ANDREI PICCININI LEGG

CODIFICAÇÃO LDPC PARA APLICAÇÕES EM CÓDIGOS DE BARRA 2D COLORIDOS

FLORIANÓPOLIS
2011

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CODIFICAÇÃO LDPC PARA APLICAÇÕES EM CÓDIGOS DE BARRA 2D COLORIDOS

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a obtenção
do grau de Doutor em Engenharia Elétrica.

ANDREI PICCININI LEGG

Florianópolis, junho de 2011

CODIFICAÇÃO LDPC PARA APLICAÇÕES EM CÓDIGOS DE BARRA 2D COLORIDOS

Andrei Piccinini Legg

'Esta Tese foi julgada adequada para obtenção do título de Doutor em Engenharia Elétrica, área de concentração em *Comunicações e Processamento de Sinais*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.'

Bartolomeu F. Uchôa Filho, Ph.D.
Orientador

Patrick Kuo-Peng, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Bartolomeu F. Uchôa Filho, Ph.D.
Presidente

Renato da Rocha Lopes, Ph.D.

Jaime Portugheis, Dr.-Ing.

Danilo Silva, Ph.D.

Carlos Aurélio Faria da Rocha, D.Sc.

Florianópolis, 21 de junho de 2011

Agradecimentos

Aos meus pais, Silvio Luiz Alves Legg e Vera Beatriz Piccinini Legg, ao meu tio, Joaquim Antonio Piccinini, e à minha avó, Nelsi Piccinini, pelo carinho e constante apoio;

à minha querida Ilka, pelo apoio, carinho e compreensão;

ao meu orientador e estimado amigo, Bartolomeu Ferreira Uchôa Filho, pela sua dedicação, incentivo e valiosíssima orientação;

aos Professores Renato da Rocha Lopes, Jaime Portugheis, Danilo Silva, Carlos Aurélio Faria da Rocha pela valiosa participação na banca examinadora;

aos meus grandes amigos Bruno Sens Chang, João Fernando Refosco Baggio, João Luiz Rebelatto, Marcelo Massayuki Sunada, Marcelo de Souza, Mário de Noronha Neto, Pedro Giassi Junior, Renato Machado, Roberto Wanderley da Nóbrega e Thiago Henrique da Silva, por me apoiarem em momentos de dificuldade;

aos Professores Raimes Moraes e Leonardo Silva Resende, pela amizade, pelos bons conselhos e pelo bom convívio profissional;

aos meus amigos e colegas de trabalho, pelos momentos de lazer e confraternização;

ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo auxílio financeiro, sem o qual não seria possível o desenvolvimento desta tese;

e a todos aqueles que não foram citados aqui, mas que de alguma forma contribuíram para a realização desta tese,

meus sinceros agradecimentos.

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

CODIFICAÇÃO LDPC PARA APLICAÇÕES EM CÓDIGOS DE BARRA 2D COLORIDOS

Andrei Piccinini Legg

Junho/2011

Orientador: Bartolomeu F. Uchôa Filho, Ph.D.
Área de Concentração: Comunicações e Processamento de Sinais
Palavras-chaves: Códigos corretores de erros, códigos LDPC, codificador LDPC, códigos de barra 2D coloridos, EXIT *charts*.
Número de páginas: xxiii + 105

Nesta tese, aborda-se o armazenamento de informação em códigos de barra 2D coloridos, os quais devem ser robustos à impressão seguida de digitalização através de um *scanner*. Estes dois processos geram uma distorção (ruído), fazendo com que o código de barra obtido na saída do *scanner* seja diferente daquele original enviado. Isso define um modelo de canal de comunicação, que recebe o nome de canal PS (*print and scan*). A contribuição desta tese é propor códigos corretores de erros para proteger os códigos de barra 2D das adversidades do canal PS. Em particular, códigos LDPC são investigados. Um estudo foi realizado a partir do qual identificou-se que o canal PS é um caso particular do modelo de canal Gaussiano multidimensional assimétrico aditivo (AWAMGN). As capacidades de canal (ou taxas alcançáveis) do canal PS são obtidas, o que estabelece um limite para as taxas máximas que os códigos LDPC podem ter para se ter uma probabilidade de erro próxima de zero. Para essas taxas, códigos LDPC são projetados. A otimização dos códigos é feita através do método de curvas de transferência de informação extrínseca (EXIT *charts*). No decorrer deste estudo, um novo codificador para códigos LDPC é proposto. Este codificador apresenta complexidade de codificação linear no comprimento da palavra-código. Novos códigos LDPC otimizados especialmente para o canal PS são propostos, e seu desempenho (probabilidade de erro) é avaliado através de simulações computacionais. Por fim, os resultados são comentados, discutidos e novas ideias para pesquisas relacionadas são apresentadas.

Abstract of Thesis presented to UFSC in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering

LDPC CODING FOR APPLICATIONS IN 2D COLORED BARCODES

Andrei Piccinini Legg

June/2011

Advisor: Bartolomeu F. Uchôa Filho, Ph.D.
Area of Concentration: Communications and Signal Processing
Keywords: Error correcting codes, LDPC codes, LDPC encoder, 2D colored barcodes, EXIT charts.
Number of pages: xxiii + 105

This thesis investigates the information storage in 2D colored barcodes, which must be robust to a print and scan process. These two processes generate a distortion (noise), which causes the barcode at the output of the scanner to be different from the original one. This defines a communication channel which is named the PS (print and scan) channel. The contribution of this thesis is to propose error correcting codes to protect the 2D barcodes from PS channel adversities. In particular, LDPC codes are investigated. A study is conducted by which it is identified that the PS channel is a particular case of the additive white asymmetric multidimensional Gaussian noise (AWAMGN) channel model. The channel capacities (or achievable rates) of the PS channel are obtained, which put a limit on the rates the LDPC codes could have in order to obtain an error probability close to zero. For these rates, LDPC codes are designed. The code optimization is done through EXtrinsic Information Transfer (EXIT) charts method. Along this study, a new LDPC encoder is proposed. This encoder has encoding complexity linear in the codeword length. New LDPC codes optimized specifically for the PS channel are proposed and their performance (error probability) is evaluated through computer simulations. Finally, the results are discussed, and new ideas for related research are presented.

Sumário

Sumário **xiii**

Lista de Figuras **xiv**

Lista de Abreviaturas **xvi**

Lista de Símbolos **xix**

1 Introdução **1**
1.1 Objetivos 1
1.2 Motivação 2
1.3 Um breve histórico sobre códigos corretores de erros . . 2
1.4 Organização da Tese 3

2 Canal *print and scan* **5**
2.1 Impressoras e *scanners* 6
2.1.1 Reprodução de cores 7
2.1.2 Conceitos de *halftoning* 8
2.2 Modelo ponto×ponto para o processo *print and scan* . . 11

3 Códigos de barra bidimensionais **15**
3.1 Códigos de barra 2D preto e branco 16
3.2 Mod. para o canal PS para cód. de barra 2D p&b . . . 17
3.2.1 Modelo com o ruído Gaussiano independente do símbolo transmitido 18
3.2.2 Modelo com o ruído Gaussiano dependente do símbolo transmitido 19

3.2.3 Modelo com o ruído Gaussiano generalizado dependente do símbolo transmitido 20
3.3 Códigos de barra 2D coloridos 21
3.4 Mod. prop. de canal PS para cód. de barra 2D coloridos 21
3.4.1 Exemplo de um código de barra 2D colorido . . . 26

4 Modulação codificada 29
4.1 Sistema de comunicações dig. com modulação codificada 29
4.1.1 Modulação codificada em treliças 30
4.1.2 Mapeamento dos bits 31
4.2 Partição de conjuntos 31
4.3 Códigos convolucionais 33
4.4 Modulação codificada multinível 36
4.5 Modulação codificada com entrelaçamento de bit 40

5 Códigos LDPC 43
5.1 Códigos LDPC regulares e irregulares 43
5.2 Grafos de Tanner 44
5.2.1 Conceito de *girth* 45
5.2.2 Codificação 46
5.2.3 Decodificação 46
5.3 Projeto de códigos LDPC 47
5.3.1 *Density evolution* 50
5.3.2 EXIT *charts* 50
5.3.3 Algoritmo de otimização de códigos LDPC usando EXIT *charts* 58

6 Códigos LDPC para códigos de barra 2D coloridos 61
6.1 Obtenção das matrizes de verificação de paridade 61
6.1.1 Método de codificação LDPC utilizado 63
6.1.2 Método de decodificação LDPC utilizado 64
6.2 Mod. codificada em sistemas de códigos de barra 2D . . 65
6.3 Otimização de códigos LDPC 68
6.4 Resultados de simulações 73

7 Considerações finais 79

A Algoritmo SPA 81
A.1 Inicialização 81
A.2 Atualizando os R_{ij}^a 82
A.3 Atualizando os Q_{ij}^a 83
A.4 Tentativa de decodificação 83

B Cálculo das taxas de informação **85**
B.1 Entropia 85
B.1.1 Entropia Diferencial 85
B.2 Entropia condicionada 86
B.2.1 Entropia diferencial condicionada 86
B.3 Informação mútua 86
B.4 Cálculo das taxas de transmissão para os canais PS . . . 87

C Distribuição Gaussiana generalizada **89**

D Distribuição de graus utilizadas **91**

E Alg. de otimização de cód. LPDC via EXIT *charts* **95**
E.1 Adaptação dos perfis de graus 95
E.2 Fluxograma 97

Referências Bibliográficas **99**

Lista de Figuras

2.1 Diagrama de amostragem de um *scanner*. 7
2.2 Sistemas de reprodução de cores. 7
2.3 Exemplos de *halftoning*. 9
2.4 Reprodução do primeiro *halftoning*: *A Scene in Shanty-town*. 10
2.5 Células de *halftone* de dimensões 5×5 pontos para tons de cinza. 11
2.6 Exemplo de um *halftoning* para cor azul. 11
2.7 Diagrama de blocos do modelo ponto×ponto para o processo *print and scan*. 12

3.1 Exemplos de códigos de barra 2D preto e branco comerciais. 16
3.2 Modulação e demodulação para o canal de comunicação PS. 18
3.3 Diagrama para o modelo de canal PS com ruído Gaussiano independente do símbolo transmitido. 19
3.4 Diagrama para o modelo de canal PS com ruído Gaussiano dependente do símbolo transmitido. 20
3.5 Diagrama para o modelo de canal PS com ruído Gaussiano generalizado dependente do símbolo transmitido. 20
3.6 Padrão de códigos de barra 2D coloridos desenvolvido pela Microsoft. 21
3.7 Diagrama de blocos para o modelo proposto do canal PS colorido. 23

3.8 Amostras obtidas na saída do "pior" canal PS no cubo CMY. 24
3.9 Amostra de um código de barra 2D. 26
3.10 Amostra de um código de barra 2D na saída do canal PS. 27

4.1 Partição da constelação 8-PSK. 32
4.2 Codificador convolucional de Ungerboeck com 4 estados. 34
4.3 Diagrama de treliça para o código convolucional. Dois caminhos concorrentes são mostrados em destaque. . . . 34
4.4 Capacidades para canais Gaussianos com modulações PSK. 36
4.5 pdfs condicionais para 8-PAM. 37
4.6 Codificador multinível. 38
4.7 Decodificador multi-estágios. 40
4.8 Decodificadores independentes em paralelo. 41
4.9 Diagrama de blocos com modulação codificada com entrelaçamento de bit. 41

5.1 Uma representação de grafo bipartido. 44
5.2 Exemplo de um *girth* 4 em um grafo bipartido. 45
5.3 Diagrama de blocos de um decodificador iterativo LDPC. 51
5.4 Diagrama de blocos para o cálculo das funções EXIT de códigos LDPC regulares. 54
5.5 EXIT *charts* de um código LDPC regular com taxa 1/3 ($d_v = 2$ e $d_c = 3$). 56
5.6 Diagrama de blocos de um decodificador iterativo LDPC para analise de EXIT *charts*. 58
5.7 Dois pares de EXIT *charts* rotulados como 1 e 2. O par 1 possui melhores propriedades de convergência. 59

6.1 Curvas de capacidade de armazenamento. 66
6.2 Curvas de capacidade de armazenamento dos canais binários equivalentes. 67
6.3 Histogramas de L_E condicionada aos bits transmitidos. 68
6.4 Gráfico dos integrandos da equação 6.13. 69
6.5 Curvas para $I(X; L_E)$ e $I(X; L_D)$ em função de SNR (dB). 70
6.6 Diagrama de blocos para os códigos LDPC no canal PS. 71
6.7 EXIT *charts* para o canal binário equivalente esquerdo. 72
6.8 EXIT *charts* para o canal binário equivalente direito. . . 73
6.9 Curvas de BER para códigos LDPC submetidos ao canal PS. 74
6.10 Comparação de BER para taxa total de $1,9$ bits/bloco. 75

6.11 Comparação de BER dos códigos LDPC otimizados e não otimizados. 76
6.12 WER dos códigos LDPC otimizados e não otimizados. . 77

A.1 Troca de mensagens em um grafo bipartido. 81

B.1 Modelo simplificado do canal PS. 87

C.1 Distribuição Gaussiana generalizada para diferentes valores de γ. 90

E.1 Exemplo de um passo fundamental qualquer. 96
E.2 Fluxograma referente ao algoritmo de otimização de LDPC baseado em EXIT *charts*. 97

Lista de abreviaturas

LDPC	*Low-density parity-check*
PS	*Print and scan*
pdf	*Probability density function*
RGB	*Red, green and blue*
CMY	*Cyan, magenta and yellow*
HVS	*Human vision system*
DBS	*Direct binary search*
lpi	*lines per inch*
dpi	*Dots per inch*
CCD	*Charged coupled devices*
TCM	*Trellis coded modulation*
PSK	*Phase shift keying*
SPA	*Sum-product algorithm*
BER	*Bit error ratio*
WER	*Word error ratio*
SNR	*Signal-to-noise ratio*
LLR	*Log likelihood ratio*
AWAMGN	*Additive white asymmetric multidimensional Gaussian noise*
AWGN	*Additive white Gaussian noise*
BEC	*Binary erasure channel*
QC-LDPC	*Quasi-cyclic Low-density parity-check*

MLC *Multilevel Coding*
MSD *Multistage Decoding*
BICM *Bit interliver coded modulation*
EXIT *Extrinsic information transfer*

Lista de símbolos

d_E^2	Distância Euclidiana quadratica
d_H	Distância de Hamming
μ,$\boldsymbol{\mu}$	Média, vetor média
$\boldsymbol{x}$,$\boldsymbol{y}$	Vetor transmitido, vetor recebido
$\boldsymbol{C}$	Matriz de covariância
ξ	Parâmetro multiplicativo escalar
$E[\cdot]$	Esperança matemática
$(\cdot)^T$	Operação transposto
$\prod$	Produtório
$\sum$	Somatório
$\int$	Integral
ν	Memórias do codificador
$P_r(e)$	Probabilidade de erro
$\mathbb{R}$	Conjunto dos números reais
$\mathcal{N}(\mu, \sigma^2)$	Distribuição Gaussiana unidimensional com média μ e desvio padrão σ

Capítulo 1

Introdução

1.1 Objetivos

O objetivo desta tese é projetar códigos LDPC (*Low-Density Parity-Check*) para códigos de barra bidimensionais (2D) coloridos com o intuito de proteger a informação contida nos códigos de barra contra as adversidades do canal de comunicação correspondente ao processo de impressão seguida de escaneamento. O modelo de canal para esse sistema de comunicação é um caso particular do canal AWAMGN (*Additive White Asymmetric Multidimensional Gaussian Noise*). A fim de se trabalhar com a otimização dos códigos LDPCs, serão realizadas adaptações nas ferramentas tradicionais normalmente utilizadas para esse fim. Desse modo, pretende-se contribuir com novas ferramentas para a otimização dos LDPCs em canais que seguem esse modelo estatístico. Buscar-se-á, portanto, resolver problemas gerais, porém sem se afastar do foco principal da tese: buscar solução para o problema de codificação de canal para inserção de informação em códigos de barras 2D coloridos.

A tese consiste em adaptar os códigos LDPC, propondo alterações na inicialização do seu decodificador e determinando os principais parâmetros necessários para o projeto dos códigos, levando as propriedades do canal AWAMGN em consideração. Serão investigados o desempenho dos códigos LDPC irregulares com diferentes distribuições de graus e também parâmetros como, por exemplo, a máxima taxa de código que pode ser usada para atingir-se um bom desempenho.

1.2 Motivação

A motivação desta tese está associada à aplicação de inserção de informação em códigos de barras 2D. Existe grande interesse comercial ligado aos códigos de barra 2D. O canal *print and scan* (PS) colorido consiste basicamente em uma impressão de símbolos em papel e uma posterior digitalização através do uso de um *scanner*. A impressora, o papel e o *scanner* introduzem muitos tipos de distorções, que devem ser combatidas utilizando diferentes técnicas a fim de se recuperar a informação original. Os códigos corretores de erros são de grande importância na solução desse problema, uma vez que as aplicações comerciais exigem erro praticamente zero.

1.3 Um breve histórico sobre códigos corretores de erros

A história dos Códigos Corretores de Erros teve início em 1948 com a publicação do artigo: "Uma Teoria Matemática das Comunicações" [1], escrito pelo engenheiro e matemático Claude E. Shannon, do Laboratório da Bell. Hoje conhecida como "A Teoria da Informação" [2], ela demonstra que cada canal de comunicações tem uma capacidade limite característica, que pode ser determinada. Os códigos corretores de erros são o caminho para se alcançar esta capacidade. Inicialmente os mais interessados nesta teoria foram os matemáticos, que a desenvolveram consideravelmente nas décadas de 50 e de 60. A partir da década de 70, com as pesquisas espaciais e a grande popularização dos computadores, essa teoria começou a interessar também aos engenheiros.

Hoje os códigos corretores de erros estão presentes no cotidiano de inúmeras formas, quando assiste-se a um programa de televisão, fala-se ao telefone, ouve-se um CD de música, assiste-se a um filme em DVD, navega-se na Internet e em praticamente todas as aplicações nas quais utiliza-se algum tipo de informação digitalizada.

Em sistemas de comunicações, códigos corretores de erros são frequentemente utilizados com o objetivo de reduzir a taxa de erros. Há também diferentes tipos de erros associados às características do canal. É necessário determinar quais são os tipos de erros que podem estar presentes na saída do canal, a fim de se determinar quais códigos corretores de erros são mais apropriados.

Os códigos LDPC foram introduzidos junto com um algoritmo de

decodificação iterativo, desenvolvido por Gallager no início da década de 60 [3]. Eles foram construídos usando matrizes de verificação de paridade esparsas e aleatórias, e devido a essa "esparsidade" estão associados a decodificadores de baixa complexidade.

Os códigos LDPC ficaram praticamente esquecidos por mais de três décadas até que, com o advento dos códigos turbo [4, 5] (que até então formavam a classe de códigos com a qual se chegava mais próximo da capacidade de canal, porém com uma complexidade de decodificação ainda bastante elevada), no inicio da década de 90 foram "redescobertos" por MacKay e Neal [6], que demonstraram o potencial desses códigos para se alcançar a capacidade de canal, comparável ao dos códigos turbo. Mais recentemente, Richardson e Urbanke [7] desenvolveram os códigos LDPC irregulares, que possuem um desempenho melhor do que aquele dos códigos turbo para blocos com comprimentos muito grandes ($> 10^5$), podendo chegar muito próximos da capacidade de Shannon para o canal Gaussiano.

