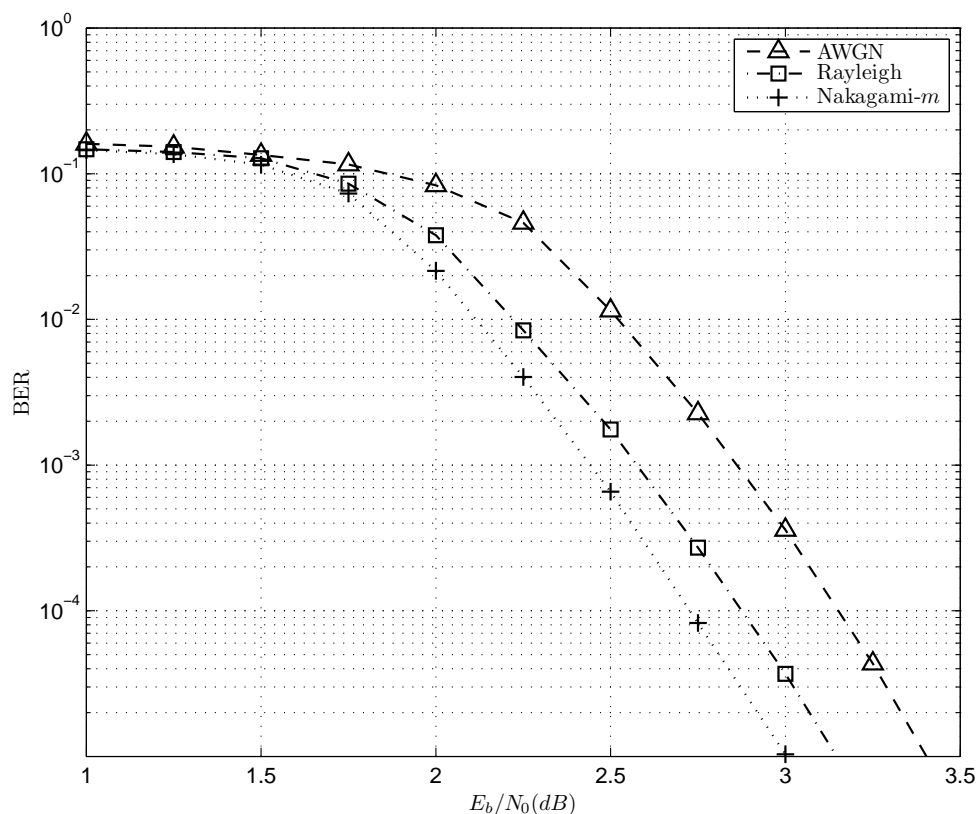


Figura 5.3: Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Nakagami- $m = 2,1$ .

sobre um canal de comunicação Nakagami- $m$  com fator  $m = 2,1$ .

A Figura 5.3 mostra que o novo código projetado para o canal Nakagami- $m$ , para  $m = 2,1$ , garante uma BER de aproximadamente  $1,03 \times 10^{-5}$  para 3 dB de SNR; o código projetado para canal Rayleigh possui desempenho de aproximadamente  $3,7 \times 10^{-5}$  para 3 dB de SNR; e o código para o canal AWGN tem desempenho de aproximadamente  $3,6 \times 10^{-4}$  para 3 dB de SNR. Novamente nota-se que o projeto específico para determinado canal possui ganhos em termos de economia de potência transmitida. Na Figura 5.3 pode-se observar que as três curvas evidenciam um desempenho superior se comparadas às três curvas da Figura 5.2, isto acontece pois o desvanecimento Nakagami- $m$  com fator  $m = 2,1$  é menos severo que o desvanecimento com  $m = 0,8$ . Quanto mais o valor do fator  $m$  cresce menos severo fica o desvanecimento, pois a linha de componente de visada direta (LOS) aumenta.

### 5.2.3 Resultados para $m = 4$

Pela figura 5.4 é possível observar o resultado da simulação realizada para o novo código LDPC projetado para canal Nakagami- $m$  com  $m = 4$  (terceira coluna da Tabela 5.1). Os mesmos códigos utilizados para comparação (canais Rayleigh e AWGN) para o primeiro e o segundo projeto foram simulados e comparados novamente. Outra vez os três códigos foram simulados supondo transmissão sobre um canal de comunicação Nakagami- $m$  com fator  $m = 4$ .

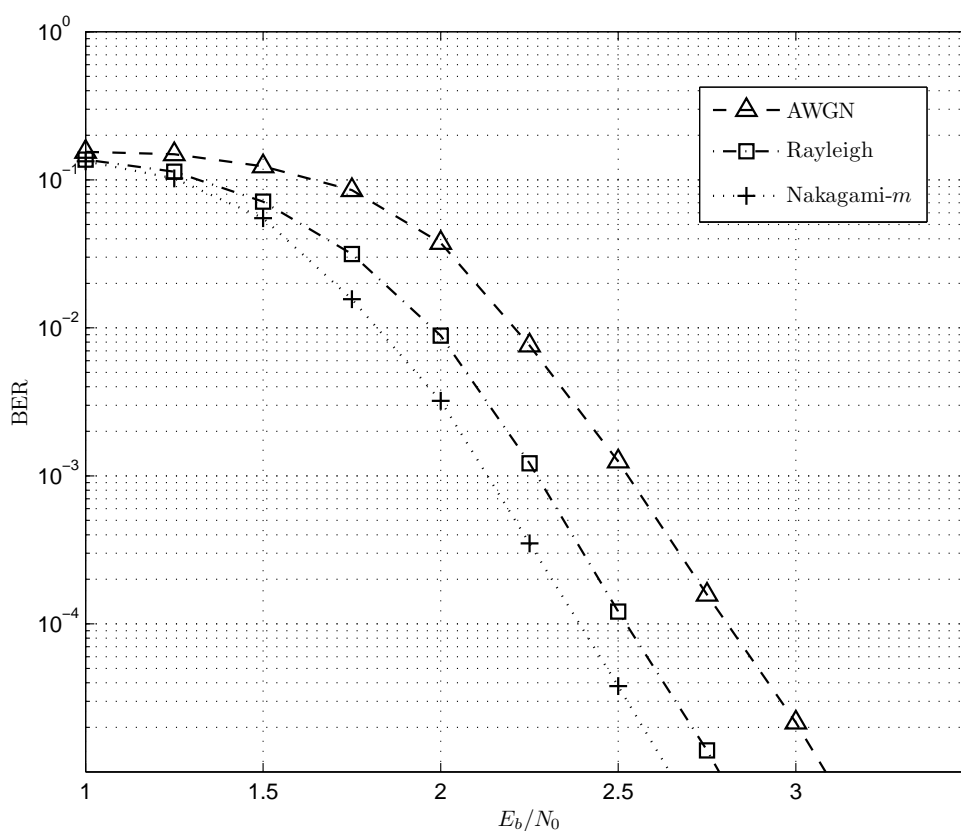


Figura 5.4: Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Nakagami- $m = 4$ .

A Figura 5.4 mostra que o novo código projetado para canal Nakagami- $m$ , para  $m = 4$ , garante uma BER de aproximadamente  $3,8 \times 10^{-5}$  para 2,5 dB de SNR; o código projetado para o canal Rayleigh consegue um desempenho de aproximadamente  $1,2 \times 10^{-4}$  para 2,5 dB de SNR; e o código para o canal AWGN tem desempenho de aproximadamente  $1,3 \times 10^{-3}$  para 2,5 dB de SNR. É possível observar na Figura 5.4 que o novo código

projetado para canal Nakagami- $m$  possui desempenho superior aos outros dois códigos. O desempenho mostrado nas três curvas da Figura 5.4 é melhor que o das curvas mostradas nas Figuras 5.2 e 5.3 por que o desvanecimento ficou ainda menos severo para este cenário.

## 5.2.4 Outros Resultados

A Figura 5.5 mostra a BER para o novo código projetado para canal Nakagami- $m$  ( $m = 0,8$ ), o código projetado para Rayleigh e o código projetado para canal AWGN, porém simulados em ambiente com desvanecimento Rayleigh. O projeto específico para canal Rayleigh possui o melhor desempenho entre os três códigos simulados. O código desenvolvido para canal AWGN possui o pior desempenho desta simulação.