1.4 Organização da Tese

No Capítulo 2 são apresentadas as principais características do canal PS. No Capítulo 3 são apresentados os principais modelos para o canal PS preto e branco e o modelo proposto para o canal PS colorido, todos desenvolvidos para códigos de barra bidimensionais (2D). No Capítulo 4 são apresentados conceitos sobre a modulação codificada. No Capítulo 5 são apresentados os códigos LDPC (*Low-density parity-check*). No Capítulo 6 é apresentado o procedimento escolhido para o projeto de códigos LDPC desenvolvidos para o canal PS, que é um caso particular do canal Gaussiano Multidimensional Assimétrico. Neste mesmo capítulo também são apresentados os EXIT *charts* e os resultados de simulação. No Capítulo 7 os resultados são comentados, discutidos e novas ideias para pesquisas relacionadas são apresentadas. No Apêndice A é apresentado o algoritmo SPA (*Sum-Product Algorithm*) em sua forma mais geral. O algoritmo SPA é o utilizado na decodificação de Códigos LDPC. No Apêndice B é detalhado o procedimento necessário para se obter as máximas taxas de transmissão alcançáveis em bits/bloco para o canal PS. No Apêndice C, faz-se uma revisão sobre distribuição Gaussiana generalizada (GGD). No Apêndice D, são apresentados os polinômios de distribuições de graus de variável e de função associados aos códigos LDPC, que foram adotados nas simulações aqui realizadas. E finalmente, no Apêndice E são apresentados detalhes sobre o método de otimização escolhido

para os códigos LDPC.

Capítulo 2

Canal *print and scan*

Quando uma imagem é impressa e depois passada pelo processo de digitalização utilizando-se um *scanner*, a imagem obtida é uma versão ruidosa e distorcida da original. Essas distorções e esses ruídos são o resultado conjunto dos efeitos da impressora, do papel e do *scanner*. Várias distorções podem ser consideradas neste processo, tais como: filtragens passa-baixa, *aliasing*, escala, rotações, ajustes de contraste e de luminância, entre outras. Um problema que aparece é que as intensidades e os tipos de distorções variam de acordo com o par impressora-*scanner* utilizado.

Um dos objetivos desta tese é propor um modelo de comunicação para o canal *print and scan* (PS) colorido que seja o mais independente possível dos dispositivos utilizados, ou seja, um modelo robusto.

Para se criar um sistema de comunicação utilizando o canal PS, primeiramente é necessário determinar quais serão os símbolos 2D utilizados, ou seja, o alfabeto de entrada do canal. Após determinados os símbolos 2D utilizados é necessária uma descrição estatística do canal.

Antes de entrar em detalhes sobre o modelo proposto de canal PS, serão apresentados os modelos para uma impressora e para um *scanner* genéricos e também um modelo, bastante utilizado na literatura, que trata os dispositivos em conjunto seguindo uma abordagem ponto×ponto (*pixel*×*pixel*), frequentemente utilizada em sistemas de marcas da água ou com informação oculta [8–11]. O modelo que será apresentado a seguir considera grande parte dos efeitos do canal PS e proporciona uma visão geral do problema. Por outro lado, os modelos existentes que tratam o canal PS como um canal de comunicação, assim

como o modelo de canal proposto para o caso de códigos de barra 2D coloridos, são muito mais simplificados. Eles foram baseados em experimentos e serão apresentados nesta Tese no Capítulo 3.

2.1 Impressoras e *scanners*

Para melhor caracterizar o canal, primeiramente será realizada uma análise dos processos de impressão e digitalização, a fim de descrever o modelo para um par impressora-*scanner* genérico.

O principal componente de um *scanner* é o sensor CCD (*Charge Coupled Device*). A tecnologia CCD é a mais utilizada para a captura de imagens em *scanners*. O CCD é um conjunto de minúsculos diodos fotossensíveis; quanto mais intensa for a luz que atingir um determinado fotodiodo, maior será a diferença de tensão elétrica obtida em seus terminais de saída.

A aquisição da imagem através do uso de um *scanner* é feita de forma diferente no sentido horizontal e no sentido vertical. A amostragem na direção horizontal é realizada por meio de uma amostragem ótica seguida por uma sub-amostragem digital realizada pelos CCDs que pode produzir uma amostragem não-uniforme. Na direção vertical a amostragem é realizada apenas de forma ótica.

Um sistema ótico composto por lentes e espelhos focaliza a imagem no campo de visão para a matriz de CCDs. O sistema ótico é projetado de tal forma que realiza uma filtragem passa-baixas. A matriz de CCDs e o sistema ótico são montados num carrinho que se move ao longo da direção vertical para produzir amostras na resolução desejada. Entretanto, uma vez que o carrinho não pára quando os sensores CCDs estão adquirindo as amostras, o resultado é uma imagem desfocada e filtrada (passa-baixas).

Assim, a amostragem ótica produz uma imagem digital uniformemente amostrada na taxa de amostragem ótica (resolução escolhida). A imagem obtida é uma versão da imagem de entrada filtrada por um filtro passa-baixas ótico na direção horizontal e filtrada por um filtro passa-baixas ótico e por um segundo filtro passa-baixas, sendo este último devido ao efeito do movimento no sentido vertical. Na Figura 2.1 é apresentado um diagrama de blocos das amostragens do *scanner*, em 2.1(a) para o sentido horizontal e em 2.1(b) para o sentido vertical. Mais detalhes sobre o funcionamento do *scanner* podem ser encontrados em [8, 12].

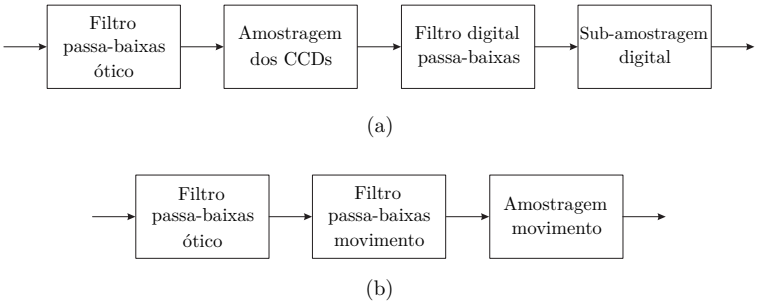
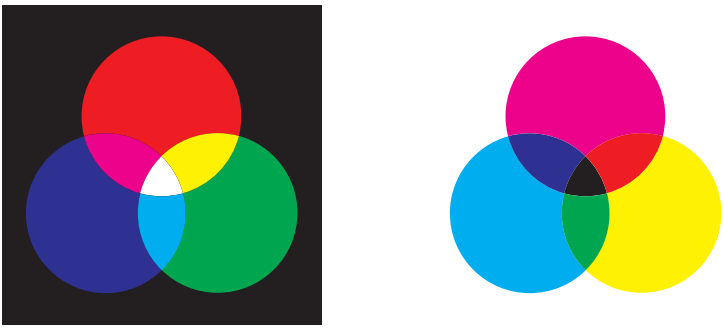


Figura 2.1: Diagrama de amostragem de um *scanner*.

2.1.1 Reprodução de cores

Para reproduzir uma cor, um dispositivo precisa misturar um número de cores primárias (básicas). Um monitor de computador ou uma tela de televisão normalmente utiliza vermelho, verde e azul (RGB) como cores primárias. A luz é emitida nos comprimentos de onda das cores básicas, e as outras cores são reproduzidas misturando-se as cores básicas com intensidades diferentes. Por exemplo, a cor branca é reproduzida com a soma das três cores, enquanto que a cor preta é formada quando nenhuma das cores está presente. Isso é chamado mistura aditiva de cores e é ilustrada na Figura 2.2(a).



(a) aditivo. (b) subtrativo.

Figura 2.2: Sistemas de reprodução de cores.

Entretanto, para reproduzir cores em impressões utilizam-se tintas que refletem parte da luz branca. As cores primárias são obtidas

utilizando-se tintas que refletem somente os comprimentos de onda desejados para cada uma das cores básicas. As três cores básicas normalmente utilizadas na impressão são ciano, magenta e amarelo (CMY). Quando não há tinta, toda a luz recebida é refletida de volta (partindo do princípio de que o papel ou o substrato seja branco) e a cor branca é obtida. Quando todas as três cores são impressas umas sobre as outras, toda a luz é absorvida e a cor preta é reproduzida. Isso é chamado mistura subtrativa de cores e é ilustrada na Figura 2.2(b).

Esses dois sistemas de cores são baseados nas características dos dispositivos. Os mesmos valores de RGB poderiam ser entendidos como cores diferentes quando reproduzidos por diferentes dispositivos. O mesmo é válido para valores de CMY. É por isso que esses sistemas de cores são chamados dispositivo dependentes. Embora ciano, magenta e amarelo impressos sobrepostos irão reproduzir teoricamente o preto, na prática, produz marrom escuro. A fim de produzir o preto puro e evitar a impressão de três cores primárias sobrepostas, o que provocaria outros problemas, normalmente o preto (K) é usado como uma quarta cor primária.

Valores para uma determinada cor no sistema RGB podem ser facilmente convertidos para o sistema de cores CMY, normalmente utilizado pelas impressoras, através da equação a seguir,

$$[\hat{c}, \hat{m}, \hat{y}]^T = [255, 255, 255^*]^T - [\hat{r}, \hat{g}, \hat{b}]^T. \tag{2.1}$$

Devido às propriedades e às limitações dos dispositivos de reprodução, não é possível reproduzir todas as cores presentes na imagem original. O conjunto de cores que podem ser reproduzidas por um dispositivo é chamado de *gamut* de cores do dispositivo. A fim de reproduzir as cores corretamente, ou tão precisamente quanto possível, as impressoras utilizam algoritmos de *halftoning*, descritos a seguir, adequados aos tipos de tintas utilizadas e a suas características físicas.

2.1.2 Conceitos de *halftoning*

O objetivo desta seção é apresentar de uma forma simples os conceitos de *halftoning* (maiores detalhes podem ser encontrados em [13, 14]). *Halftoning* é a uma técnica que simula imagens de tom contínuo através do uso de pontos, variando-os tanto em tamanho quanto em afastamento. É a característica passa-baixas da visão humana que faz com que padrões de pontos organizadamente

*O sistema de cores utiliza 24 bits, 8 bits para cada componente, resultando em 256 niveis diferentes para cada componente de cor.

distribuídos sejam percebidos como imagens de tom contínuo com uma gama infinita de cores ou de tons de cinza. Na Figura 2.3 são apresentadas imagens originais e imagens em *halftoning*, nas quais é possível visualizar os mecanismos empregados nesta técnica.

(a) original. (b) *halftoning* colorido.

(c) preto e branco. (d) *halftoning* preto e branco.

Figura 2.3: Exemplos de *halftoning*.

O primeiro algoritmo de *halftoning* foi proposto por Stephen H. Horgan. O primeiro uso de *halftone* foi feito pela *New York Daily Graphic*, em 4 de março de 1880, para imprimir uma fotografia intitulada: *A Scene in Shanty-town* (reproduzida na Figura 2.4).

A maneira mais simples de realizar o *halftone* consiste em dividir a imagem original em pequenos quadrados chamados de células de *halftone*. A área abrangida pela célula de *halftone* representará o tom de cinza da área correspondente na imagem original. Cada célula de *halftone* consiste em uma série de pontos menores. Na Figura 2.5 são apresentados exemplos de *halftoning* em células de 5×5 pontos. Na Figura 2.5(a) estão também representados os valores L e l que correspondem respectivamente ao comprimento da célula de *halftone*

Figura 2.4: Reprodução do primeiro *halftoning*: *A Scene in Shanty-town.*

e ao comprimento de cada um dos pontos que compõem a célula. A Figura 2.5(a) representa o tom de cinza de 1/25 e a Figura 2.5(b) o tom de cinza de 13/25. Portanto, é possível representar $26 = (5^2 + 1)$ diferentes tons de cinza com células de 5×5 pontos. O número de células de *halftone* por polegada é dada em lpi (*lines per inch*), mas não representa a resolução da impressora. A resolução de impressora é normalmente indicada em dpi (*dots per inch*). Quanto maior a lpi menor será a célula de *halftone* e, consequentemente, mais difícil será para o olho humano detectar os pontos de *halftoning*. Estudos têm mostrado que os pontos de *halftoning* não são detectados pelo olho humano na distância normal de visualização para valores superiores a 200 lpi [15, 16].

Considerando um *halftoning* com a utilização apenas da tinta preta, o número de tons de cinza que podem ser reproduzidos por uma impressora está relacionado com o tamanho da célula de *halftone* e a resolução da impressora da seguinte forma,

$$\text{tons de cinza} = \left(\frac{\text{dpi}}{\text{lpi}}\right)^2 + 1. \tag{2.2}$$

Halftoning também é comumente usado para imprimir imagens em cores. A ideia geral é a mesma: variando a densidade de impressão das quatro cores primárias, ciano, magenta, amarelo e preto (CMYK),

várias tonalidade de cor podem ser reproduzidas.

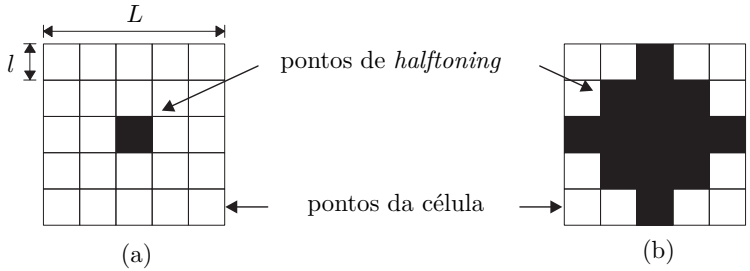


Figura 2.5: Células de *halftone* de dimensões 5×5 pontos para tons de cinza.

Pode-se criar um *halftoning* usando as mesmas técnicas utilizadas para impressão de tons de cinza, mas neste caso para manter-se os pontos com cores diferentes de impressão fisicamente próximos uns dos outros para enganar os olhos, fazendo-os perceber uma única cor. A indústria padronizou um conjunto de ângulos conhecidos, que resultam em pontos que formam círculos ou rosetas. Os pontos não podem ser facilmente vistos a olho nu, mas podem ser percebidos através de um microscópio ou uma lupa. Na Figura 2.6 é apresentado um exemplo com o *halftoning* para uma tonalidade da cor azul.

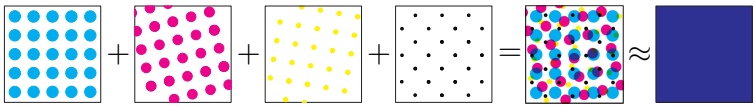


Figura 2.6: Exemplo de um *halftoning* para cor azul.

2.2 Modelo ponto×ponto para o processo *print and scan*

O modelo ponto×ponto para o processo *print and scan* apresentado a seguir foi proposto em [17]. Este modelo, ou simplificações dele, serviu como base para o desenvolvimento de vários trabalhos, tais como em [8–11]. Os valores das variações de luminância após o processo de impressão e digitalização para o ponto com coordenadas (x, y) pode ser

modelado como:

$$\begin{aligned} \hat{u}(x,y) &= g[w(x,y)] + \eta(x,y) \\ w(x,y) &\triangleq u(x,y) * h(x,y) \\ \eta(x,y) &\triangleq f[g(w(x,y))]\eta_1(x,y) + \eta_2(x,y) \end{aligned} \tag{2.3}$$

em que $u(x,y)$ representa a imagem original e $\hat{u}(x,y)$ é a imagem na saída do canal. Na Figura 2.7 é apresentado um diagrama de blocos do modelo para o processo *print and scan*. A resposta ao impulso $h(x,y)$ modela o sistema linear correspondente ao conjunto impressora e *scanner*:

$$h(x,y) = h_p(x,y) * h_s(x,y) \tag{2.4}$$

em que h_p é a resposta ao impulso da impressora e h_s é a resposta ao impulso do *scanner*, que considera as filtragens passa-baixas ótica e aquela devida tanto ao movimento como às interações locais entre elementos adjacentes na matriz de CCDs. Todas as filtragens irão gerar interferência intersimbólica (ISI) entre pulsos (pontos) sucessivos que são transmitidos (impressos) ao longo do papel. O grau de desfocagem introduzido pode ser julgado pela forma e largura da função de resposta ao impulso.

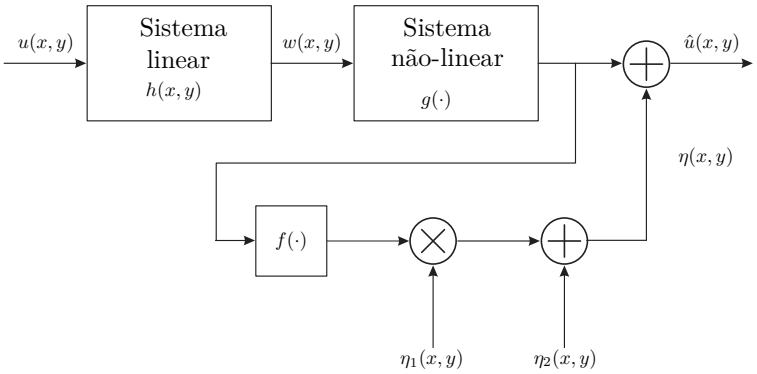


Figura 2.7: Diagrama de blocos do modelo ponto×ponto para o processo *print and scan*.

As funções $f(\cdot)$ e $g(\cdot)$ são geralmente não lineares e representam as características do mecanismo de detecção. A resposta $g(\cdot)$ dos detectores de imagem é tipicamente escrita como:

$$g(w) = \alpha w^{\beta}, \tag{2.5}$$

em que α e β são constantes dependentes dos dispositivos. No entanto, enquanto se está operando na região linear do detector, pode-se simplificar a resposta $g(\cdot)$ por uma função linear.

O termo $\eta(x, y)$ representa o ruído aditivo, que tem uma componente aleatória $f[g(w)]\eta_1$ dependente da imagem e uma componente aleatória η_2 independente da imagem. Em sistemas foto-eletrônicos, como este, muitas vezes o ruído é modelado como:

$$\eta(x, y) = \sqrt{g(x, y)}\eta_1(x, y) + \eta_2(x, y), \tag{2.6}$$

em que η_1 e η_2 são ruídos Gaussianos com média zero e mutuamente independentes. O processo de detecção, realizado pelos CCDs, envolve uma emissão aleatória de elétrons que possuem uma distribuição de Poisson com o valor médio de g, o que provoca o termo dependente do sinal transmitido. Essa distribuição é aproximada por uma distribuição Gaussiana como um caso limite. O outro termo η_2 representa o ruído térmico, que pode ser modelado como um ruído branco Gaussiano.

Existem muitas outras fontes diferentes de ruídos que não foram consideradas. Por exemplo: ruídos relacionados ao *scanner* podem ser causados por uma flutuação na intensidade da luz e uma variação no movimento mecânico do scanner. A cada pulso de clock do *scanner* uma variação indesejada significativa de tensão elétrica pode causar uma interferência no seu sinal analógico. Outro tipo de ruído é representado pelo ruído padrão fixo associado à matriz de CCDs. Por fim, outras distorções aleatórias podem surgir, devidas a poeira e arranhões no *scanner*, manchas durante a impressão e pela qualidade do papel usado.

Capítulo 3

Códigos de barra bidimensionais

Códigos de barras bidimensionais (2D) são largamente utilizados em várias aplicações devido às numerosas vantagens sobre as tecnologias alternativas. Além de serem baratos e simples, a sua principal vantagem é que eles podem transportar uma quantidade significativa de informação sobre superfícies como papel, plástico, ou qualquer outra superfície que permita a sua impressão. A impressão pode ser realizada utilizando tintas visíveis, ultravioletas ou infravermelho, dependendo da aplicação e da preocupação com a segurança. Alguns deles ainda podem ser lidos por leitores de baixa resolução, que possuem um baixo custo por serem equipados com os CCDs, tais como: *scanners*, câmeras fotográficas digitais, webcams, ou até mesmo câmeras de telefones celulares.

A ampla disseminação de impressoras e dispositivos de leitura de códigos de barra 2D torna essa tecnologia muito mais barata e atrativa para diversas aplicações multimídia. Por exemplo, códigos de barra 2D estão sendo utilizados em novas aplicações emergentes, como o *M-ticketing*, no qual um código de barra 2D recebido através de um telefone celular é equivalente a um bilhete de cinema, teatro ou até mesmo de uma passagem aérea [18]. Muitas outras aplicações podem adotar esse tipo de tecnologia, tais como: autenticação de documentos, selos postais, formulários fiscais, etc. Além disso, quando documentos são impressos inevitavelmente ocorre uma perda na qualidade dos documentos. Os códigos de barra 2D podem ser utilizados como um recurso auxiliar para transmitir informações adicionais sobre o documento com o objetivo de recuperar parte ou até mesmo todas as informações perdidas durante este processo.