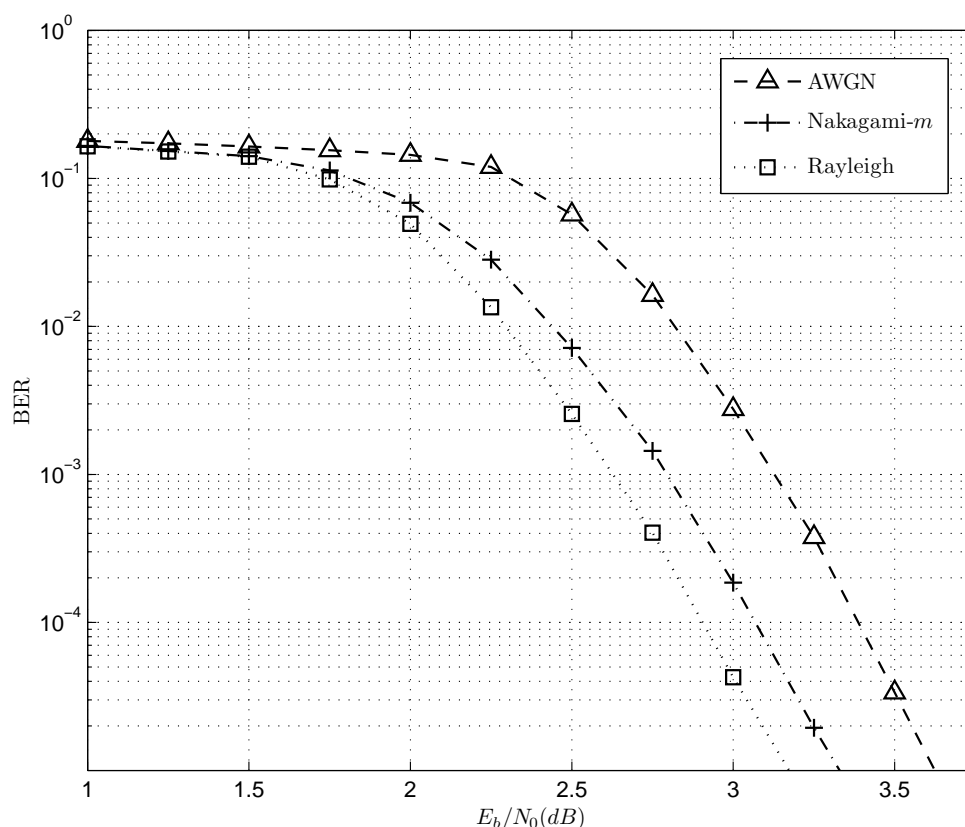


Figura 5.5: Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal Rayleigh.

A Figura 5.6 mostra a BER para os mesmos códigos usados na Figura 5.5, mas agora supondo ambiente AWGN. Através da Figura 5.6 é possível observar que o código proje-

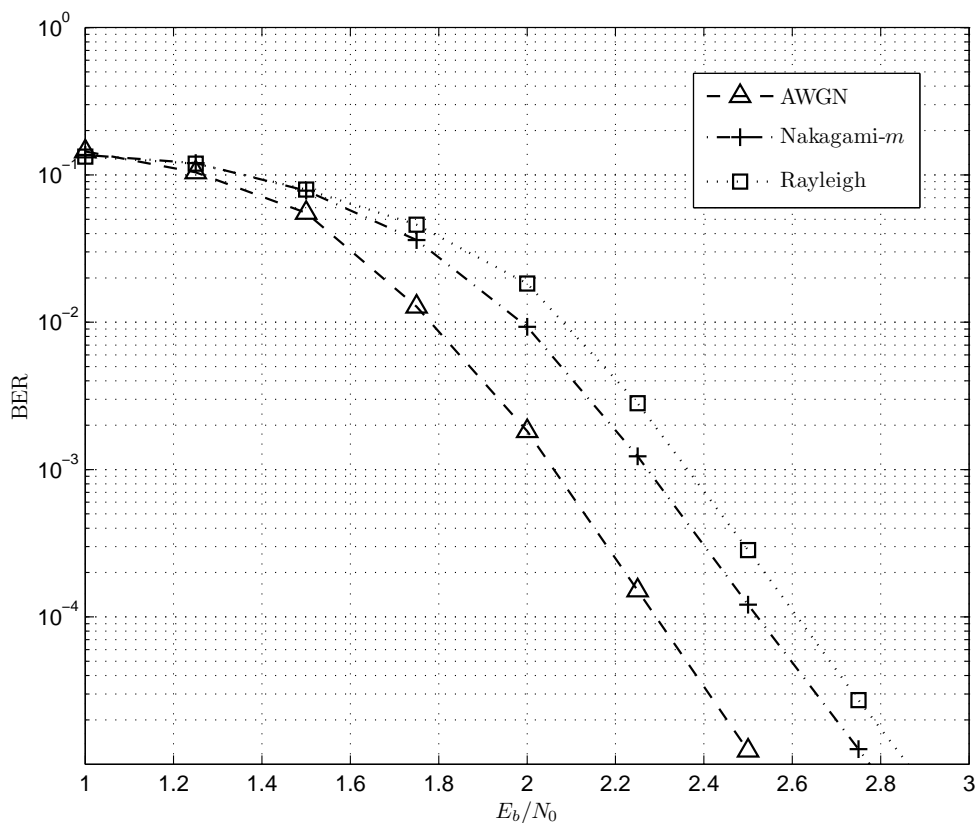


Figura 5.6: Desempenho de diversos códigos LDPC otimizados para diferentes canais de comunicação, utilizando transmissão sobre um canal AWGN.

tado especificamente para o canal AWGN possui o melhor desempenho entre os três.