3.1 Códigos de barra 2D preto e branco

Existem muitos padrões de códigos de barra 2D preto e branco em uso comercial. Na Figura 3.1 são apresentados quatro exemplos de padrões comerciais de códigos de barra 2D bastante populares; destes, apenas o Maxicode não é de domínio publico. Alguns desses códigos podem até mesmo ser lidos por câmeras de baixa resolução como as presentes em aparelhos de telefonia celular. Entretanto, considerando o *scanner* como dispositivo de leitura, é possível armazenar mais informação, devido à maior resolução, que permite que os símbolos 2D possam ocupar uma área menor e ainda apresentar níveis de distorção aceitáveis. Nesta tese vamos apenas considerar códigos de barra 2D submetidos ao canal PS.

O padrão de códigos de barra 2D PDF417 possui alta densidade de armazenamento de dados. Um código PDF417 pode ser imaginado como vários códigos de barra lineares empilhadas um sobre os outros. Essa é a razão pela qual o padrão PDF417 é às vezes chamado de "simbologia linear empilhada". A forma geral de um PDF417 é retangular. Os bits são modulados pela largura das barras ou pelos espaços entre elas. Devido a essa forma de modulação, para que o PDF417 tenha símbolos com alta qualidade são exigidas uma impressão de precisão e uma alta resolução. A densidade máxima de armazenamento de informação é de aproximadamente 686 $bytes/in^2$.

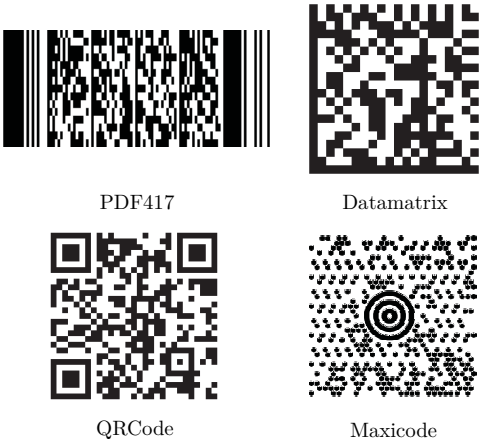


Figura 3.1: Exemplos de códigos de barra 2D preto e branco comerciais.

O padrão DataMatrix também possui alta densidade de

armazenamento de dados. Esses códigos de barra 2D podem aparecer em formatos quadrados ou retangulares. Cada bloco quadrado (um símbolo DataMatrix) modula um bit. Isto está em contraste com o formato de modulação utilizado pelos códigos PDF417, que utilizam a mesma modulação dos códigos de barra lineares na qual os bits são modulados em proporção à largura das barras ou em relação ao espaçamento entre elas. No DataMatrix, normalmente um quadrado preto (um símbolo 2D) é equivalente ao bit de valor 1, mas DataMatrix também podem ser impresso em branco sobre preto. Possuem densidade máxima de armazenamento de informação aproximadamente igual a 1687 $bytes/in^2$. O padrão QRCode é bastante semelhante ao DataMatrix. Também possui alta densidade de armazenamento de dados. Cada bloco quadrado (um símbolo QRCode) também representa um bit. Possuem densidade máxima de armazenamento de informação aproximadamente igual a 2122 $bytes/in^2$.

No padrão MaxiCode os símbolo 2D são hexagonais. Embora a capacidade de armazenamento de um MaxiCode seja bem inferior à dos outros padrões de códigos de barra 2D (aproximadamente igual a 88 $bytes/in^2$), ele foi projetado para ser lido em alta velocidade e independente da orientação em que foi digitalizado. Foi idealizado principalmente para armazenar dados de endereços para serviços postais.

3.2 Modelos para o canal PS para códigos de barra 2D preto e branco

Nesta seção serão apresentados modelos para o canal PS presentes na literatura [8, 19, 20]. Estes foram especificamente desenvolvidos para códigos de barra 2D preto e branco, que são projetados para alcançar alta capacidade de armazenamento. Os modelos utilizam impressoras comerciais preto e branco que usam *halftoning* e *scanners* com tecnologia CCD com resolução máxima de 600 dpi. Consideram-se uma perfeita sincronização entre os símbolos 2D e ausência de interferência intersimbólica.

O modelo para o canal PS ponto×ponto que foi apresentado na Seção 2.2 não é adequado para o caso de códigos de barra 2D. Os dois motivos principais para isso são os seguintes. Em primeiro lugar, os parâmetros para o modelo de canal PS ponto×ponto são muito difíceis de se avaliar. Em segundo lugar, um modelo adequado deve considerar os processos de modulação e demodulação, em outras palavras, o

modelo desejado deve avaliar o processo *print and scan* levando em conta que a informação está associada a um bloco de pontos e não a um único ponto.

Antes de descrever a modulação, alguns conceitos são importantes. Considere que uma imagem possui uma resolução igual a r_{im}, a impressora uma resolução nominal igual a r_p e o *scanner* uma resolução nominal igual a r_s. Para não existir perda de pontos, é necessário que $r_{im} \leqslant r_p$. Entretanto, somente essa condição não garante uma reprodução fiel da imagem. Se a resolução da imagem não for proporcional à resolução da impressora não é possível estabelecer uma equivalência entre os pontos da imagem original e os pontos da imagem impressa. Para não criar confusão entre essas resoluções, considerar-se-á que a resolução da imagem seja igual à resolução de impressão ($r_{im} = r_p$). Essa consideração não é válida quando se utiliza *halftoning*. Mas neste caso pode-se considerar hipoteticamente que a imagem original já era uma imagem em *halftoning*.

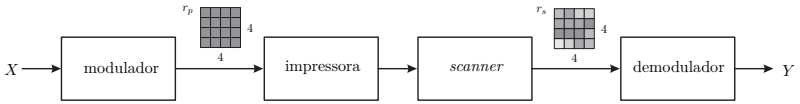


Figura 3.2: Modulação e demodulação para o canal de comunicação PS.

O esquema de modulação utilizado em [19, 20] considera uma resolução de 600 dpi na impressora e no *scanner* ($r_p = r_s$). Cada bloco quadrado (símbolo 2D) é composto por 16 pontos (4×4 pontos) com um espaçamento entre os blocos de 2 pontos. O diagrama de blocos dos processos de modulação e demodulação é apresentado na Figura 3.2. Considerando esse sistema de modulação e demodulação, nas próximas seções serão apresentados três modelos para o canal PS preto e branco que os autores de [19] desenvolveram baseados em seus resultados experimentais.

3.2.1 Modelo com o ruído Gaussiano independente do símbolo transmitido

Esse primeiro modelo considera que o ruído é Gaussiano e independente do símbolo transmitido. Um diagrama do modelo é apresentado na Figura 3.3 e ele é representado matematicamente por:

$$Y = \varphi(X) + Z, \tag{3.1}$$

em que normalmente X (entrada do canal) é uma variável aleatória discreta uniformemente distribuída em algum subconjunto de $\mathcal{X} = [0, 255]$, que representa todos os valores de cinza de um símbolo 2D, $\varphi : \mathcal{X} \to \mathbb{R}$ é uma função não-linear que representa a resposta do canal PS, Z representa o ruído Gaussiano aditivo de média zero e variância σ_Z^2 e Y (saída do canal) representa o valor obtido de tom de cinza para o correspondente símbolo 2D. Os valores de cinza são representados como números do conjunto $\{0, 1, \ldots, 255\}$. A função $\varphi(\cdot)$ geralmente é diferente para cada canal PS, em outras palavras, é diferente para cada combinação de impressora e *scanner*.

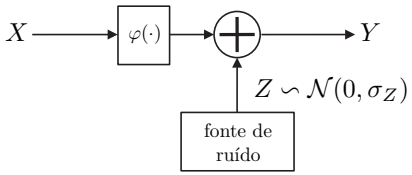


Figura 3.3: Diagrama para o modelo de canal PS com ruído Gaussiano independente do símbolo transmitido.

Para sistemas práticos baseados nesse modelo, a resposta do canal $\varphi(\cdot)$ é aproximada pela média da amostra, ou seja, $\varphi(x) = \hat{\mu}_{Y|X}(x)$, e ao desvio padrão do ruído é atribuído o máximo desvio padrão das amostras, ou seja, $\sigma_Z = \max_x[\hat{\sigma}_{Y|X}(x)]$.

3.2.2 Modelo com o ruído Gaussiano dependente do símbolo transmitido

Nesse modelo, contrariamente ao que é habitualmente suposto, supõe-se que o ruído Z não é independente do tom de cinza X. Isso está de acordo com o modelo ponto×ponto apresentado na Seção 2.2. Segundo os resultados experimentais obtidos pelos autores de [19], o ruído é dependente do símbolo transmitido e pode ser modelado como um ruído Gaussiano de média zero e desvio padrão $\sigma_Z(x)$. Um diagrama de blocos do modelo é apresentado na Figura 3.4

Para sistemas práticos baseados nesse modelo, a resposta do canal $\varphi(\cdot)$ é aproximada pela média da amostra $\hat{\mu}_{Y|X}(\cdot)$, ou seja, $\varphi(x) = \hat{\mu}_{Y|X}(x)$, e o desvio padrão do ruído $\sigma_Z(\cdot)$ é aproximado pelo desvio padrão da amostra, ou seja, $\sigma_Z(x) = \hat{\sigma}_{Y|X}(x)$.

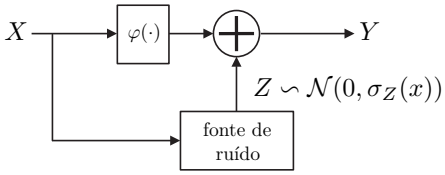


Figura 3.4: Diagrama para o modelo de canal PS com ruído Gaussiano dependente do símbolo transmitido.

3.2.3 Modelo com o ruído Gaussiano generalizado dependente do símbolo transmitido

Este terceiro modelo de canal foi considerado o mais apropriado pelos autores de [19] segundo os resultados experimentais obtidos por eles. Nesse modelo, o ruído do canal (Z) possui uma distribuição Gaussiana generalizada ($\mathcal{GGD}$), também dependente da entrada do canal (X). Uma distribuição Gaussiana generalizada é representada por três parâmetros: média, desvio padrão e fator de forma. Uma breve explicação sobre a $\mathcal{GGD}$ é apresentada no Apêndice C. O diagrama de blocos para esse modelo está apresentado na Figura 3.5 e, seguindo o mesmo procedimento realizado para o modelo anterior, $(Z|X = x)$ é modelado como:

$$(Z|X = x) \sim \mathcal{GGD}(0, \sigma_Z(x), \gamma_Z(x)) \tag{3.2}$$

em que, $\sigma_Z(x)$ e $\gamma_Z(x)$ são, respectivamente, o desvio padrão e o fator de forma do ruído dado que $X = x$.

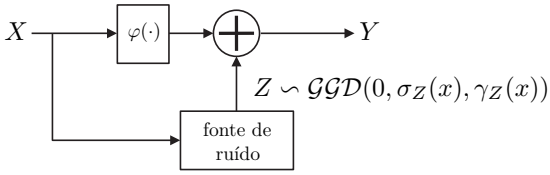
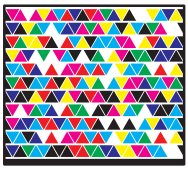


Figura 3.5: Diagrama para o modelo de canal PS com ruído Gaussiano generalizado dependente do símbolo transmitido.

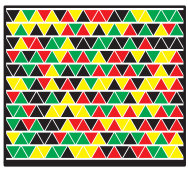
Lembre-se que para sistemas práticos a resposta do canal $\varphi(\cdot)$ e o desvio padrão do ruído $\sigma_Z(\cdot)$ são aproximados como na Seção 3.2.2, e o fator de forma do ruído $\gamma_Z(\cdot)$ é aproximado pela amostra do fator de forma, ou seja, $\gamma_Z(x) = \hat{\gamma}_{Y|X}(x)$.

3.3 Códigos de barra 2D coloridos

O estudo de códigos de barra 2D coloridos é de grande interesse comercial. Recentemente, a Microsoft divulgou duas vertentes de seu padrão de códigos de barra 2D coloridos [21]. Na Figura 3.6 são apresentados dois códigos de barra 2D no padrão desenvolvido pela Microsoft, um utilizando 4 cores, e outro, 8 cores. Os símbolos 2D são triangulares e possuem um espaçamento branco entre eles, mas no caso do padrão com 8 cores a cor branca também é utilizada para representar um dos símbolos. Em testes de laboratório, a Microsoft conseguiu atingir uma densidade de armazenamento de informação igual a 2000 $bytes/in^2$ utilizando impressoras e *scanners* com 600 dpi.



HCCB (Microsoft) 8 cores



HCCB (Microsoft) 4 cores

Figura 3.6: Padrão de códigos de barra 2D coloridos desenvolvido pela Microsoft.

Na seção a seguir será apresentado o modelo proposto para o canal PS colorido, especificamente desenvolvido para códigos de barra 2D utilizando 4 cores. O objetivo é alcançar uma alta capacidade de armazenamento com esses códigos de barra quando submetidos ao canal PS sob esse modelo. O modelo proposto considera o uso de impressoras comerciais coloridas *halftoning* e *scanners* com tecnologia CCD com resolução maxima de 600 dpi, bem como uma perfeita sincronização entre os símbolos 2D, ausência de interferência intersimbólica e uso somente de cores puras (sem uso de *halftoning*).

3.4 Modelo proposto de canal PS para códigos de barra 2D coloridos

O modelo proposto de canal PS para códigos de barra 2D coloridos trata o canal PS como um canal de comunicação. A informação submetida ao canal é modulada em blocos quadrados (símbolos 2D) formados por 25 pontos (5×5 pontos). Somente as cores primárias

das impressoras são utilizadas. Desta forma, as cores que os símbolos 2D podem ter são ciano, magenta, amarelo ou preto. Como se está utilizando cores, os pontos da imagem transmitida são compostos pelos valores de intensidade das três cores básicas no sistema CMY* de cores. Esse sistema foi o escolhido por ser o sistema de cores utilizado pelas impressoras comerciais com 4 cores. Os pontos (*pixels*) na entrada do canal PS podem ser definidos como vetores com três componentes, uma para cada intensidade de cor:

$$\boldsymbol{x}(i,j) = [c\ m\ y]^T, \tag{3.3}$$

em que os inteiros i e j representam as coordenadas do ponto na imagem.

O código de barra 2D colorido é representado por uma matriz $\boldsymbol{X}$ com l linhas e c colunas na qual cada elemento $\boldsymbol{x}(i,j)$ corresponde a um vetor com três dimensões (uma dimensão para cada componente do sistema de cores). Cada conjunto de 25 pontos (5×5 pontos) da matriz $\boldsymbol{X}$ corresponde a um símbolo 2D que é transmitido.

Na saída do canal PS, a imagem digitalizada passa por um processo no qual se determina o começo e o fim dos símbolos 2D transmitidos. Esse processo, chamado de segmentação [22], não será abordado nesta tese. Considera-se que a segmentação foi bem sucedida.

Cada ponto (*pixel*) recebido na saída do canal PS consiste em um vetor cujas componentes são as intensidades das três cores básicas utilizadas pela impressora. O *scanner* nos fornece para cada ponto um valor quantizado em 24 bits, 8 bits para cada intensidade de cor primária. Portanto, obtém-se um valor inteiro que varia de 0 a 255 para cada intensidade de cor no sistema de cores RGB (vermelho, verde e azul). Esses valores podem ser facilmente convertidos para o sistema de cores CMY (ciano, magenta e amarelo), pela equação 2.1, apresentada na seção 2.1.1, e por conveniência reproduzida a seguir:

$$[\hat{c}\ \hat{m}\ \hat{y}]^T = [255\ 255\ 255]^T - [\hat{r}\ \hat{g}\ \hat{b}]^T.$$

Assim, obtém-se um valor para cada ponto recebido com todas as distorções e ruídos, representado pelo vetor normalizado

$$\boldsymbol{y}(i,j) = \frac{1}{255}[\hat{c}\ \hat{m}\ \hat{y}]^T. \tag{3.4}$$

*As impressoras comerciais utilizam o sistema CMYK. O preto (K) é utilizado por representar melhor a cor e diminuir o custo de impressão. A cor preta no sistema CMY normalizado é dada por: K=[1, 1, 1]

Como já mencionado, durante o processo de modulação cada bloco colorido (símbolo 2D) corresponde a 25 pontos (*pixels*) na imagem original, todos de uma mesma cor. Para se obter o valor de um símbolo 2D recebido é necessário determinar o valor médio da intensidade do bloco de pontos recebido. Então, tem-se:

$$\boldsymbol{y} = \frac{1}{25} \sum_{i=i_o}^{i_o+5} \sum_{j=j_o}^{j_o+5} \boldsymbol{y}(i,j), \tag{3.5}$$

em que i_o e j_o indicam as posições iniciais (canto inferior esquerdo) do símbolo 2D na imagem recebida determinadas pelo processo de segmentação. O vetor $\boldsymbol{y}$ é um vetor tridimensional com componentes reais porque é obtido a partir da média de 25 vetores tridimensionais $\boldsymbol{y}(i,j)$ com componentes inteiras. Cada vetor $\boldsymbol{y}$ corresponde a um símbolo 2D transmitido através do canal. Na Figura 3.7 é apresentado um diagrama de blocos com todos os processos.

Figura 3.7: Diagrama de blocos para o modelo proposto do canal PS colorido.

Assumindo que as cores puras foram submetidas à entrada do canal, os possíveis valores normalizados para os vetores $\boldsymbol{x}$ são $[1\ 0\ 0]^T$ para a cor ciano, $[0\ 1\ 0]^T$ para a cor magenta, $[0\ 0\ 1]^T$ para a cor amarela e $[1\ 1\ 1]^T$ para a cor preta. Para se determinar qual é o melhor modelo estatístico para o canal PS foram realizados experimentos, nos quais vários conjuntos de impressoras e *scanners* foram utilizados. Uma sequência conhecida e longa de símbolos 2D foi submetida a esses diferentes canais PS. Com isso, foram estimados os momentos centrais de primeira a quarta ordem para o vetor aleatório $\boldsymbol{Y}$, lembrando que o primeiro momento corresponde à média, o segundo à variância, o terceiro à assimetria e o quarto à curtose.

A partir dos experimentos, observou-se que para cada uma das dimensões e natureza (cor) de símbolo 2D, os valores do vetor aleatório $\boldsymbol{Y}$ para os parâmetros de assimetria foram muito próximos de zero, e para os parâmetros de curtose foram muito próximos de 3, permitindo uma boa caracterização através de distribuições Gaussianas multidimensionais. Experimentos foram realizados para vários conjuntos de impressoras e *scanners*. Para diferentes conjuntos,

ou seja, para diferentes canais PS, foi observado que os valores das médias variam menos que os valores das matrizes de covariância. Apesar dessa variação, os canais são muito semelhantes.

Para se trabalhar com o desenvolvimento de códigos corretores de erros, optou-se pela adoção do pior conjunto de dispositivos, uma vez que o código que apresentar bom desempenho neste "pior" canal terá boas chances de ser bem sucedido para qualquer conjunto de impressora e *scanner*.

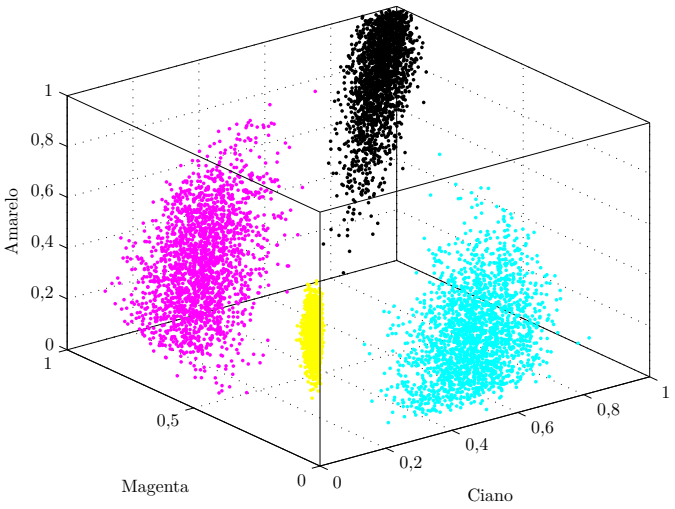


Figura 3.8: Amostras obtidas na saída do "pior" canal PS no cubo CMY.

Na Figura 3.8 é apresentado um conjunto de amostras obtidas na saída do "pior" canal PS. Semelhante a um histograma, para cada símbolo 2D obtido na saída do canal PS um ponto é plotado com a sua respectiva cor original no cubo CMY. Quatro nuvens são formadas, cada uma delas correspondendo a uma das cores. Pode-se perceber pelo formato das nuvens o comportamento assimétrico do canal. Como os experimentos indicam, o canal pode ser descrito estatisticamente como um caso particular de canal AWAMGN (*Additive White Asymmetric Multidimensional Gaussian Noise*); o canal PS para códigos de barra 2D considera o canal Gaussiano com três dimensões. Para o modelo de

canal Gaussiano Multidimensional Assimétrico tem-se:

$$p(\boldsymbol{y}|\boldsymbol{x}_i) = \frac{1}{(2\pi)^{3/2}|\boldsymbol{C}_i|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_i)^T \boldsymbol{C}_i^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_i)}, \tag{3.6}$$

em que o subscrito i indica uma das quatro cores (magenta, ciano, amarelo e preto).

O canal sintético adotado possui a média e a matriz de covariância do "pior" resultado obtido dos experimentos, como já mencionado. Mas, com o propósito de se avaliar o comportamento do canal PS para outras intensidades de ruído, foi definida uma relação entre a energia média dos símbolos 2D recebidos e a potência média do ruído, que foi considerada igual à média dos traços das matrizes de covariância. As matrizes utilizadas no canal sintético são:

$$\boldsymbol{C}_c = \begin{bmatrix} 0,0644 & 0,0406 & 0,0378 \\ 0,0406 & 0,0406 & 0,0392 \\ 0,0378 & 0,0392 & 0,0420 \end{bmatrix}, \quad \boldsymbol{\mu}_c = \begin{bmatrix} 0,6523 & 0,2483 & 0,0991 \end{bmatrix}^T,$$

$$\boldsymbol{C}_m = \begin{bmatrix} 0,0448 & 0,0336 & 0,0434 \\ 0,0336 & 0,0490 & 0,0350 \\ 0,0434 & 0,0350 & 0,0462 \end{bmatrix}, \quad \boldsymbol{\mu}_m = \begin{bmatrix} 0,1952 & 0,7352 & 0,3739 \end{bmatrix}^T,$$

$$\boldsymbol{C}_y = \begin{bmatrix} 0,0001 & 0,0001 & 0,0001 \\ 0,0001 & 0,0011 & -0,0004 \\ 0,0001 & -0,0004 & 0,0114 \end{bmatrix}, \quad \boldsymbol{\mu}_y = \begin{bmatrix} 0,0002 & 0,0170 & 0,5048 \end{bmatrix}^T,$$

$$\boldsymbol{C}_p = \begin{bmatrix} 0,0630 & 0,0574 & 0,0560 \\ 0,0574 & 0,0574 & 0,0546 \\ 0,0560 & 0,0546 & 0,0546 \end{bmatrix}, \quad \boldsymbol{\mu}_p = \begin{bmatrix} 0,9370 & 0,9468 & 0,9501 \end{bmatrix}^T.$$

A energia média dos símbolos foi definida como:

$$E_s = \frac{1}{4}\left(\boldsymbol{\mu}_c^T\boldsymbol{\mu}_c + \boldsymbol{\mu}_m^T\boldsymbol{\mu}_m + \boldsymbol{\mu}_y^T\boldsymbol{\mu}_y + \boldsymbol{\mu}_p^T\boldsymbol{\mu}_p\right), \tag{3.7}$$

e a potência média do ruído foi definida como:

$$E_n = \frac{1}{4}\left[\text{traço}(\boldsymbol{C}_c) + \text{traço}(\boldsymbol{C}_m) + \text{traço}(\boldsymbol{C}_y) + \text{traço}(\boldsymbol{C}_p)\right]. \tag{3.8}$$

Um parâmetro ξ foi criado para fornecer diferentes razões sinal-ruído (SNRs), tal que:

$$(\mathrm{SNR})_{\mathrm{dB}} = 10 \log \left(\frac{E_s}{\xi E_n} \right). \tag{3.9}$$

Quando a SNR for igual a 0 dB[†], significa que se está considerando o "pior" canal adotado como referência para o desenvolvimento dos códigos.

O modelo de canal proposto para códigos de barra 2D coloridos foi apresentado em [23, 24]. Nestes trabalhos, são apresentados, além do modelo de canal, detalhes sobre os experimentos. Os códigos corretores de erros utilizados em [23, 24] foram os códigos de bloco BCH (Bose, Chaudhuri and Hocquenghem) e RS (Reed-Solomon).

3.4.1 Exemplo de um código de barra 2D colorido

Nesse modelo de código de barra adotado os símbolos utilizados são blocos quadrados coloridos de dimensões $(1/120) \times (1/120)$ polegadas, impressos em papel branco e utilizando uma resolução de impressão de 600 dpi. Como consequência, os blocos correspondem a uma matriz de 5×5 pontos (*pixels*), como já observado anteriormente. Os blocos são separados entre si por uma distância de 1/300 polegadas (ou 2 *pixels*). Na Figura 3.9 é apresentada uma amostra ampliada de um código de barra 2D ainda sem os efeitos do canal PS e, na Figura 3.10, é mostrado o mesmo código na saída do canal PS. Olhando as duas figuras, é possível observar diversos efeitos do canal PS, tais como: perda de blocos, atenuação na intensidade das cores, espalhamento da tinta, entre outras distorções.

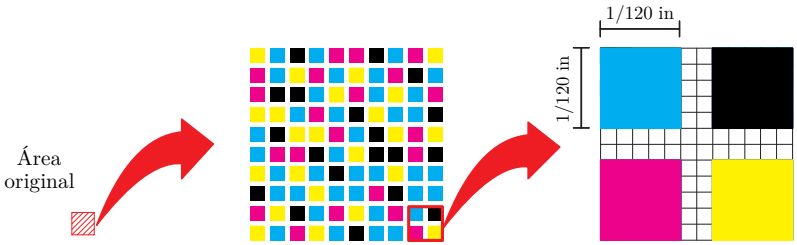


Figura 3.9: Amostra de um código de barra 2D.

[†]O canal PS utilizado como referência corresponde a $\xi = 1$, que resulta numa SNR igual a 9,41 dB. Por exemplo, para o valor de SNR normalizada igual a -1 dB corresponde a uma SNR igual a 8,41 dB.

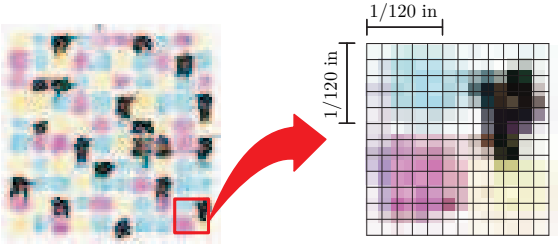


Figura 3.10: Amostra de um código de barra 2D na saída do canal PS.

Capítulo 4

Modulação codificada

Neste capítulo serão apresentados os principais conceitos sobre a modulação codificada. Estes conceitos são válidos independentemente da constelação de sinais utilizada na modulação.

4.1 Sistema de comunicações digitais com modulação codificada

Em um sistema de comunicação digital não codificado, para se aumentar a taxa de transmissão da informação precisa-se aumentar o número de sinais da constelação no modulador. Mas para manter a energia de transmissão, isso traz como consequência uma aproximação dos possíveis sinais a serem transmitidos. Isto refletir-se-á numa maior probabilidade de erro e, por conseguinte, numa degradação da qualidade de serviço da aplicação. Aumentar a potência de transmissão pode reduzir essa probabilidade de erro, mas a maioria das aplicações práticas tem restrição de potência.

Alternativamente, para se reduzir a probabilidade de erros, pode-se adotar códigos corretores de erros clássicos. Mas isso leva a uma redução na taxa de transmissão ou uma expansão espectral (devidas à adição de redundância). Em sistemas com restrição de largura de faixa de frequências, portanto, códigos clássicos não podem ser adotados.

Um exemplo já bem conhecido de canal com restrição na taxa é o canal de satélite móvel, em que a largura de faixa de um usuário é restrita pelo desejo de acomodar um maior número de usuários numa determinada largura de faixa de transmissão total, e a potência também

é restrita em função das grandes distâncias envolvidas.

4.1.1 Modulação codificada em treliças

Mediante a combinação de códigos convolucionais com diferentes constelações de sinais de modulação, Ungerboeck [25, 26] construiu códigos com capacidade de correção de erros que permitem, com grande eficiência, transmissão confiável de dados em situações de largura de faixa limitada. Tal esquema é conhecido como modulação codificada em treliças (TCM) *Trellis Coded Modulation*, devido à utilização de treliças para representar as sequências de sinais da constelação a serem transmitidos. Desde a publicação do artigo [25], houve um grande número de pesquisas na construção de códigos TCM [27–34].

Em TCM, a constelação de sinais do modulador possui o dobro do número de sinais da constelação de referência. Por exemplo, se deseja-se alcançar 2 bits por símbolo, no caso não codificado uma constelação com 4 sinais é requerida. Em TCM, para os mesmos 2 bits de informação por símbolo, adota-se uma constelação com 8 sinais. Esse excedente de recursos é tudo de que se precisa para se inserir a redundância, necessária ao bom desempenho de erro, sem nenhum custo em termos de taxa ou de largura de faixa. A constelação expandida é primeiramente dividida em subconjuntos, formando uma cadeia de partições, de tal maneira que os sinais de um subconjunto em um nível da partição tenham separação mínima maior do que os de um subconjunto em um nível anterior. Um codificador convolucional é então utilizado, e parte dos bits de informação é codificada. Os bits codificados, na saída do codificador, são usados para escolher um dos subconjuntos formados em um certo nível de partição. Os bits de informação restantes servem para escolher um determinado sinal do subconjunto escolhido no passo anterior.

No receptor dos sistemas TCM, o sinal recebido, ao invés de ser primeiramente demodulado para então ser decodificado, como no sistema de codificação tradicional, é processado diretamente pelo receptor, que combina demodulação com decodificação num processo só. Como consequência disso, em sistemas TCM para o canal Gaussiano o parâmetro que governa o desempenho não é mais a distância de Hamming do código convolucional, mas sim a distância Euclidiana entre as sequências transmitidas. Portanto, a escolha do código e da constelação de sinais não poderá ser feita separadamente.

A modulação codificada pode prover ganhos de codificação significativos se comparada aos esquemas de codificação convencional.

Os ganhos podem ser de 3-5 dB com uma complexidade de decodificação razoável, e podem chegar a até 6 dB com alta complexidade de decodificação.

4.1.2 Mapeamento dos bits

A grande inovação do sistema proposto por Ungerboeck reside na maneira como os m bits de informação são mapeados nos sinais da constelação. Tal mapeamento é feito através da técnica de *partição de conjuntos*, proposta por Ungerboeck [25] em 1982. A maioria das constelações de sinais pode ser particionada de forma sistemática para formar uma série de subconjuntos menores. Se a partição é bem feita, as subconstelações resultantes têm uma distância Euclidiana mínima maior do que a constelação original.

Em qualquer projeto de modulação codificada, a partição da constelação de sinais do modulador em subconjuntos com distâncias mínimas internas maiores tem um papel principal, pois define o mapeamento a ser usado pelo modulador e provê um limitante aproximado da mínima distância Euclidiana entre duas sequências do código.

4.2 Partição de conjuntos

Seja $\mathbf{\Lambda}_0$ o conjunto de símbolos utilizados no esquema de modulação. A regra de partição de conjuntos consiste em dividir o conjunto $\mathbf{\Lambda}_0$ em l_0 subconjuntos disjuntos $\mathbf{\Lambda}_1(i)$, levando em consideração os seguintes critérios:

- $\mathbf{\Lambda}_1(i) \subseteq \mathbf{\Lambda}_0 \quad \forall \quad i = 1, \ldots, l_0$.
- A distância Euclidiana quadrática mínima ($d^2_{E,1}$) de $\mathbf{\Lambda}_1(i)$ para $i = 1, \ldots, l_0$, é maior ou igual à distância Euclidiana quadrática mínima de $\mathbf{\Lambda}_0$, $d^2_{E,0}$.
- Se as distâncias Euclidianas quadráticas forem iguais, o número de sinais que distam entre si de d^2_E em $\mathbf{\Lambda}_1(i)$ deve ser menor do que em $\mathbf{\Lambda}_0$.
- Uma vez finalizada a primeira partição, rotulam-se os l_0 subconjuntos $\mathbf{\Lambda}_1$ obtidos com valores do alfabeto $\{0, 1, \ldots, l_0-1\}$.

O procedimento é repetido para cada um dos subconjuntos $\mathbf{\Lambda}_1(i)$. O processo só termina quando, depois de m níveis de partição, a cardinalidade dos subconjuntos resultantes for igual a 1 ($|\mathbf{\Lambda}_m|$=1).

Finalmente, cada sinal no último nível da partição será rotulado com um vetor m-dimensional, cujas componentes correspondem aos rótulos

dos subconjuntos nas m partições que contêm o sinal.

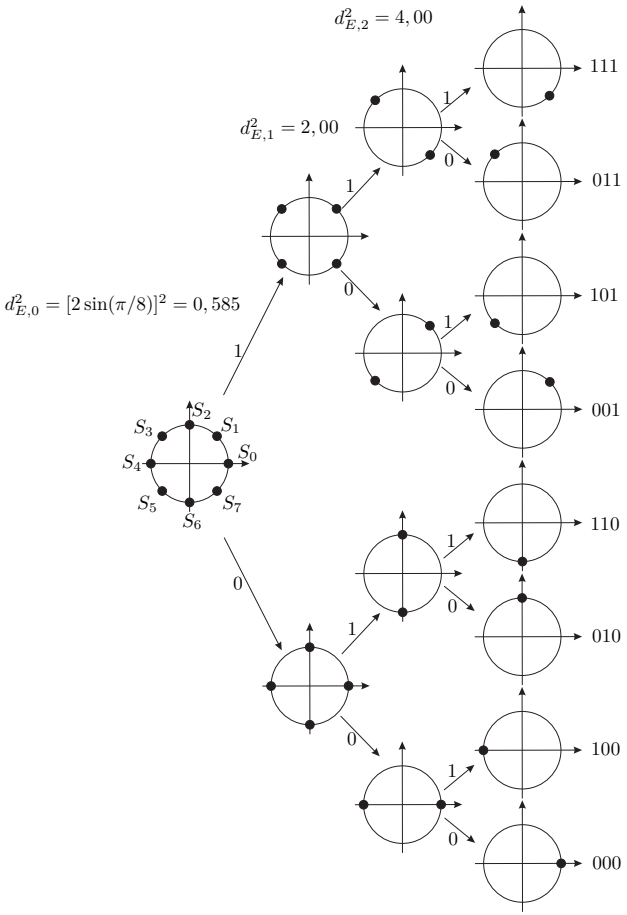


Figura 4.1: Partição da constelação 8-PSK.

Na Figura 4.1 é apresentada a aplicação da regra de partição de Ungerboeck para uma constelação 8-PSK. Cada um dos sinais da constelação é representado (rotulado) por uma sequência de bits conforme o esquema de partição. O rótulo dos símbolos é apresentado abaixo de cada um dos sinais resultantes da última partição. A d^2_E é apresentada ao lado direito da constelação particionada, e representa a distância Euclidiana quadrática mínima para cada partição da constelação 8-PSK.

4.3 Códigos convolucionais

Os códigos convolucionais possuem memória, portanto os bits codificados dependem não só dos bits de informação atuais, mas também da informação armazenada pela memória do código. O comprimento da memória do código (*constraint length*) representa a dependência da saída face aos bits de entrada, permitindo caracterizar a eficiência e a complexidade do código.

O processo de codificação convolucional é realizado passando-se os símbolos de informação recebidos da fonte de informação por um circuito sequencial composto por ν registradores (memórias) e por somadores módulo dois. Estes componentes são dispostos adequadamente a fim de produzir n equações algébricas geradoras. A partir de k bits de entradas são gerados n bits de saída, assim o código possui taxa $R = k/n$.

Nos codificadores utilizados por Ungerboeck cada bloco de k bits gerados pela fonte é mapeado em um sinal de uma constelação com 2^{k+1} símbolos. A taxa dos códigos convolucionais utilizados por Ungerboeck é portanto igual a $R = k/(k+1)$.

A probabilidade de erro em um canal AWGN é uma função da distância Euclidiana entre sinais (representados no espaço dos sinais). Chamamos a distância Euclidiana mínima entre pares de sequências de símbolos de distância livre mínima (d_{free}). O codificador convolucional utilizado na modulação codificada deve ser projetado de forma a maximizar a sua distância livre.

Seguindo o sistema de partição explicado anteriormente e apresentado na Figura 4.1, uma constelação 8-PSK é utilizada para transmitir uma quantidade de informação que poderia ser transmitida utilizando uma constelação 4-PSK (QPSK).

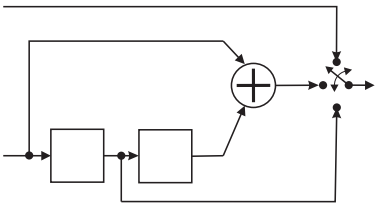


Figura 4.2: Codificador convolucional de Ungerboeck com 4 estados.

Um codificador convolucional com taxa $R = k/n$, ou seja, com k entradas e n saídas, e com quantidade de memória ν pode ser

referenciado de forma sintética como (k, n, ν). Na Figura 4.2 é apresentado um codificador convolucional de taxa 2/3 e com duas memórias ($\nu = 2$). O conjunto de todas as sequências formadas na saída do codificador, a partir de todas as possíveis sequências de informação, é denominado o *código convolucional*. Associado a um código convolucional tem-se um diagrama denominado treliça. A treliça representa um diagrama temporal das transições possíveis entre estados do codificador, gerando um gráfico das sequências codificadas transmitidas. O número de estados na treliça que representa um código convolucional binário com ν memórias é 2^{ν}. A Figura 4.3 mostra a representação do código convolucional gerado pelo codificador da Figura 4.2 utilizando um diagrama de treliça, em que os rótulos no formato $uu/vvv/S_i$ representam dois bits de informação (uu), três bits codificados (vvv) e sinal (S_i) correspondente da constelação 8-PSK seguindo o mapeamento por partição de conjunto apresentado na Figura 4.1.

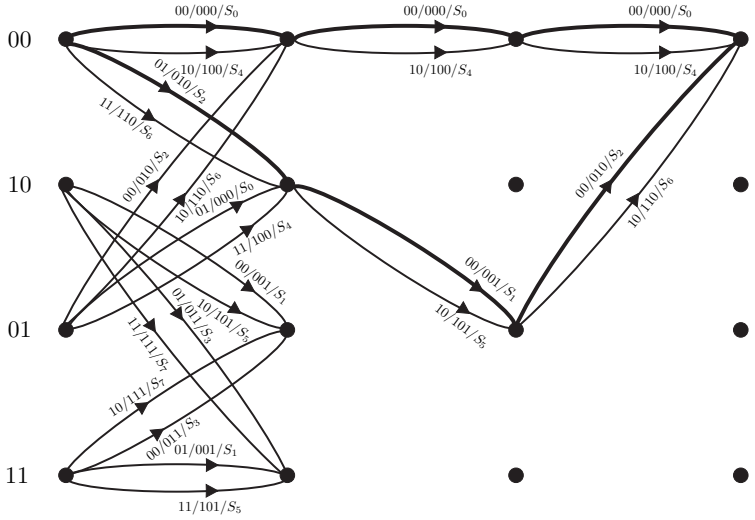


Figura 4.3: Diagrama de treliça para o código convolucional. Dois caminhos concorrentes são mostrados em destaque.