Estes resultados, aliados aos das Figuras 5.2, 5.3, 5.4 e 5.5, mostram que o projeto específico de códigos para o canal desejado proporciona um ganho em termos de SNR, o que reflete em economia de potência transmitida para os agentes transmissores da informação. Os novos códigos, projetados para canal Nakagami- $m$ , obtêm cerca de 0,2 dB de ganho em relação ao projeto Rayleigh quando o canal é Nakagami- $m$ . Esta economia é muito significativa pois o ganho é multiplicado pelo número de transmissões realizadas em um sistema de telecomunicações. Além disso, o projeto de código para um canal específico é realizado apenas uma vez, valendo o esforço computacional e o tempo de busca de código.

## CAPÍTULO 6

### CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho teve como objetivo a construção de códigos LDPC para canais com desvanecimento Nakagami- $m$ . Pode-se verificar que existe um ganho real em termos de desempenho de erro ou de economia de potência transmitida.

Para chegar ao resultado obtido foi utilizada a técnica *Density Evolution* para que fossem encontrados os parâmetros de construção de *ensembles* otimizados para o canal em questão. As duas propriedades do algoritmo DE foram provadas para garantir que o par de distribuição de grau gerasse códigos otimizados para o canal e que o processo de decodificação convergisse quando o número de iterações tente ao infinito. Para realizar a busca de *ensembles* em paralelo foi utilizado o algoritmo *Differential Evolution*. Em seguida, a partir do *ensemble* encontrado, a matriz  $H$  do código foi construída utilizando o algoritmo *Progressive Edge Growth* (PEG).

O desempenho de cada novo código projetado foi analisado via simulação e pode ser verificado que estes códigos são superiores a outros códigos existentes na literatura. Os códigos projetados neste trabalho tiveram um ganho de cerca de 0,2 dB aproximadamente se comparados ao código projetado para canal Rayleigh, e quando simulados em ambiente Nakagami- $m$ .

Este trabalho pode servir como base para trabalhos futuros que envolvam comunicação cooperativa [40], neste caso o canal relay pode ser composto por diferentes canais com desvanecimento Nakagami- $m$  com fator  $m$  diferente para cada canal. Outra sugestão para trabalhos futuros é a análise e desenvolvimento de códigos LDPC para canais com desvanecimento utilizando o método EXIT charts. O método é uma simplificação da técnica *Density Evolution* e fornece resultados muito bons de forma mais rápida. Ainda não foram encontrados trabalhos relacionando o canal Nakagami- $m$  com o projeto de códigos LDPC utilizando a técnica EXIT Charts. Outra possibilidade de trabalho futuro

seria buscar otimizações nos algoritmos de busca, dois algoritmos sugeridos são o Enxame de Partículas (PSO) [41] e o Enxame de Partículas híbrido com *Differential Evolution* (DEPSO) [42].

## BIBLIOGRAFIA

- [1] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, julho de 1948.
- [2] S. J. Johnson. Iterative error correction: Turbo, low-density parity-check and repeat-accumulate codes. *Cambridge*, 2010.
- [3] D. Mackay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–430, março de 1999.
- [4] S. Chung; G. D. Forney Jr; T. J. Richardson e R. Urbanke. On the design of low-density parity-check codes within 0.0045 db of the shannon limit. *IEEE Communications Letters*, páginas 58–60, fevereiro de 2001.
- [5] R. G. Gallager. Low-density parity-check codes. *Monograph*.
- [6] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, setembro de 1981.
- [7] M. Luby; M. Mitzenmacher; A. Shokrollahi; D. Spielman e V. Stemann. Practical loss-resilient codes. *STOC '97 Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, páginas 150–159, 1997.
- [8] ETSI Standard: EN 300 744 V1.5.1. Digital video broadcasting (dvb); framing structure, channel coding and modulation for digital terrestrial television. 2004.
- [9] IEEE STD 802.3an. Ieee standard for local and metropolitan area networks part 3: Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications amendment 1: Physical layer and management parameters for 10 gb/s operation, type 10gbase-t.

- [10] IEEE STD 802.11N-2009. Ieee standard standard for local and metropolitan area networks part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, amendment 5: Enhancements for higher throughput. outubro de 2009.
- [11] IEEE STD 802.16E. Ieee standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems. maio de 2009.
- [12] T. J. Richardson e R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2).
- [13] A. Shokrollahi e R. Storn. Design of efficient erasure codes with differential evolution. *ISIT 2000*, páginas 5, junho de 2000.
- [14] T. J. Richardson; M. A. Shokrollahi e R. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2).
- [15] X. Hu; E. Eleftheriou e D. M. Arnold. Regular and irregular progressive edge-growth tanner graphs. *IEEE Transactions on Communications*.
- [16] J. Hou; P. Siegel e L. Milstein. Performance analysis and code optimization of low-density parity-check codes on rayleigh fading channels. *IEEE Journal on Selected Areas in Communications*, páginas 924–934, fevereiro de 2001.
- [17] J. Lin e W. Wu. Performance analysis of ldpc on rician fading channels. *Front. Electr. Electron. Eng. China*, páginas 120–124, 2006.
- [18] B. Sklar. Rayleigh fading channels in mobile digital communication systems part i: Characterization. *IEEE Communications Magazine*, 35(7):90–100, julho de 1997.
- [19] T. S. Rappaport. Wireless communications: Principles and practice. 2nd. *Prentice-Hall*, 2002.
- [20] M. Nakagami. The m-distribution, a general formula of intensity of rapid fading. *Statistical Methods in Radio Wave Propagation: Proceedings of a Symposium*, páginas 3–36, junho de 1958.

- [21] M. D. Yacoub. The k-u distribution and the n-u distribution. *IEEE Antennas and Propagation Magazine*, páginas 68–81, fevereiro de 2007.
- [22] A. Lakhzouri; E. S. Lohan; I. Saastamoinen e M Renfors. On second order statistics of the satellite-to-indoor channel based on field measurements. *Proc. IEEE Symposium on Personal, Indoor, Mobile and Radio Communications (PIMRC)*, páginas 2632–2636, setembro de 2005.
- [23] E. Pajala; T. Isotalo; A. Lakhzouri e E. S. Lohan. An improved simulation model for nakagami-m fading channels for satellite positioning applications. *Proceedings Of The 3rd Workshop On Positioning, Navigation And Communication*, páginas 81–90, 2006.
- [24] M. D. Yacoub. Statistical properties of urban wcdma channel for mobile positioning applications. *International Journal of Wireless and Optical Communications*, páginas 1–16, dezembro de 2004.
- [25] Y. Zhang e D. Yuan. Rate-compatible ldpc codes for cross-layer design combining of amc with harq. *6th International Conference on ITS Telecommunications Proceedings*, páginas 537–540, 2006.
- [26] B. S. Tan; K. H. Li e K. C. Teh. Performance analysis of ldpc codes with maximum-ratio combining cascaded with selection combining over nakagami-m fading. *IEEE Transactions on Wireless Communications*, 10(6):1886–1894, junho de 2011.
- [27] W. E. Ryan e S. Lin. Channel codes: Classical and modern. *Cambridge University Press*, 2009.
- [28] W. C. Jakes. Microwave mobile communications. *IEEE Press, New York*, 1974.
- [29] M. Abramowitz e I. A. Stegun. Handbook of mathematical functions with formulas, graphs, and mathematical tables. *New York: Dover*, páginas 355, 1965.
- [30] M. Abramowitz e I. A. Stegun. Handbook of mathematical functions with formulas, graphs, and mathematical tables. *New York: Dover*, páginas 456, 1965.

- [31] M. Matthaiou e D.I. Laurenson. Rejection method for generating nakagami-m independent deviates. *Electronics Letters*, 43(25):159–168, dezembro de 2007.
- [32] L. Cao e N. C. Beaulieu. Simple efficient methods for generating independent and bivariate nakagami-m fading envelope samples. *IEEE Transactions on Vehicular Technology*, 56(4):1573–1579, julho de 2007.
- [33] J. C. S. S. Filho e M. D. Yacoub. On the simulation and correlation properties of phase-envelope nakagami fading processes. *IEEE Transactions on Communications*, 57(4):906–909, abril de 2009.
- [34] K. Zhang; Z. Song e Y. L. Guan. Simulation of nakagami fading channels with arbitrary cross-correlation and fading parameters. *IEEE Transactions on Wireless Communications*, 3(5):1463–1468, setembro de 2004.
- [35] A. Papoulis. Probability, random variables, and stochastic processes. *McGraw Hill*, 1965.
- [36] R. Storn e K. V. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, páginas 341–359, junho de 1997.
- [37] X. Hu. Software peg disponível em [http : //www.inference.phy.cam.ac.uk/mackay/peg\\_cc.html](http://www.inference.phy.cam.ac.uk/mackay/peg_cc.html). Acessado em julho de 2010.
- [38] R. H. Morelos-Zaragoza. Software the art of error correcting coding disponível em [http : //the - art - of - ecc.com/](http://the-art-of-ecc.com/). Acessado em maio de 2010.
- [39] R. M. Neal. Software for low density parity check (ldpc) codes disponível em [http : //www.cs.utoronto.ca/ radford/ldpc.software.html](http://www.cs.utoronto.ca/radford/ldpc.software.html). Acessado em março de 2010.
- [40] G. G. O. Brante. Comunicação cooperativa usando retransmissão parcial codificada, harq tipo-i e códigos ldpc. *UTFPR, Curitiba*, 2010.