A distância mínima de Hamming entre duas sequências nesse diagrama é 1, obtida a partir de qualquer transição paralela entre um estado num instante de tempo atual e outro no próximo instante de tempo. Devem-se considerar os bits codificados quando se fala em distância de Hamming entre sequências codificadas. Observe que

a distância de Hamming entre os caminhos concorrentes realçados na figura também é pequena ($d_H = 3$), resultando num código convolucional muito ineficiente se aplicado ao canal binário. Porém, a distância Euclidiana quadrática entre os mesmos caminhos concorrentes é $d_E^2 = 4,585$, que é bastante grande, resultando num bom desempenho em um canal AWGN. Entretanto, o desempenho dessa modulação codificada não é limitado por essa distância. Note que, como já observado, nesse diagrama de treliça existem transições paralelas, devido à estrutura do codificador; para cada possível estado das memórias, existem duas possíveis sequências transmitidas resultando num mesmo estado futuro para as memórias do codificador. O desempenho dessa modulação codificada no canal AWGN é limitado pela distância Euclidiana quadrática entre os símbolos das transições paralelas, a saber, $d_E^2 = 4$. Mas isso ainda é uma distância elevada, pois é o dobro da distância Euclidiana quadrática mínima que limita o desempenho da modulação QPSK sem a utilização de códigos. Portanto, mesmo num exemplo simples como esse apresentado nesta seção, o ganho conseguido com a modulação codificada é de 3 dB, mantendo-se a mesma eficiência espectral de 2 bits/s/Hz (ou 2 bits por uso do canal) do QPSK não codificado.

Na Figura 4.4 é apresentado um gráfico da capacidade de transmissão de bits por uso do canal em função da relação sinal ruído (SNR). Também é apresentado o ponto no qual, para o caso de um sinal não codificado, a probabilidade de erro é igual a 10^{-5}. As curvas foram obtidas para as constelações de sinais: 2-PSK, 4-PSK e 8-PSK, seguindo o método explicado no Apêndice B.

Considere na Figura 4.4 a curva correspondente à modulação 4-PSK, na qual a $P_r(e) = 10^{-5}$ ocorre para uma SNR $= 12,9$ dB. Se dobrado o número de sinais utilizados, escolhendo-se a modulação 8-PSK, a transmissão de 2 bits/uso do canal, de forma confiável, já é teoricamente possível para uma SNR $= 5,9$ dB (assumindo ilimitados recursos de codificação e decodificação). Portanto, é vantajoso duplicar o número de sinais utilizados fazendo o uso de modulação codificada (assumindo uma SNR intermediária e o uso de um codificador de baixa complexidade), dado que não é possível atingir a probabilidade de erro desejada utilizando a modulação tradicional, sem presença de código.

A partir do exemplo de modulação codificada anterior, a transmissão de 2 bits por uso do canal com um 8-PSK (ao invés de QPSK) apresenta um ganho em SNR de 3 dB. Note, entretanto, que pelas curvas da Figura 4.4 há espaço para outros 4 dB de ganho. Ungerboeck mostrou que uma parte desse ganho pode ser alcançada

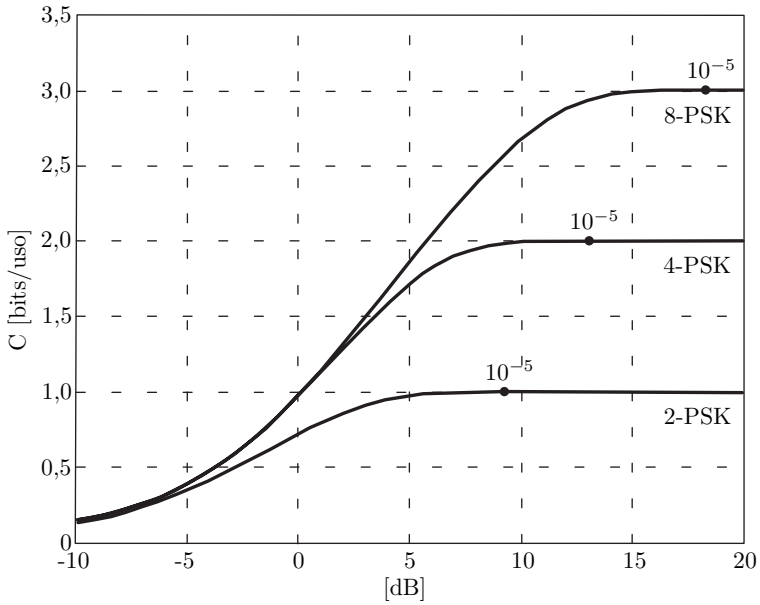


Figura 4.4: Capacidades para canais Gaussianos com modulações PSK.

se for aumentado o número de estados do codificador convolucional. A conquista definitiva da capacidade do canal AWGN só foi conseguida posteriormente com os códigos turbos e os códigos LDPC.

4.4 Modulação codificada multinível

O objetivo da modulação codificada multinível (MLC: do inglês "*Multilevel Coding*") é o mesmo da modulação codificada em treliça: otimizar conjuntamente codificação e modulação a fim de melhorar o desempenho dos sistemas de comunicação digital. A decodificação mais apropriada para a MLC é a decodificação multi-estágios (MSD, do inglês: "*Multistage Decoding*"). MLC/MSD é bem conhecida pela sua eficiência de banda para o canal AWGN em alta SNR [35]. Antes de descrever MLC/MSD, será apresentada a seguir uma análise da informação mútua do canal AWGN sujeito a uma modulação digital na sua entrada. Dessa análise, MLC/MSD decorrerá naturalmente.

Considere o modelo de canal AWGN discreto:

$$Y = X + Z, \qquad X \in \mathcal{X} \equiv \mathbb{R}, \qquad Y \in \mathcal{Y} \equiv \mathbb{R}, \qquad Z \sim \mathcal{N}(0, \sigma_Z^2),$$

em que X e Y são respectivamente a entrada e a saída do canal, $\mathcal{X}$ e $\mathcal{Y}$ são respectivamente o conjunto de entrada e de saída e Z representa o ruído. Para cada uso do canal, o ruído Z é independente e identicamente distribuído (i.i.d.) com distribuição Gaussiana de média zero e variância σ_Z^2, e é assumido como sendo independente da entrada X. Dada a variância na entrada do canal σ_X^2, que pode ser entendida como a potência média do sinal transmitido, a capacidade para este canal, em bits por uso do canal é:

$$C_{AWGN} = \max_{f_x(\cdot)} I(X;Y) = \frac{1}{2}\log_2(1 + \frac{\sigma_X^2}{\sigma_Z^2}),$$

que é atingida quando a entrada do canal X é Gaussiana, ou seja: $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$.

Devido à impossibilidade técnica de se utilizar uma entrada contínua qualquer ou uma entrada infinita, sistemas práticos geralmente utilizam um alfabeto de entrada discreto e finito $M = 2^L$-ário (constelação de sinais), ou seja, $|\mathcal{X}| = 2^L$. Por exemplo, na Figura 4.5 são apresentadas as pdfs condicionais para uma modulação 8-PAM ($L = 3$) com sinais equidistantes.

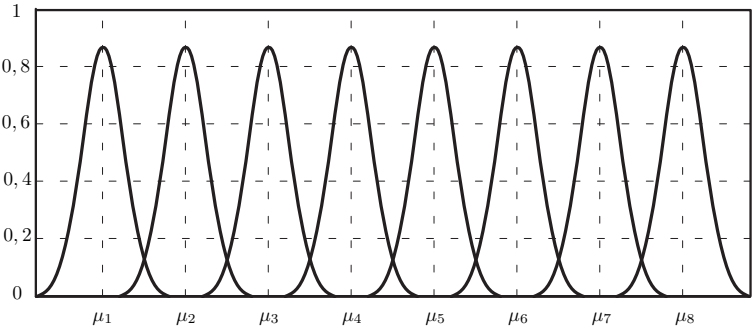


Figura 4.5: pdfs condicionais para 8-PAM.

É comum atribuir um rótulo binário para cada sinal da constelação por meio de um mapeamento bijetivo Ψ:

$$\Psi : (b^0, b^1, \ldots, b^{L-1}) \to x,$$
$$b^i \in \mathcal{B} = \{0,1\}, \quad x \in \mathcal{X}, \quad i = 0, 1, \ldots, L-1. \tag{4.1}$$

Dada uma distribuição de probabilidades específica $\{p(x) : x \in \mathcal{X}\}$ na entrada do canal, a máxima taxa de transmissão para comunicação

confiável é dada pela informação mútua $I(X;Y)$. Surpreendentemente, MLC/MSD [36, 37] é uma consequência da regra da cadeia para a informação mútua. Uma vez que o mapeamento de (4.1) é bijetivo, a informação mútua $I(X;Y)$ entre o sinal transmitido X e o sinal recebido Y é igual à informação mútua $I(b^0, b^1, \ldots, b^{L-1}; Y)$ entre os rótulos binários de X e Y. Aplicando-se a regra da cadeia para a informação mútua, obtém-se:

$$\begin{aligned} I(X;Y) &= I(b^0, b^1, \ldots, b^{L-1}; Y) \\ &= I(b^0;Y) + I(b^1;Y|b^0) + \ldots + I(b^{L-1};Y|b^0, b^1, \ldots, b^{L-2}). \end{aligned} \quad (4.2)$$

A equação (4.2) pode ser interpretada como a transmissão de vetores com dígitos binários b^i, $i \in \{0, 1, \ldots, L-1\}$, ao longo do canal físico que pode ser visto como a transmissão de bits individuais b^i em L canais equivalentes em paralelo, desde que $b^0, b^1, \ldots, b^{i-1}$ sejam conhecidos.

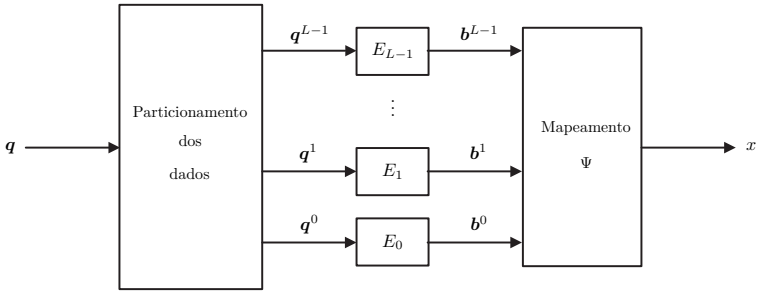


Figura 4.6: Codificador multinível.

Na Figura 4.6 é apresentado um diagrama de blocos para o codificador multinível. No lado do transmissor, um bloco de dados binários de comprimento K bits é particionado em L sub-blocos:

$$\begin{aligned} \boldsymbol{q} &= (q_1, \ldots, q_K), \qquad q_k \in \mathcal{B}, \qquad k = 1, \ldots, K, \\ \boldsymbol{q} &= (\boldsymbol{q}^0, \ldots, \boldsymbol{q}^{L-1}), \qquad \boldsymbol{q}^i = (q_1^i, \ldots, q_{K_i}^i), \\ i &= 0, 1, \ldots, L-1, \qquad \sum_{i=0}^{L-1} K_i = K. \end{aligned}$$

Cada sub-bloco de dados $\boldsymbol{q}^i$ é alimentado em um codificador binário indivídual E_i com taxa $R_i = K_i/N$ produzindo uma palavra código:

$$\begin{aligned} \boldsymbol{b}^i &= (b_1^i, \ldots, b_N^i), \qquad b_n^i \in \mathcal{B}, \\ n &= 1, \ldots, N, \qquad i = 0, 1, \ldots, L-1, \end{aligned}$$

com o correspondente código componente. Dessa forma, L níveis de codificação são criados. Por simplicidade, considera-se que todas as palavras código têm igual comprimento (N símbolos binários) em todos os níveis. Em seguida, o n-ésimo bit b_n^i, $n = 1, \ldots, N$, de cada palavra código $\boldsymbol{b}^i$ é selecionado para formar um rótulo binário $(b_n^0, b_n^1, \ldots, b_n^{L-1})$ de L bits, que é mapeado por Ψ para um sinal da constelação $x_n \in \mathcal{X}$. Assim, obtém-se um vetor:

$$\boldsymbol{x} = (x_1, \ldots, x_N), \qquad x_n \in \mathcal{X}, \qquad n = 1, \ldots, N, \tag{4.3}$$

com N entradas de canal, que são serialmente transmitidas pelo canal AWGN. A taxa geral do código, $R = K/N$, é igual à soma das taxas dos códigos individuais:

$$\sum_{i=0}^{L-1} R_i = \sum_{i=0}^{L-1} \frac{K_i}{N} = \frac{1}{N} \sum_{i=0}^{L-1} K_i = \frac{K}{N} = R. \tag{4.4}$$

No lado do receptor, os códigos componentes são sucessivamente decodificados pelo seus correspondentes decodificadores partindo-se do nível mais baixo. Em qualquer estagio i, $i = 0, 1, \ldots, L-1$, o processo de decodificação não envolve somente os N sinais recebidos:

$$\boldsymbol{y} = (y_1, \ldots, y_N), \qquad y_n \in \mathcal{Y}, \qquad n = 1, \ldots, N, \tag{4.5}$$

mas também as decisões dos estágios anteriores de decodificação:

$$\hat{\boldsymbol{b}}^j = (\hat{b}_1^j, \ldots, \hat{b}_N^j), \;\; \hat{b}_n^j \in \mathcal{B}, \;\; n = 1, \ldots, N, \;\; j = 0, 1, \ldots, i-1. \tag{4.6}$$

O Diagrama de blocos simplificado do decodificador multi-estágios é mostrado na Figura 4.7. Na Figura 4.8 é apresentado um diagrama de blocos simplificado para o decodificador com canais paralelos, que considera a decodificação de cada nível individualmente, não explorando a informação dos níveis inferiores. Isso resulta numa perda de informação, à medida que se desconsidera as decisões já efetuadas nos níveis inferiores. Entretanto, para alta SNR, esse decodificador mais simples pode ser adotado sem grande perda de desempenho.

Foi mostrado em [36] que o valor de taxa máximo com um esquema de modulação digital, com uma certa distribuição de probabilidades a priori para os símbolos da constelação, é alcançado por MLC/MSD se, e somente se, as taxas individuais R_i dos códigos componentes forem escolhidas para serem iguais às capacidades dos canais equivalentes, ou seja:

$$R_i = I(b^i; Y | b^0, b^1, \ldots, b^{i-1}), \;\; i = 0, 1, \ldots, L-1. \tag{4.7}$$

Essa é a chamada regra da capacidade para a escolha da taxa de código individual [36]. A regra da capacidade não indica, contudo, que códigos utilizar nos canais equivalentes. Na prática, uma vez que as estatísticas dos canais equivalentes $f(y|b^i, b^0, \ldots, b^{i-1})$ são não Gaussianas, não existem códigos ótimos conhecidos para estes canais. Tem-se, portanto, uma certa liberdade de escolha para os códigos individuais. Foi mostrado que códigos Turbo muito longos e códigos LDPC possuem um desempenho muito bom em MLC/MSD para canal AWGN [38]. Além disso, dado que a regra da capacidade é válida para qualquer rotulamento, não há restrição sobre um rotulamento particular a ser utilizado em MLC/MSD. No entanto, para palavras código com comprimento finito, o rotulamento de Ungerboeck apresenta o melhor desempenho em MLC/MSD se comparado a outros rotulamentos [36].

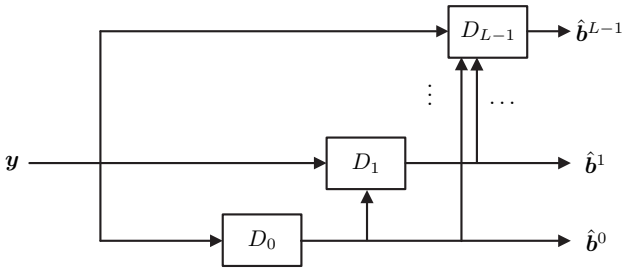


Figura 4.7: Decodificador multi-estágios.

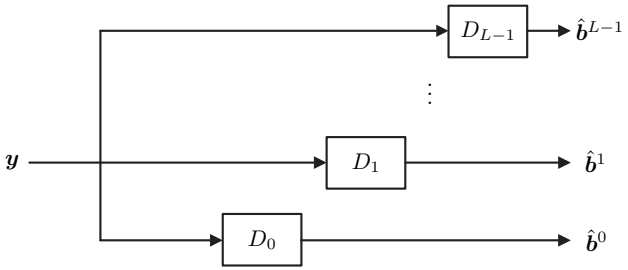


Figura 4.8: Decodificadores independentes em paralelo.

4.5 Modulação codificada com entrelaçamento de bit

A modulação codificada com entrelaçamento de bit (BICM) pode ser representada pelo diagrama de blocos da Figura 4.9. O sistema é composto por um codificador, um entrelaçador de bit, um modulador, canal, demodulador, desentrelaçador de bit e decodificador.

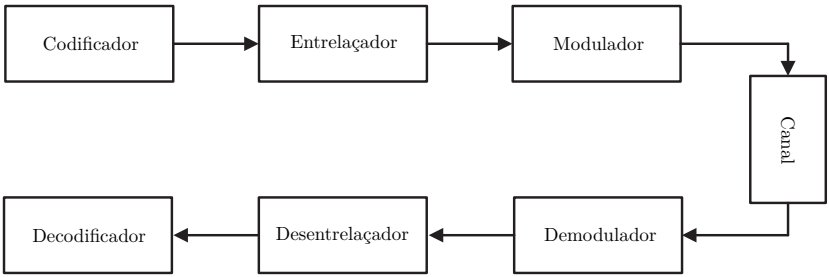


Figura 4.9: Diagrama de blocos com modulação codificada com entrelaçamento de bit.

A técnica foi originalmente proposta por Zehavi em [39]. BICM consiste em entrelaçar os bits que compõem a palavra código antes de realizar o processo de modulação que rotula os bits em símbolos para serem transmitidos pelo canal. No receptor, depois de demodular os símbolos, atribui-se a probabilidade de cada bit ser igual 0 ou 1, os bits são desentrelaçados e passam novamente a possuir uma relação bit a bit com a palavra código original. Este processo faz com que os bits que seriam mapeados em um símbolo e transmitidos em um uso do canal passem a ser transmitidos em símbolos diferentes e em diferentes usos do canal.

Capítulo 5

Códigos LDPC

Neste capítulo serão abordados os códigos LDPC (*Low-density parity-check*), que pertencem à classe dos códigos de bloco e são caracterizados por uma matriz de verificação de paridade esparsa, isto é, com uma baixa densidade de 1's (caso binário). No que se refere à topologia dos códigos LDPC, é comum fazer-se a distinção entre códigos regulares e irregulares.

5.1 Códigos LDPC regulares e irregulares

Os códigos LDPC regulares foram propostos em 1962 por Robert Gallager em sua tese de doutorado [40]. Um código LDPC regular é definido como sendo um código de bloco linear (n, k, d_v), com $d_v \ll m = n - k$, cuja matriz de verificação de paridade $\boldsymbol{H}$ tem dimensões $(n-k) \times n$, contendo d_v 1's por coluna e $d_c = d_v \times n/m$ 1's por linha. A taxa R de um código LDPC regular é determinada pelos graus d_v e d_c através da seguinte equação:

$$R = 1 - \frac{d_v}{d_c}. \tag{5.1}$$

Gallager demonstrou que fazendo $d_v \geq 3$, os conjuntos de códigos LDPC (n, k, d_v) que podem ser obtidos, na sua grande maioria, possuem uma distância mínima elevada. Mas para isso é necessário seguir algumas regras de construção. Por exemplo, duas colunas da matriz $\boldsymbol{H}$ devem possuir no máximo um 1 na mesma posição. O menor número de colunas de $\boldsymbol{H}$ que somadas resultam no vetor nulo

correspondente à distância mínima do código, como as matrizes $\boldsymbol{H}$ dos códigos LDPC são esparsas e possuem grandes dimensões, códigos LDPC bem projetados possuem distância mínima elevada.