- [41] F. A. Newagy; Y. A. Fahmy e M. S. El-Soudani. Designing near shannon limit ldpc codes using particle swarm optimization algorithm. *Proceedings of the 2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, páginas 119–123, maio de 2007.
- [42] W. Zhang e X. Xie. Depso: Hybrid particle swarm with differential evolution operator. *IEEE International Conference on Systems, Man e Cybernetics (SMCC)*, páginas 3816–3821, 2003.

## APÊNDICE A

A matriz irregular  $H_2$  do Capítulo 2 possui  $N = 7$ ,  $K = 4$  e  $N - K = 3$ , e é possível observar que as quatro primeiras colunas da matriz  $H_2$  formam a matriz  $A$  e as três últimas colunas da matriz  $H_2$  formam uma matriz identidade composta de três linhas e três colunas, de acordo com:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix},$$

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Desta maneira a matriz  $G$  é construída de acordo com (2.10), ou seja, é formada por uma matriz identidade de quatro linhas e quatro colunas e a matriz  $A^T$  nas três últimas colunas, de acordo com:

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

Supondo que a mensagem  $\mathbf{u}$  tenha de ser codificada, é realizado o processo mostrado em (2.8). Portanto se:

$$\mathbf{u} = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix},$$

então:

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

Chegando a um vetor  $\mathbf{c}$  que é a mensagem codificada, sendo que os primeiros quatro bits são a mensagem  $\mathbf{u}$  e os últimos três bits são redundância.

## APÊNDICE B

Suponha a matriz  $H_3$  de um código LDPC regular(2,3)[2]:

$$H_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

O grafo da matriz  $H_3$  é mostrado na Figura 6.1.

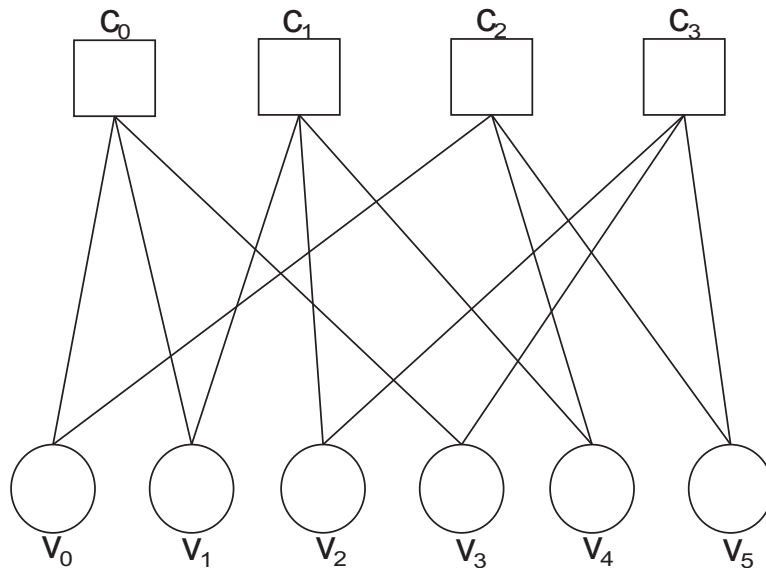


Figura 6.1: Grafo correspondente à matriz  $H_3$ .

Suponha que a mensagem  $\mathbf{u}$  foi codificada na mensagem  $\mathbf{c}$  utilizando a matriz  $H_3$ , desta forma:

$$\mathbf{u} = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

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

A mensagem  $\mathbf{c}$  é enviada por um canal AWGN com modulação BPSK e  $E_b/N_0 = 3,75$

(ou  $0.9691dB$ ) e o sinal recebido é:

$$\mathbf{y} = \begin{bmatrix} -0,1 & 0,5 & -0,8 & 1,0 & -0,7 & 0,5 \end{bmatrix}.$$

Se for realizada apenas a demodulação de forma abrupta na decodificação, existirão dois bits de erro neste vetor recebido, o 1º e 6º bits. Para um canal AWGN a relação logarítmica *a priori* para a entrada do canal é:

$$R_i = 4y_i R \frac{E_b}{N_0}, \quad (6.1)$$

deste modo:

$$\mathbf{R} = \begin{bmatrix} -0,5 & 2,5 & -4,0 & 5,0 & -3,5 & 2,5 \end{bmatrix},$$

é a entrada do decodificador *Belief Propagation*.

#### *Iteração 1*

O primeiro elemento de  $\mathbf{R}$  está incluído no primeiro e no terceiro nós de verificação, portanto as mensagens  $M_{1,1}$  e  $M_{3,1}$  que saem dos nós de variáveis para os nós de verificação são inicializadas com  $R_1$  [2]:

$$M_{1,1} = R_1 = -0,5; \quad M_{3,1} = R_1 = -0,5;$$

repetindo para todos os outros elementos de  $R$ :

$$M_{1,2} = R_2 = 2,5; \quad M_{2,2} = R_2 = 2,5;$$

$$M_{2,3} = R_3 = -4,0; \quad M_{4,3} = R_3 = -4,0;$$

$$M_{1,4} = R_4 = 5,0; M_{4,4} = R_4 = 5,0;$$

$$M_{2,5} = R_5 = -3,5; M_{3,5} = R_5 = -3,5;$$

$$M_{3,6} = R_6 = 2,5; M_{4,6} = R_6 = 2,5.$$

O próximo passo é o cálculo das mensagens que saem do nó de verificação  $j$  para o nó de variável  $i$ , desta forma utilizando a equação (2.13):

$$E_{1,1} = \ln \left( \frac{1 + \tanh(M_{1,2}/2)\tanh(M_{1,4}/2)}{1 - \tanh(M_{1,2}/2)\tanh(M_{1,4}/2)} \right) = \ln \left( \frac{1 + \tanh(2,5/2)\tanh(5,0/2)}{1 - \tanh(2,5/2)\tanh(5,0/2)} \right) = 2,4217;$$

É realizado o cálculo para cada uma das mensagens:

$$E_{1,2} = \ln \left( \frac{1 + \tanh(M_{1,1}/2)\tanh(M_{1,4}/2)}{1 - \tanh(M_{1,1}/2)\tanh(M_{1,4}/2)} \right) = \ln \left( \frac{1 + \tanh(-0,5/2)\tanh(5,0/2)}{1 - \tanh(-0,5/2)\tanh(5,0/2)} \right) = -0,4930;$$

$$E_{1,4} = \ln \left( \frac{1 + \tanh(M_{1,1}/2)\tanh(M_{1,2}/2)}{1 - \tanh(M_{1,1}/2)\tanh(M_{1,2}/2)} \right) = \ln \left( \frac{1 + \tanh(-0,5/2)\tanh(2,5/2)}{1 - \tanh(-0,5/2)\tanh(2,5/2)} \right) = -0,4217;$$

$$E_{2,2} = \ln \left( \frac{1 + \tanh(M_{2,3}/2)\tanh(M_{2,5}/2)}{1 - \tanh(M_{2,3}/2)\tanh(M_{2,5}/2)} \right) = \ln \left( \frac{1 + \tanh(-4,0/2)\tanh(-3,5/2)}{1 - \tanh(-4,0/2)\tanh(-3,5/2)} \right) = 3,0265;$$

$$E_{2,3} = \ln \left( \frac{1 + \tanh(M_{2,2}/2)\tanh(M_{2,5}/2)}{1 - \tanh(M_{2,2}/2)\tanh(M_{2,5}/2)} \right) = \ln \left( \frac{1 + \tanh(2,5/2)\tanh(-3,5/2)}{1 - \tanh(2,5/2)\tanh(-3,5/2)} \right) = -2,1892;$$

$$E_{2,5} = \ln \left( \frac{1 + \tanh(M_{2,2}/2)\tanh(M_{2,3}/2)}{1 - \tanh(M_{2,2}/2)\tanh(M_{2,3}/2)} \right) = \ln \left( \frac{1 + \tanh(2,5/2)\tanh(-4,0/2)}{1 - \tanh(2,5/2)\tanh(-4,0/2)} \right) = -2,3001.$$

Repetindo para todos os nós de verificação, chega-se ao resultado final para todas as mensagens  $E$ :

$$E = \begin{bmatrix} 2,4217 & -0,4930 & \cdot & -0,4217 & \cdot & \cdot \\ \cdot & 3,0265 & -2,1892 & \cdot & -2,3001 & \cdot \\ -2,1892 & \cdot & \cdot & \cdot & -0,4217 & 0,4696 \\ \cdot & \cdot & 2,4217 & -2,3001 & \cdot & -3,6869 \end{bmatrix}.$$

Os resultados são mostrados em forma matricial, onde o  $(j,i)$ -ésimo entrada de  $E$  é denotado  $E_{j,i}$ . Uma entrada ' $\cdot$ ' em  $E$  indica que a mensagem não existe para tal  $i$  e  $j$ .