Os códigos LDPC, mesmo com suas excelentes características, foram esquecidos pela comunidade científica, salvo raras exceções, até meados dos anos 90, quando foram redescobertos por Mackay e Neal [6, 41, 42]. Mackay e Neal demonstraram que com probabilidade de erro muito baixa os códigos LDPC conseguem se aproximar do limite estabelecido por Shannon [1]. O que levou ao seu esquecimento por tantos anos foi a elevada complexidade computacional requerida (para a época) tanto para a geração de uma matriz $\boldsymbol{H}$ que garanta uma boa distância mínima do código, como para sua codificação e decodificação. Porém, com os avanços nos campos dos circuitos integrados e do processamento digital de sinais, a implementação dos códigos LDPC se tornou possível.

Mais recentemente, surgiram os códigos LDPC irregulares [43–45], para os quais a matriz $\boldsymbol{H}$ possui uma baixa densidade de 1's mas o número de 1's por coluna e por linha não é constante. Em [7], pode ser encontrada uma demonstração de que os códigos LDPC irregulares são superiores aos regulares para blocos longos.

5.2 Grafos de Tanner

Um importante trabalho sobre códigos LDPC realizado durante o seu esquecimento pós-Gallager, e antes do seu ressurgimento, foi o estudo de Michael Tanner [46]. Tanner considerou que qualquer código linear de bloco, em particular códigos LDPC, poderiam ser representados por grafos bipartidos. Grafos bipartidos são formados por dois conjuntos de nós, onde os nós de um mesmo conjunto não se ligam entre si, mas sim aos nós do outro conjunto.

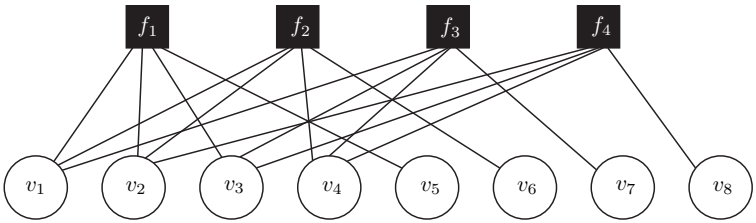


Figura 5.1: Uma representação de grafo bipartido.

O grafo bipartido apresentado na Figura 5.1 adequadamente

utilizado pode representar um codificador/decodificador de um código de bloco. Os ■ representam os nós de função e os ◯ representam os nós de variável. A propriedade de a soma dos bits enviados dos nós de variável aos nós de função ser sempre zero módulo 2 (no caso de LDPC binários) possibilita a geração de palavras-código com grande eficiência e baixa complexidade. A representação grafica está sempre associada à matriz de verificação de paridade do código. As linhas da matriz $\boldsymbol{H}$ correspondem aos nós de função e as colunas da matriz, aos nós de variável. Para cada conexão entre os nós, existe um 1 na correspondente posição na matriz $\boldsymbol{H}$. A matriz $\boldsymbol{H}$ correspondente ao grafo apresentado na Figura 5.1, é dada por:

$$\boldsymbol{H} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{array} \begin{array}{c} \begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\ \left[\begin{array}{cccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}. \tag{5.2}$$

5.2.1 Conceito de *girth*

Um dos conceitos mais importantes relativos aos grafos de Tanner é o de ciclo com comprimento $\mathcal{L}$, definido como sendo um percurso fechado formado por $\mathcal{L}$ caminhos no grafo de Tanner [47, 48]. Tendo como base a Figura 5.2, pode-se observar um ciclo de comprimento 4 formado pelas arestas em negrito. O menor comprimento de todos os ciclos existentes num grafo de Tanner é designado por *girth*.

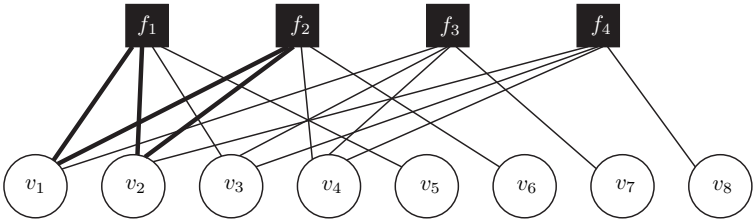


Figura 5.2: Exemplo de um *girth* 4 em um grafo bipartido.

Na prática, ao projetar um código LDPC procura-se evitar a existência de ciclos de comprimento pequeno de forma a melhorar o desempenho do algoritmo de decodificação, conhecido como Algoritmo Soma-produto (SPA). Prova-se que o algoritmo SPA tem desempenho

ótimo quando aplicado a grafos sem ciclos [49–52]. Na presença de ciclos, a eficiência do código diminui, sendo esta inferior para um *girth* baixo. Por outro lado, é provado também que grafos de Tanner sem ciclos não geram bons códigos [53]. É necessário obedecer a algumas regras na construção dos códigos LDPC para que estes possuam boas propriedades (distância mínima e *girth* elevados).

Qualquer ciclo de um grafo de Tanner tem obrigatoriamente um comprimento par, e o seu valor mínimo é 4, correspondendo a uma matriz de verificação de paridade $\boldsymbol{H}$ em que existem duas colunas com dois 1's na mesma posição. Uma das regras de construção da matriz $\boldsymbol{H}$, para evitar a existência de ciclos de dimensão 4, consiste em garantir que quaisquer duas colunas da matriz $\boldsymbol{H}$ possuam no máximo um 1 na mesma posição.

5.2.2 Codificação

Um código LDPC é definido a partir de sua matriz de verificação $\boldsymbol{H}$, a partir da qual é possível se obter uma matriz $\boldsymbol{G}$, geradora de palavras-código. Esse processo de criar palavras-código recebe o nome de codificação, em que a informação a ser transmitida é dividida em blocos de bits, com comprimento definido, e, através da equação

$$\boldsymbol{v} = \boldsymbol{G}\boldsymbol{u}, \tag{5.3}$$

obtém-se as palavras-código $\boldsymbol{v}$ referentes aos blocos de informação $\boldsymbol{u}$. O processo descrito é utilizado em códigos de bloco lineares em geral, mas no caso dos códigos LDPC, que são códigos de bloco lineares com matrizes de verificação de paridade enormes, esse processo envolve um número muito grande de operações e o uso de grande quantidade de memória, o que torna o processo custoso, inviabilizando sua utilização. Em particular, enquanto a complexidade de decodificação via grafos de um código LDPC de comprimento n é linear em n, a complexidade de codificação é da ordem de n^2.

Existem muitos métodos alternativos para se realizar a codificação sem o uso da matriz geradora, mas a grande maioria deles envolve também um grande número de operações. O método escolhido nesta tese é uma composição de duas técnicas, e será descrito no Capítulo 6.

5.2.3 Decodificação

Devido ao seu excelente desempenho, os códigos LDPC são atualmente reconhecidos como a classe de códigos que melhor se

aproxima do limite de Shannon [1]. O sucesso dos códigos LDPC deve-se em grande parte ao algoritmo de decodificação iterativo, introduzido por Gallager [40] e depois redescoberto por MacKay e Neal [6, 42], que evidenciaram a sua importância.

O algoritmo apresentado por Gallager [40] tem por base a representação gráfica dos códigos LDPC, e é conhecido como Algoritmo Soma-Produto (SPA). No entanto, o seu campo de aplicação vai muito além da decodificação de códigos LDPC, abrangendo áreas do processamento de sinais, das comunicações digitais e da inteligência artificial, onde é conhecido por *belief propagation* devido a sua aplicação em redes Bayesianas. A designação do algoritmo SPA por *belief propagation* é comum no contexto da decodificação de códigos LDPC. Ambos os termos são encontrados em diversos trabalhos e têm o mesmo significado. Quanto ao processo de decodificação de um código, existem duas abordagens possíveis:

(*i*) A decodificação abrupta (*hard decoding*) considera que o conjunto de símbolos na entrada do decodificador é finito, ou seja, no caso de um código binário na entrada do decodificador teremos apenas bits, as decisões se cada bit recebido foi 0 ou 1 sendo tomadas antes da decodificação. Depois é escolhida a palavra-código mais próxima do vetor recebido;

(*ii*) A decodificação suave (*soft decoding*) considera a distribuição probabilística dos símbolos recebidos, sem a decisão prévia bit a bit, e ambas tecnicas tomam sua decisão baseada na palavra-código inteira.

Os algoritmos do tipo *hard decoding* são na sua grande maioria menos complexos. Também é verdade que, levando-se em conta a distribuição probabilística dos dados recebidos pelo decodificador, os algoritmos *soft decoding* apresentam um melhor desempenho em termos de uma menor taxa de erros (BER). Nesta tese será utilizado o Algoritmo SPA, que é um algoritmo iterativo e do tipo *soft decoding*. Sua descrição é feita no Apêndice A.

5.3 Projeto de códigos LDPC

O projeto de um código LDPC consiste na construção da matriz de verificação de paridade $\boldsymbol{H}$ que atinja os objetivos pretendidos, levando-se em consideração se o código LDPC é regular ou irregular. Existem diversos métodos para se obter essa matriz de verificação de paridade, e, impondo-se algumas restrições nessa matriz, como o

número de 1's por coluna, a taxa do código, etc., podem ser criadas várias famílias (ou *ensembles*) de códigos LDPC.

Dado um comprimento n para o código LDPC regular, um *ensemble* de códigos é determinado pelos seus graus de nós de variável (d_v) e de nós de função ou *check* (d_c). Esses graus estarão representados na matriz de verificação de paridade respectivamente como número de elementos diferentes de zero por coluna e linha. Portanto, um grupo de códigos LDPC regulares pode ser representado por uma notação (n, d_v, d_c). A análise de desempenho de um *ensemble* é caracterizada pelo desempenho médio dos códigos pertencentes a este conjunto.

Embora o pressuposto de regularidade simplifique muito a análise do desempenho dos códigos, ela impõe restrições desnecessárias sobre a estrutura da matriz de verificação de paridade. Os códigos LDPC irregulares permitem um número diferente de elementos não-nulos em cada linha e coluna. A construção via grafos bipartidos não está conectada à regularidade do código e, sendo assim, aplica-se aos códigos LDPC irregulares também. No caso irregular, o número de arestas que conectam os nós pode variar de nó para nó, independentemente do tipo de nó. Em outras palavras, os nós de variável (ou de função) podem ter diferentes graus.

Para descrever os *ensembles* dos códigos LDPC irregulares, introduz-se o conceito de distribuição de graus. A distribuição de graus para um código LDPC é descrita pelo seguinte par de polinômios:

$$\lambda(x) = \sum_{i=1}^{v_{(\max)}} \lambda_i x^{i-1}, \tag{5.4}$$

$$\rho(x) = \sum_{j=1}^{c_{(\max)}} \rho_j x^{j-1}, \tag{5.5}$$

em que o coeficiente λ_i é a fração das arestas ligadas a nós de variável com grau i e ρ_j é a fração das arestas ligadas a nós de função com grau j. O polinômio $\rho(x)$ é referente à distribuição de graus para os nós de função e $\lambda(x)$ é referente à distribuição de graus para os nós de variável. Esses coeficientes têm que satisfazer as seguintes restrições:

$$\begin{aligned} &0 \leq \rho_j \leq 1 \ \text{ e } \ j \geq 1, \\ &0 \leq \lambda_i \leq 1 \ \text{ e } \ i \geq 1, \\ &\textstyle\sum_{j=1}^{c_{(\max)}} \rho_j = 1, \\ &\textstyle\sum_{i=1}^{v_{(\max)}} \lambda_i = 1, \end{aligned} \tag{5.6}$$

em que a terceira e quarta restrições são consequência de a soma de todas as frações de distribuição dos graus para cada tipo de nó ter que ser igual a um.

Quando o grafo bipartido possui l arestas, a correspondente matriz de verificação de paridade $\boldsymbol{H}$ possui l elementos não-nulos. O número de nós de variável com grau i é dado por:

$$v_i = \frac{l\lambda_i}{i},$$

e o número de nós de função de grau j é dado por:

$$c_j = \frac{l\rho_j}{j}.$$

Dessa forma, o numero n de nós de variável é obtido por:

$$n = \sum_i v_i = l \sum_i \frac{\lambda_i}{i} = l \int_0^1 \lambda(x)dx.$$

O número m de nós de função é obtido por:

$$m = \sum_j c_j = l \sum_j \frac{\rho_j}{j} = l \int_0^1 \rho(x)dx.$$

A taxa do código $R \leq k/n^*$ é calculada como

$$\begin{aligned} R &\leq \frac{k}{n} = \frac{n-m}{n} = 1 - \frac{m}{n} \\ &\leq 1 - \frac{l\sum_j \frac{\rho_j}{j}}{l\sum_i \frac{\lambda_i}{i}} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}. \end{aligned}$$

Note que, dada a taxa R para o código LDPC, os coeficientes de distribuição de graus têm que satisfazer a seguinte restrição:

$$\sum_{j=1}^{c_{(\max)}} \frac{\rho_j}{j} = (1-R) \sum_{i=1}^{v_{(\max)}} \frac{\lambda_i}{i}. \tag{5.7}$$

Códigos LDPC irregulares estão entre os mais poderosos códigos binários conhecidos hoje. Em [45], é mostrado como projetar as distribuições de graus para códigos LDPC irregulares. E em [54], é mostrado que, cuidadosamente projetados, códigos LDPC irregulares chegam a 0,0045 dB do limite de Shannon para um canal AWGN (com ruído aditivo Gaussiano branco).

*A taxa é menor que k/n quando a matriz $\boldsymbol{H}$ não possui posto completo.

5.3.1 *Density evolution*

O desempenho de códigos LDPC, tanto os regulares como os irregulares, pode ser determinado utilizando-se uma abordagem chamada "*density evolution*". *Density evolution* tem sua origem nas fórmulas de evolução da probabilidade de erro de Gallager [40], e as extensões modernas para canais com saída suave são essencialmente uma generalização sofisticada dos métodos antigos.

A principal indicação que o *density evolution* pode fornecer para os códigos LDPC é a seguinte: Um código LDPC com um dado par de distribuições de graus (λ, ρ), operando em um canal $\mathcal{C}(\sigma)$, tem um limite associado σ^*. O limite é análogo à capacidade de Shannon, no sentido de que a probabilidade de erro para uma transmissão através do canal $\mathcal{C}(\sigma)$, utilizando um código com distribuições de graus (λ, ρ), escolhido aleatoriamente, pode ser feita arbitrariamente pequena à medida que o tamanho de bloco para o código cresça, se e somente se $\sigma < \sigma^*$. Um par de distribuições de graus é dito ser melhor que outro se o seu limite é mais próximo da capacidade do canal.

A probabilidade de erro em um canal específico para um código com distribuições de graus (λ, ρ) escolhido aleatoriamente pode ser arbitrariamente pequena se o parâmetro do canal é melhor do que o limiar (σ^*). Resultados da otimização de graus para diversos canais são apresentados em [55]. Diferentes *ensembles* de códigos LDPC regulares e irregulares são estudados e otimizados através de *density evolution* em [45, 56].

No projeto para se obter bons códigos, pode-se determinar a distribuição de graus que fornece o melhor limiar e, a partir desta, pode-se selecionar o maior comprimento de bloco permitido pela aplicação. Tipicamente, verifica-se que quase todos os códigos pertencentes ao *ensemble* possuem um desempenho semelhante. Para finalizar o projeto do código, realiza-se uma amostragem aleatória de alguns dos códigos do *ensemble*, escolhendo-se os melhores dentre eles.

5.3.2 EXIT *charts*

Os EXIT (do inglês, "*EXtrinsic Information Transfer*") *charts* são gráficos que quantificam a transferência de informação extrínseca entre decodificadores na decodificação iterativa. Inventados por ten Brink em 1999 [57], os EXIT *charts* têm-se revelado muito úteis na avaliação do desempenho e na ajuda ao projeto de códigos LDPC e turbo. A ideia original de ten Brink foi a de utilizar a informação mútua média para monitorar o processo de decodificação iterativa de códigos turbo [4].

Os EXIT *charts* foram posteriormente aplicados aos códigos LDPC.

A idéia central do EXIT *chart* é avaliar o incremento de informação a respeito de X (sinal transmitido) quando um certo bloco de decodificação, que tem na sua entrada uma informação *a priori*, realiza um processamento e fornece na sua saída a informação extrínseca. Em termos mais concretos, calcula-se, por um lado, a informação mútua entre X e a informação *a priori* (entrada do bloco de decodificação) e, por outro lado, calcula-se a informação mútua entre X e a informação extrínseca (saída do bloco de decodificação). A relação entrada-saída dá origem à chamada "função de transferência" de informação. Deve-se obter essa função para ambos os blocos de decodificação de um decodificador iterativo. Como, numa decodificação iterativa, a informação extrínseca de um bloco de decodificação é usada como informação *a priori* para o outro bloco de decodificação, faz-se uma plotagem, num mesmo gráfico, da função de transferência de um bloco e da função de transferência inversa do outro. Como no início da decodificação iterativa a informação *a priori* é zero (não se tem informação *a priori*, ou seja, X pode ser +1 ou -1 com igual probabilidade), inicia-se uma trajetória no ponto de informação *a priori* zero para o primeiro bloco e, saltando de uma curva para outra, a trajetória (EXIT *chart*) segue em direção ao ponto de convergência do algoritmo de decodificação, indicando a sua evolução em termos de ganho de informação a respeito do sinal transmitido X.

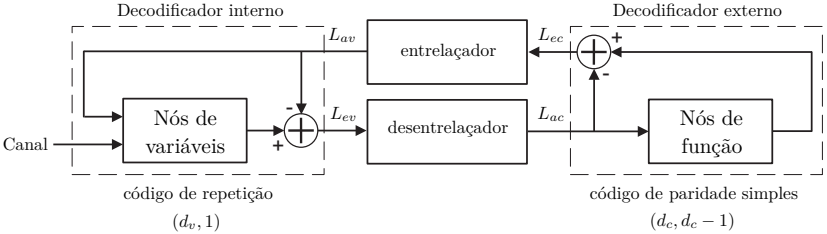


Figura 5.3: Diagrama de blocos de um decodificador iterativo LDPC.

A decodificação iterativa de códigos LDPC pode ser vista como a decodificação de dois códigos concatenados em série, como na Figura 5.3. O conjunto de nós de variável do grafo de Tanner representa o decodificador interno, e o conjunto de nós de função representa o decodificador externo. As arestas que ligam os nós entre si representam as operações de entrelaçamento e desentrelaçamento realizadas entre os códigos concatenados [58]. Em códigos LDPC regulares (d_v, d_c),

a decodificação nos nós de variável equivale a calcular LLR (*Log Likelihood Ratio*) extrínsecas num código de repetição $(d_v, 1)$, e a decodificação nos nós de paridade equivale a calcular LLR extrínsecas num código de paridade simples $(d_c, d_c - 1)$.

As LLR condicionadas *a priori* $L(y|x)$ e *a posteriori* $L(x|y)$ são obtidas através do procedimento a seguir. Pelo teorema de Bayes, tem-se:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}.$$

Como x pode ser igual a ± 1, a razão $\dfrac{p(x=1|y)}{p(x=-1|y)}$ é dada por:

$$\frac{p(x=1|y)}{p(x=-1|y)} = \frac{p(y|x=1)p(x=1)}{p(y|x=-1)p(x=-1)}.$$

Aplicando-se o logaritmo a esta razão, obtém-se a LLR condicionada *a posteriori* $L(x|y)$:

$$L(x|y) = \ln \frac{p(x=1|y)}{p(x=-1|y)} = \underbrace{\ln \frac{p(y|x=1)}{p(y|x=-1)}}_{L(y|x)} + \underbrace{\ln \frac{p(x=1)}{p(x=-1)}}_{L_a(x)}.$$

Assim, obtém-se a relação entre as LLR dada por:

$$L(x|y) = L(y|x) + L_a(x). \tag{5.8}$$

Para o canal AWGN, o ruído tem variância igual a:

$$\sigma^2 = \frac{1}{2R\dfrac{E_b}{N_o}}, \tag{5.9}$$

em que R é a taxa do código, E_b a energia do bit e $N_o/2$ a densidade espectral de potência do ruído. As densidades de probabilidades condicionadas $p(y|x = \pm 1)$ são distribuições Gaussianas, e $L(y|x)$ é dado por:

$$L(y|x) = \ln \left[\frac{p(y|x=1)}{p(y|x=-1)} \right] = \ln \left[\frac{\dfrac{1}{\sqrt{2\pi\sigma^2}} e^{-\dfrac{(y-\mu)^2}{2\sigma^2}}}{\dfrac{1}{\sqrt{2\pi\sigma^2}} e^{-\dfrac{(y+\mu)^2}{2\sigma^2}}} \right] = \frac{2\mu y}{\sigma^2}. \tag{5.10}$$

A informação mútua entre uma variável aleatória binária X em que os valores ± 1 são igualmente prováveis e uma variável aleatória Y é dada por [57]:

$$I(X;Y) = \frac{1}{2}\sum_{x=1,-1}\int_{-\infty}^{\infty} p(y|X{=}x)\log_2\frac{2p(y|X{=}x)}{p(y|X{=}-1)+p(y|X{=}1)}dy. \quad (5.11)$$

Assumindo-se que Y seja simétrica, ou seja, $p(y|X{=}{-}1) = p(-y|X{=}1)$, pode-se simplificar a equação (5.11) para:

$$I(X;Y) = \int_{-\infty}^{\infty} p(y|X=1)\log_2\frac{2p(y|X=1)}{p(-y|X=1)+p(y|X=1)}dy. \quad (5.12)$$

Para a análise de EXIT *charts*, na verdade, tem-se o interesse na informação mútua $I(X;L)$, em que, para o canal Gaussiano, L se relaciona estatisticamente com X pela equação (5.10). Substituindo-se y por $x+n$ em (5.10), em que x pode ser ± 1 e n é uma variável aleatória Gaussiana com média zero e variância σ^2, verifica-se que, condicionada ao evento $X = x$, L é uma variável aleatória Gaussiana dita *consistente*, o que simplifica o cálculo. Uma variável aleatória Y é condicionalmente Gaussiana consistente dado o evento $X = x$ se $p(y|X = x) = p(-y|X = x)e^y$, ou, equivalentemente, se a sua média for igual à metade da sua variância. Sob essa condição, tem-se que a informação mútua entre X e uma LLR L que tem a forma dada por (5.10) é dada por

$$I(X;L) = 1 - \int_{-\infty}^{\infty} p(y|X=x)\log_2(1+e^{-y})dy. \quad (5.13)$$

Chamando $I(X;L)$ de $J(\sigma)$, e substituindo-se $p(y|X = x)$ pela correspondente função densidade de probabilidades Gaussiana consistente no integrando de (5.13), tem-se

$$J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{e^{\frac{-(y-\sigma^2/2)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\log_2(1+e^{-y})dy. \quad (5.14)$$

A Figura 5.4 mostra os esquemas de decodificação apropriados para se determinarem as funções EXIT de códigos LDPC regulares. Na figura, o bloco "Repetição" representa o código de repetição e o bloco "Paridade" o código de paridade simples. Note que no esquema de decodificação correspondente aos nós de função não se considera nenhuma informação do canal, pois os nós de função no grafo de Tanner

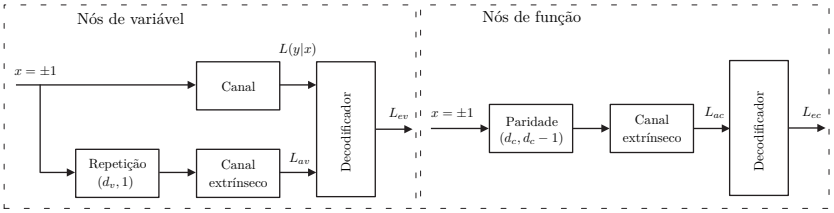


Figura 5.4: Diagrama de blocos para o cálculo das funções EXIT de códigos LDPC regulares.

não representam símbolos transmitidos pelo canal, ao contrário dos nós de variável. Sendo assim, o decodificador exterior só tem uma entrada: L_{ac}. Com base no algoritmo Soma-Produto (SPA), a mensagem de saída do i-ésimo nó de variável é dada pela equação

$$L_{ev}^{(i)} = L(y|x)^{(i)} + \sum_{j \neq i} L_{av}^{(j)}, \tag{5.15}$$

em que os $L_{av}^{(j)}$, para $j \neq i$, são as d_v mensagens enviadas pelos nós de função conectados ao i-ésimo nó de variável e $L(y|x)^{(i)}$ é a mensagem associada ao canal de comunicação, mais conhecida com a "LLR do canal".

Com base na Figura 5.4, para se calcular a função EXIT dos nós de variável, consideram-se as mensagens vindas do canal de comunicação $L(y|x)$ e as do canal extrínseco L_{av}. As mensagens $L(y|x)$ são estatisticamente ligadas ao canal de comunicação utilizado. As mensagens L_{av} podem ser consideradas variáveis aleatórias com distribuição Gaussiana. Esta é uma aproximação válida devido ao mecanismo de decodificação LDPC, que realiza uma soma de várias mensagens de valores L (ver equação (5.15)). Por força do teorema do limite central, o resultado desta operação segue uma distribuição Gaussiana. Sendo assim, como mostrado em [57], pode-se escrever a função EXIT para os nós de variável como:

$$\begin{aligned} I(X; L_{ev}) &= I_{E_v}(I_{A_v}, d_v, E_b/N_o) \\ &= J\left(\sqrt{(d_v - 1)[J^{-1}(I_{A_v})]^2 + J^{-1}(I_{E_v,DET})^2}\right), \end{aligned} \tag{5.16}$$

em que $I_{E_v,DET}$ representa a informação mútua entre a LLR do canal de comunicação (mensagens iniciais $L(y|x)$) e os bits enviados. Este valor pode ser obtido analiticamente, se o modelo estatístico do canal

de comunicação permitir, ou através de métodos numéricos, como via simulação de Monte Carlo. Para o canal de comunicação AWGN com modulação BPSK o valor $I_{E_v,DET}$ pode ser obtido analiticamente e é dado por:

$$I_{E_v,DET} = J(\sigma_c), \tag{5.17}$$

em que $\sigma_c = \dfrac{2}{\sigma}$.

Segundo [59] a mensagem associada à saída do j-ésimo nó de função é dada por:

$$L_{ec}^{(j)} = \ln \left(\frac{1 - \prod\limits_{k \neq j} \dfrac{1 - e^{L_{ac}^{(k)}}}{1 + e^{L_{ac}^{(k)}}}}{1 + \prod\limits_{k \neq j} \dfrac{1 - e^{L_{ac}^{(k)}}}{1 + e^{L_{ac}^{(k)}}}} \right). \tag{5.18}$$

As mensagens de saída dos nós de função, segundo [60, 61], também podem ser aproximadas por variáveis aleatórias gaussianas.

Alem disso, é sabido que os códigos de paridade são os códigos duais dos códigos de repetição, e existe uma propriedade de dualidade que relaciona as funções EXIT de códigos duais em canais binarios com apagamento (BEC) [58]. No caso dos códigos LDPC, essa propriedade de dualidade facilita o cálculo das funções EXIT. Embora a propriedade de dualidade seja exata apenas para o canal BEC, ela é muito bem aproximada para outros canais de comunicação e, portanto, pode ser utilizada em geral. Sendo assim, de [58], tem-se que:

$$I_{E,PAR}(I_A, d_c) \approx 1 - I_{E,REP}(1 - I_A, d_c). \tag{5.19}$$

Assim, explorando-se a dualidade entre códigos de repetição e de paridade simples, bem como explorando-se a aproximação Gaussiana supramencionada, a função de transferência relativa aos nós de função é dada por [62]:

$$I(X; L_{ec}) = I_{E_c}(I_{A_c}, d_c) \approx 1 - J\left(\sqrt{(d_c - 1)} J^{-1}(1 - I_{A_c})\right), \tag{5.20}$$

Na Figura 5.5 é apresentado o EXIT *chart* para um canal AWGN de um código LDPC regular ($d_v = 2$ e $d_c = 3$) com taxa $1/3$ e $E_b/N_o =$ 4dB. O ponto inicial, quando $I_{A_v} = 0$ corresponde a $I_{E_v} = J(\sigma_c)$. A trajetória de decodificação está representada pela linha tracejada, e é indicado o ponto do fim da primeira iteração. Através desse gráfico é possível afirmar que para $E_b/N_o = 4$dB o decodificador converge com poucas iterações.

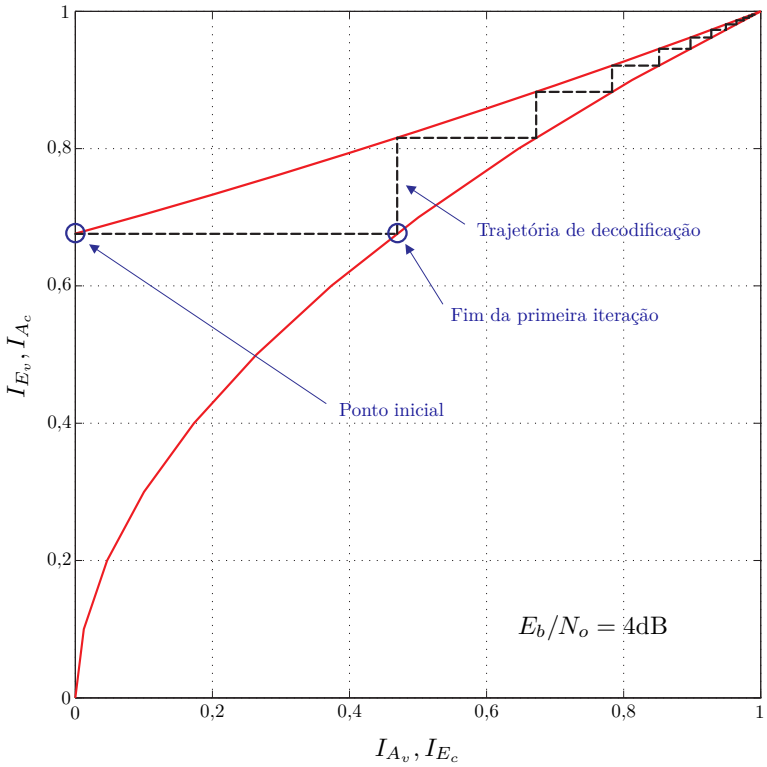


Figura 5.5: EXIT *charts* de um código LDPC regular com taxa 1/3 ($d_v = 2$ e $d_c = 3$).

Sempre que as curvas não se interceptarem, situação na qual se diz haver um "túnel", exitirá uma trajetória de decodificação. Então, pode-se afirmar que o decodificador converge. Por outro lado, quando as duas funções de transferência (curvas vermelhas) encontram-se muito afastadas, significa que o valor de E_b/N_o está muito longe da capacidade de canal. É necessário então realizar uma otimização do par (λ, ρ) a fim de aproximar as duas curvas, aproximando-se, assim, da capacidade de canal. Essa é uma das mais importantes aplicações do EXIT *chart*.

As funções EXIT para códigos LDPC irregulares podem ser obtidas a partir das expressões para os códigos regulares. Como se sabe, num código LDPC irregular, o número de ramos do grafo de Tanner ligados a cada nó não é o mesmo. Nesses códigos, a função EXIT de um grupo de nós (de variável ou de função) é a média das funções EXIT

associadas a cada grau ponderada pelas proporções dos ramos ligados aos nós desse grau. Assim, considere um código LDPC irregular em que num dos grupos de nós o grau máximo é D. Se b_k representar a fração dos ramos que incidem nos nós de grau k, a informação mútua média I_E é igual à média ponderada das informações mútuas médias I_{E_k} associadas aos nós de grau k [58, 62]:

$$I_E = \sum_{k=1}^{D} b_k I_{E_k}, \tag{5.21}$$

em que $\sum_{k=1}^{D} b_k = 1$. Se os nós de variável e de função tiverem graus máximos D_v e D_c, respectivamente, e os coeficientes dos polinômios de distribuição de graus $\lambda(x)$ e $\rho(x)$ forem λ_i e ρ_j, respectivamente, as funções EXIT para códigos LDPC irregulares em canais AWGN são expressas por:

$$\begin{aligned} I(X; L_{ev}) &= I_{E_v}(I_{A_v}, \lambda(x), E_b/N_o) \\ &= \sum_{i=1}^{D_v} \lambda_i J\left(\sqrt{(i-1)[J^{-1}(I_{A_v})]^2 + \sigma_c^2}\right), \end{aligned} \tag{5.22}$$

$$\begin{aligned} I(X; L_{ec}) &= I_{E_c}(I_{A_c}, \rho(x)) \\ &\approx \sum_{j=1}^{D_c} \rho_j \left[1 - J\left(\sqrt{(j-1)} J^{-1}(1 - I_{A_c})\right)\right]. \end{aligned} \tag{5.23}$$

Na Figura 5.6 é apresentado um diagrama de blocos simplificado do processo de decodificação LDPC. Este diagrama foi anteriormente apresentado com mais detalhes na Figura 5.3.

Considere os dois pares de curvas EXIT para os blocos A e B apresentadas na Figura 5.7, denotadas por $(I_{1,A}(\cdot), I_{1,B}^{-1}(\cdot))$ e $(I_{2,A}(\cdot), I_{2,B}^{-1}(\cdot))$. Para estas curvas, tem-se que:

$$\begin{aligned} I_{1,A}(I) &\geq I_{2,A}(I) \ \ \forall I \in [0,1]; \\ I_{1,B}^{-1}(I) &\leq I_{2,B}^{-1}(I) \ \ \forall I \in [0,1]. \end{aligned} \tag{5.24}$$

Como $I_{1,A}$ é maior que $I_{2,A}$ e $I_{1,B}^{-1}$ é menor que $I_{2,B}^{-1}$, então a convergência no processo de decodificação para o sistema referente às curvas EXIT $(I_{1,A}(\cdot), I_{1,B}^{-1}(\cdot))$ será mais rápida (menor número de iterações) do que o processo de convergência para o sistema referente às

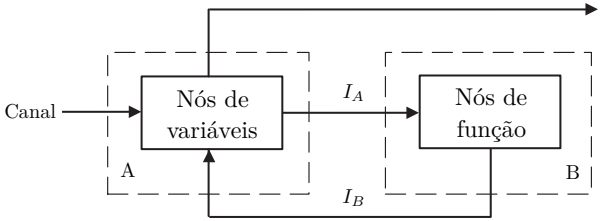


Figura 5.6: Diagrama de blocos de um decodificador iterativo LDPC para analise de EXIT *charts*.

curvas EXIT $(I_{2,A}(\cdot), I_{2,B}^{-1}(\cdot))$. Existe uma forma simples de visualizar a velocidade de convergência através dos pares de curvas EXIT. Observe que as curvas formam "túneis", e quanto mais abertos forem esses "túneis" mais rápida será a convergência do decodificador iterativo.

O processo de decodificação não convergirá se existir um valor I^*, $0 < I^* < 1$, tal que $I_A(I^*) < I_B^{-1}(I^*)$, ou seja, se o "túnel" para as curvas EXIT $(I_A(\cdot), I_B^{-1}(\cdot))$ estiver fechado. Para a otimização dos códigos LDPC é definida uma função de abertura do "túnel" como:

$$f(\lambda, \rho) = \min_{I \in [0,1]} \{I_A(I) - I_B^{-1}(I)\}, \tag{5.25}$$

em que estão indicados os polinômios de distribuições de graus para evidenciar a sua dependência.

5.3.3 Algoritmo de otimização de códigos LDPC usando EXIT *charts*

O algoritmo que será descrito abaixo foi apresentado em [63], e pode ser visto como um caso particular do chamado algoritmo "*differential evolution*" [64].

O algoritmo é inicializado com um determinado par de distribuições de graus: uma distribuição de graus de nós de variável λ e uma distribuição de graus de nós de função ρ, válidas para uma dada taxa de código, de acordo com a equação (5.7). Se o túnel não estiver fechado, ou seja, se $f(\lambda, \rho) = 0$, diminui-se a SNR até que o túnel se feche e $f(\lambda, \rho) < 0$. Em seguida, realiza-se uma adaptação do par (λ,ρ), buscando-se melhores resultados. A adaptação corresponde a uma caminhada "inteligente" no espaço paramétrico definido pelas distribuições de graus. Em particular, nesta tese, sugere-se o chamado "passo fundamental", sobre o qual uma descrição sucinta é dada a

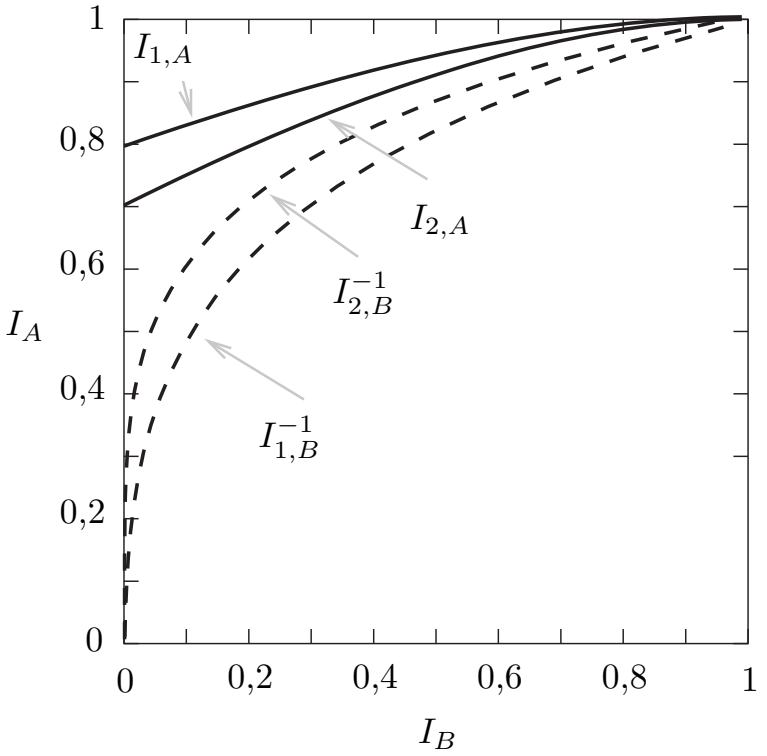


Figura 5.7: Dois pares de EXIT *charts* rotulados como 1 e 2. O par 1 possui melhores propriedades de convergência.

seguir. Dar um passo fundamental significa que uma determinada fração da distribuição dos graus é incrementada minimamente e as demais são ajustadas de modo a manter a validade da nova distribuição, pois, há restrições topológicas que precisam ser respeitadas. Em particular, o incremento de uma das frações força uma mudança em outras. Mais detalhes sobre o "passo fundamental" são fornecidos na Seção E.1 do Apêndice E.

A partir do novo par de distribuições de graus, calcula-se $f(\lambda, \rho)$: se esse valor for maior do que o anterior substitui-se o par de distribuições anterior pelo novo par. Se o túnel estiver aberto, diminui-se novamente a SNR, e as etapas anteriores são repetidas. O algoritmo pára quando uma exigência específica for cumprida, como por exemplo: o *ensemble* de códigos obtido corresponder a um EXIT *chart* com túnel aberto e

para a SNR desejada, ou um número máximo definido de iterações do algoritmo for atingido. Na Seção E.2 do Apêndice E é apresentado o fluxograma referente a esse algoritmo.

Capítulo 6

Códigos LDPC para códigos de barra 2D coloridos

Neste capítulo serão descritos os processos de codificação, decodificação e o método utilizado na otimização de códigos LDPC especialmente projetados para aplicações de códigos de barra 2D coloridos. Finalmente, serão apresentados os resultados obtidos.

6.1 Forma de obtenção das matrizes de verificação de paridade

A forma de obtenção da matriz de verificação de paridade para os códigos LDPC utilizados nessa tese é baseada na combinação de uma abordagem algébrica e de uma abordagem pseudo-aleatória de construção de matrizes esparsas. As matrizes $\boldsymbol{H}$ são construídas em duas etapas através da concatenação de duas matrizes:

$$\boldsymbol{H} = [\boldsymbol{H}_1 \;\; \boldsymbol{H}_2], \tag{6.1}$$

em que $\boldsymbol{H}_1$ corresponde a uma matriz esparsa com dimensões $m \times k$ e $\boldsymbol{H}_2$ a uma matriz quadrada $m \times m$ também esparsa e inversível.