A informação *a priori* de cada bit da palavra código recebido  $R$  é adicionada à respectiva mensagem  $E$ , resultando na mensagem  $L$ . Desta forma:

$$L_1 = R_1 + E_{1,1} + E_{3,1} = -0,2676;$$

$$L_2 = R_2 + E_{1,2} + E_{2,2} = 5,0334;$$

$$L_3 = R_3 + E_{2,3} + E_{4,3} = -3,7676;$$

$$L_4 = R_4 + E_{1,4} + E_{4,4} = 2,2783;$$

$$L_5 = R_5 + E_{2,5} + E_{3,5} = -6,2217;$$

$$L_6 = R_6 + E_{3,6} + E_{4,6} = -0,7173;$$

portanto:

$$\mathbf{L} = \begin{bmatrix} -0,2676 & 5,0334 & -3,7676 & 2,2783 & -6,2217 & -0,7173 \end{bmatrix}.$$

A mensagem  $\mathbf{z}$  é a decisão abruta sobre o sinal dos valores de  $L$ , resultando na seguinte palavra código:

$$\mathbf{z} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix},$$

para verificar se  $\mathbf{z}$  é uma palavra código válida, é realizado o cálculo da síndrome:

$$\mathbf{s} = H\mathbf{z}^T = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^T.$$

Como o resultado não é zero, a palavra decodificada não é válida e portanto é realizada uma nova itereação do algoritmo de decodificação.

#### *Iteração 2*

É realizado um novo cálculo das mensagens  $M$ , de acordo com (2.14):

$$M = \begin{bmatrix} -2,6892 & 5,5265 & \cdot & 2,6999 & \cdot & \cdot \\ \cdot & 2,0070 & -1,5783 & \cdot & -3,9217 & \cdot \\ 1,9217 & \cdot & \cdot & \cdot & -5,8001 & -1,1869 \\ \cdot & \cdot & -6,1892 & 4,5783 & \cdot & 2,9696 \end{bmatrix}.$$

Em seguida são realizados os mesmos procedimentos descritos na Iteração 1, chegando-se aos seguintes resultados para a Iteração 2:

$$E = \begin{bmatrix} 2,6426 & -2,0060 & \cdot & -2,6326 & \cdot & \cdot \\ \cdot & 1,4907 & -1,8721 & \cdot & -1,1041 & \cdot \\ 1,1779 & \cdot & \cdot & \cdot & -0,8388 & -1,9016 \\ \cdot & \cdot & 2,7877 & -2,9305 & \cdot & -4,3963 \end{bmatrix},$$

$$\mathbf{L} = \begin{bmatrix} 3,3206 & 1,9848 & -3,0845 & -0,5630 & -5,4429 & -3,7979 \end{bmatrix},$$

$$\mathbf{z} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\mathbf{s} = H\mathbf{z}^T = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T.$$

Como o resultado da verificação da palavra código ainda não é válido, realiza-se uma nova iteração, até o algoritmo convergir.

*Iteração 3*

$$M = \begin{bmatrix} 0,6779 & 3,9907 & \cdot & 2,0695 & \cdot & \cdot \\ \cdot & 0,4940 & -1,2123 & \cdot & -4,3388 & \cdot \\ 2,1426 & \cdot & \cdot & \cdot & -4,6041 & -1,8963 \\ \cdot & \cdot & -5,8721 & 2,3674 & \cdot & 0,5984 \end{bmatrix},$$

$$E = \begin{bmatrix} 1,9352 & 0,5180 & \cdot & 0,6515 & \cdot & \cdot \\ \cdot & 1,1733 & -0,4808 & \cdot & -0,2637 & \cdot \\ 1,8332 & \cdot & \cdot & \cdot & -1,3362 & -2,0620 \\ \cdot & \cdot & 0,4912 & -0,5948 & \cdot & -2,3381 \end{bmatrix},$$

$$L = \begin{bmatrix} 3,2684 & 4,1912 & -3,9896 & 5,0567 & -5,0999 & -1,9001 \end{bmatrix},$$

$$\mathbf{z} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix},$$

$$\mathbf{s} = H\mathbf{z}^T = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

O cálculo da síndrome foi zero e a mensagem  $\mathbf{z}$  obtida na terceira iteração é igual a mensagem transmitida. Entretanto como pode ser visto o custo computacional é alto.