A matriz $\boldsymbol{H}_2$ é construída utilizando-se as técnicas empregadas para códigos *Quasi-Cyclic* LDPC (QC-LDPC) [65], que são códigos LDPC estruturados e de fácil implementação. Além disso, possuem um desempenho praticamente equivalente ao dos códigos LDPC pseudo-aleatórios. A memória necessária para armazenar as matrizes para seus codificadores e decodificadores pode ser reduzida por um

fator L quando são utilizadas matrizes circulantes de dimensões $L \times L$. Os códigos QC-LDPC também têm vantagens sobre códigos LDPC pseudo-aleatórios por possuírem uma complexidade de codificação linear no comprimento da palavra-código (n).

Os códigos QC-LDPC são caracterizados por possuírem uma matriz de verificação de paridade formada pela composição de diversas matrizes quadradas de dimensões $L \times L$. Estas podem ser matrizes de permutação circulantes ou matrizes nulas. Para explicar o procedimento será escolhida como matriz circulante a matriz identidade $\boldsymbol{P}(0)$; a partir desta, define-se a matriz de permutação circulante $\boldsymbol{P}(1)$ como:

$$\boldsymbol{P}(1) = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}. \tag{6.2}$$

De modo mais geral, define-se a matriz $\boldsymbol{P}(i)$ como uma matriz identidade ($L \times L$) com um deslocamento circular de i colunas para a esquerda, em que $0 \leq i < L$. Normalmente, para simplificar a notação, $\boldsymbol{P}(\infty)$ é definido como a matriz nula de dimensões $L \times L$. Desta forma, uma matriz de verificação de paridade para um código QC-LDPC corresponde a uma matriz $cL \times rL$ dada por:

$$\boldsymbol{H}_{QC} = \begin{bmatrix} \boldsymbol{P}(a_{11}) & \boldsymbol{P}(a_{12}) & \dots & \boldsymbol{P}(a_{1r}) \\ \boldsymbol{P}(a_{21}) & \boldsymbol{P}(a_{22}) & \dots & \boldsymbol{P}(a_{2r}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{P}(a_{c1}) & \boldsymbol{P}(a_{c2}) & \dots & \boldsymbol{P}(a_{cr}) \end{bmatrix}, \tag{6.3}$$

em que $a_{ij} \in \{\infty, 0, 1, 2, \dots, L-1\}$.

Para obter a matriz $\boldsymbol{H}_2$ em (6.1), utilizou-se o método proposto em [66], que permite a obtenção de uma matriz quadrada inversível em que o correspondente grafo é livre de ciclos de comprimento 4. É importante ressaltar que, em [66], a matriz $\boldsymbol{H}_{QC}$ é adotada como sendo a matriz de verificação de paridade, com $c < r$. Nesta tese, essa estrutura de matriz é usada apenas para a sub-matriz $\boldsymbol{H}_2$ em (6.1), na qual faz-se $r = c$. De [66], a matriz $\boldsymbol{H}_2$ foi escolhida nesta tese como:

$$\boldsymbol{H}_2 = \begin{bmatrix} \boldsymbol{P}(0) & \boldsymbol{P}(\infty) & \boldsymbol{P}(\infty) & \boldsymbol{P}(\infty) & \boldsymbol{P}(0) \\ \boldsymbol{P}(2) & \boldsymbol{P}(0) & \boldsymbol{P}(\infty) & \boldsymbol{P}(0) & \boldsymbol{P}(\infty) \\ \boldsymbol{P}(\infty) & \boldsymbol{P}(12) & \boldsymbol{P}(0) & \boldsymbol{P}(\infty) & \boldsymbol{P}(16) \\ \boldsymbol{P}(\infty) & \boldsymbol{P}(24) & \boldsymbol{P}(0) & \boldsymbol{P}(12) & \boldsymbol{P}(\infty) \\ \boldsymbol{P}(8) & \boldsymbol{P}(\infty) & \boldsymbol{P}(\infty) & \boldsymbol{P}(18) & \boldsymbol{P}(52) \end{bmatrix}, \tag{6.4}$$

em que todas as sub-matrizes que formam $\boldsymbol{H}_2$ possuem dimensões $L \times L$. A partir da variação do parâmetro L, diferentes taxas do código LDPC poderão ser obtidas.

A matriz $\boldsymbol{H}_1$ é construída com uma abordagem pseudo-aleatória. Mas, com o prévio conhecimento da matriz $\boldsymbol{H}_2$, restrições são impostas a $\boldsymbol{H}_1$ de forma que a matriz $\boldsymbol{H}$ resultante não possua ciclos de comprimento 4 e siga a distribuição de graus desejada para o código LDPC conforme projetado.

Em [67] é apresentado um método muito semelhante, entretanto, naquele trabalho, a matriz $\boldsymbol{H}_2$ não possui inversa esparsa e, no processo de codificação, é utilizado o método iterativo de Jacobi de inversão de matrizes. O processo de codificação para o método proposto nesta tese será descrito na próxima seção e é mais simples do que aquele proposto em [67], uma vez que a matriz inversa utilizada é também esparsa.

6.1.1 Método de codificação LDPC utilizado

Seja $\boldsymbol{u}$ um vetor de comprimento k que corresponde ao vetor de informação. A partir do vetor $\boldsymbol{u}$, obtém-se o vetor $\boldsymbol{v}$ com comprimento n, que corresponde à palavra-código. A forma tradicional de se obter $\boldsymbol{v}$ a partir de $\boldsymbol{u}$ é através da matriz $\boldsymbol{G}$ (matriz geradora):

$$\boldsymbol{v} = \boldsymbol{G}\boldsymbol{u}.$$

Entretanto, em geral, obter $\boldsymbol{G}$ a partir de $\boldsymbol{H}$ é um processo bastante complexo. Além disso, a matriz obtida $\boldsymbol{G}$ normalmente não corresponde a uma matriz esparsa. Foi para contornar esse problema que a matriz $\boldsymbol{H}$ foi escolhida como uma composição de duas matrizes, como descrito na equação (6.1), em que $\boldsymbol{H}_2$ é inversível. Considere a matriz $m \times m$ $\boldsymbol{G}_2$ dada por: $\boldsymbol{G}_2 = \boldsymbol{H}_2^{-1}$. O código LDPC em consideração é um código sistemático, ou seja, $\boldsymbol{v} = [u_1\ u_2\ \ldots\ u_k\ p_1\ p_2\ \ldots\ p_m]^T$. As palavras-código podem ser obtidas a partir de $\boldsymbol{H}_1$, $\boldsymbol{G}_2$ e $\boldsymbol{u}$ como segue.

Considere o processo intermediário no qual um vetor $\tilde{\boldsymbol{p}}$ é obtido a partir de $\boldsymbol{H}_1$ e $\boldsymbol{u}$ da seguinte forma:

$$\tilde{\boldsymbol{p}} = \boldsymbol{H}_1\boldsymbol{u}. \tag{6.5}$$

Como $\boldsymbol{H}_1$ é esparsa, esse processo é de baixa complexidade. A partir de $\tilde{\boldsymbol{p}}$ e $\boldsymbol{G}_2$, os bits de paridade são obtidos:

$$\boldsymbol{p} = \boldsymbol{G}_2\tilde{\boldsymbol{p}}. \tag{6.6}$$

Finalmente, a palavra-código é obtida pela composição do vetor de informação $\boldsymbol{u}$ e do vetor com as paridades $\boldsymbol{p}$:

$$\boldsymbol{v} = [\boldsymbol{u}^T \;\; \boldsymbol{p}^T]^T. \tag{6.7}$$

Note que, como esperado, para uma palavra-código $\boldsymbol{v}$, tem-se que: $\boldsymbol{H}\boldsymbol{v} = \boldsymbol{0}$.

É importante notar que, como $\boldsymbol{G}_2$ foi feita uma matriz esparsa, o processo em (6.6) será de baixa complexidade. Adicionalmente, como nesta tese os códigos LDPC têm taxas muito altas, e como $\boldsymbol{G}_2$ tem dimensão $m \times m$, em que m é o número de bits de paridade, a complexidade de (6.6) para o caso em apreço é ainda mais baixa.

Para se ter um exemplo, em um dos códigos LDPC obtidos nesta tese, enquanto a densidade de 1's de $\boldsymbol{H}_2$ é de $0,58\%$, a densidade de 1's de $\boldsymbol{G}_2$ foi de $2,42\%$. Esse valor de densidade ainda qualifica $\boldsymbol{G}_2$ como matriz esparsa. Utilizando-se este procedimento adotado para obtenção da matriz $\boldsymbol{G}_2$ é possível obter códigos LDPC de taxas iguais a $(n-5L)/n$, em que L corresponde às dimensões das matrizes circulantes que pode variar na faixa $L \geq 52$.

6.1.2 Método de decodificação LDPC utilizado

O algoritmo utilizado na decodificação é o SPA, descrito no Apêndice A, de forma geral, para um corpo q-ário. Como os códigos LDPC aqui utilizados são binários, o decodificador se torna menos complexo. As inicializações são realizadas baseadas nas probabilidades *a posteriori* para os bits. Portanto, são dependentes do modelo proposto para o canal PS, apresentado na Seção 3.4. Essas inicializações também dependem do rotulamento utilizado, que foi o seguinte:

$$\begin{aligned} \boldsymbol{x}_{1=\text{ciano}} &\rightarrow 00; \\ \boldsymbol{x}_{2=\text{magenta}} &\rightarrow 01; \\ \boldsymbol{x}_{3=\text{amarelo}} &\rightarrow 10; \\ \boldsymbol{x}_{4=\text{preto}} &\rightarrow 11. \end{aligned}$$

As probabilidades condicionais que descrevem os canais PS específicos, um para cada cor de símbolo 2D, são dadas pela equação (3.6), apresentada novamente a seguir por conveniência:

$$p(\boldsymbol{y}|\boldsymbol{x}_i) = \frac{1}{(2\pi)^{3/2}|\boldsymbol{C}_i|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_i)^T \boldsymbol{C}_i^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_i)}.$$

Como se está utilizando um código binário, é necessário calcular as probabilidades condicionais para cada bit do símbolo 2D. Para o bit da esquerda, elas são dadas por:

$$p(x^1 = 0|y) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_{\text{ciano}}) + p(\boldsymbol{y}|\boldsymbol{x}_{\text{magenta}})}{\sum_{i=1}^{4} p(\boldsymbol{y}|\boldsymbol{x}_i)}, \tag{6.8}$$

$$p(x^1 = 1|y) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_{\text{amarelo}}) + p(\boldsymbol{y}|\boldsymbol{x}_{\text{preto}})}{\sum_{i=1}^{4} p(\boldsymbol{y}|\boldsymbol{x}_i)}, \tag{6.9}$$

para o bit da direita elas são dadas por:

$$p(x^2 = 0|y) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_{\text{ciano}}) + p(\boldsymbol{y}|\boldsymbol{x}_{\text{amarelo}})}{\sum_{i=1}^{4} p(\boldsymbol{y}|\boldsymbol{x}_i)}, \tag{6.10}$$

$$p(x^2 = 1|y) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_{\text{magenta}}) + p(\boldsymbol{y}|\boldsymbol{x}_{\text{preto}})}{\sum_{i=1}^{4} p(\boldsymbol{y}|\boldsymbol{x}_i)}. \tag{6.11}$$

6.2 Modulação codificada em sistemas de códigos de barra 2D

No Capítulo 4 foram revistas três técnicas bastante conhecidas de modulação codificada. Como o sistema de códigos de barra 2D apresentado utiliza apenas quatro símbolos 2D (4 cores), foi descartada a utilização de modulação codificada em treliças (TCM). Esta técnica poderia ser utilizada em sistemas de códigos de barra 2D com um número maior de símbolos 2D. Por exemplo, para um sistema com oito símbolos 2D (8 cores), a técnica de TCM poderia ser utilizada de forma análoga a sua aplicação em constelações 8-PSK.

Na Figura 6.1 são apresentadas as taxas de armazenamento em bits/bloco para o canal PS; estas foram obtidas através de simulação Monte Carlo, descrita no Apêndice B. A curva "capacidade símbolo" corresponde à máxima taxa de armazenamento (capacidade) considerando a decodificação dos símbolos. Mas como nesta tese é utilizada uma codificação binária, a curva "capacidade bit", que corresponde à máxima taxa de armazenamento (capacidade) considerando a decodificação dos dois bits separadamente, é mais adequada para determinar a máxima taxa de armazenamento, porque considera a decodificação independente dos bits. Ou seja, a curva "capacidade bit" é resultado da soma das capacidades dos dois canais binários equivalentes: canal esquerdo e canal direito. Na Figura 6.2 são apresentadas as curvas de máxima taxa de armazenamento (capacidade) para ambos os casos.

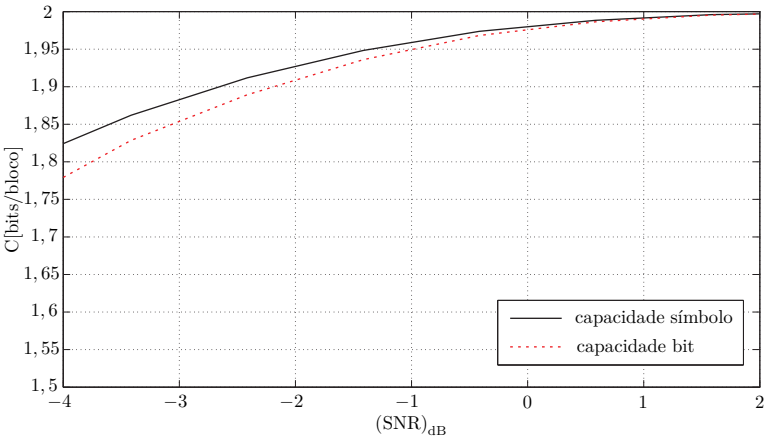


Figura 6.1: Curvas de capacidade de armazenamento.

Observe que na Figura 6.1 para a referência SNR = 0 dB, que corresponde ao canal utilizado como base nos experimentos da Seção 3.4, uma diferença muito pequena entre as taxas, de aproximadamente $0,004$ bits/bloco, é observada. Portanto, não compensa pagar o alto preço da complexidade de códigos LDPC não binários para uma margem tão pequena de ganho. Considerando-se a curva de "capacidade bit" observa-se que a taxa total de $1,9$ bits/bloco pode ser alcançada com o uso de códigos corretores de erros. Como um primeiro palpite, códigos LDPC binários de taxa $0,95$ parecem ser uma solução viável. Entretanto, observou-se que tal escolha não é apropriada uma vez que, como veremos a seguir, os dois canais de bit têm indivualmente capacidades diferentes. Deve-se observar também que, a partir da curva "capacidade bit", o limite teórico em termos de SNR para a taxa 1,9 bits/bloco de aproximadamente -2,2 dB. Sendo assim, essa deve ser considerada como a SNR alvo.

Para se poder considerar a curva "capacidade símbolo" como o limite a ser atingido, seria necessário utilizar um código LDPC em GF(4) ou, por exemplo, utilizar um decodificador semelhante ao decodificador ótimo para MLC/MSD apresentado na Seção 4.4, que explora a informação dos níveis inferiores fazendo assim com que a regra da cadeia de informação mútua seja válida.

Com a utilização da modulação codificada multinível (MLC/MSD), passa-se a trabalhar com dois canais binários equivalentes independentes; isto implica o projeto de dois códigos LDPC. Os

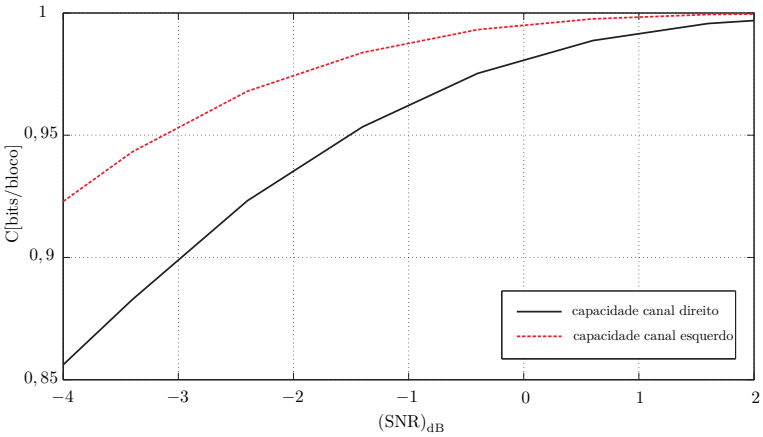


Figura 6.2: Curvas de capacidade de armazenamento dos canais binários equivalentes.

códigos LDPC componentes possuem taxas compatíveis com as capacidades de seus canais binários equivalentes, que podem ser observadas nas curvas da Figura 6.2. Através da análise destas curvas foi possível determinar a taxa adequada para os códigos em cada um dos canais binários equivalentes, lembrando-se que a taxa total anteriormente determinada foi igual a $1,9$ bits/bloco. A taxa para o código LDPC do canal esquerdo foi escolhida como igual a $0,9688$ e para o canal direito igual a $0,9313$.

Como uma última observação, antes de considerar a otimização dos códigos LDPC apresentada a seguir, deve-se mencionar que a modulação codificada com entrelaçamento de bit (BICM), apresentada na Seção 4.5, embora possua baixa complexidade, não é apropriada no contexto de canal PS. A técnica BICM é normalmente utilizada para canais com memória [63]. A ideia em BICM de entrelaçar os bits codificados serve para fazer com que bits (codificados) associados a um mesmo grupo de bits de informação sejam transmitidos em símbolos distintos, de modo que se um símbolo transmitido for severamente afetado pelas adversidades do canal, possa existir um outro símbolo também relacionado ao mesmo grupo de bits de informação que não tenha sido tão afetado pelo canal, podendo assim melhorar o desempenho. No caso aqui considerado, a codificação LPDC pseudo-aleatória já proporciona um bom embaralhamento. Naturalmente, um mesmo grupo de bits de informação é mapeado (espalhado) em bits codificados que ocupam posições arbitrárias.

Apesar de haver uma dependência do ruído em relação aos símbolos quaternários (cores) transmitidos, a codificação LDPC remove naturalmente essa memória com respeito aos bits de informação.

6.3 Otimização de códigos LDPC

Na Seção 5.3, dois métodos para otimização de códigos LDPC foram apresentados: *density evolution* e EXIT *charts*. Dada a sua simplicidade e a sua eficácia, será explorado nesta tese de doutorado apenas o método de otimização baseado em EXIT *charts*. A obtenção dos EXIT *charts* para o canal PS seguirá um processo semelhante ao apresentado para o caso do canal AWGN, porém com algumas modificações segundo as características próprias do canal PS.

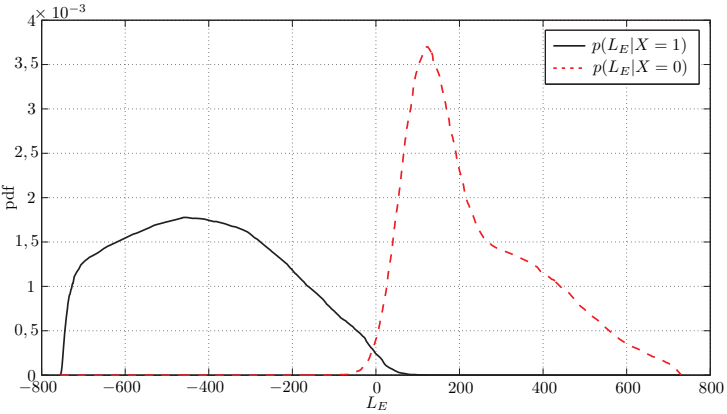


Figura 6.3: Histogramas de L_E condicionada aos bits transmitidos.

A equação (5.11), que expressa a informação mútua entre uma variável aleatória binária X em que os valores ± 1 são igualmente prováveis e uma variável aleatória Y qualquer, é também utilizada para o cálculo da informação mútua entre o bit de dados (bipolar) X e a LLR do canal binário equivalente (esquerdo e direito) associado ao canal PS. Entretanto, como veremos a seguir, a expressão para esta LLR é de difícil tratamento, inviabilizando uma solução analítica da integral em (5.11). A forma encontrada para contornar este problema foi resolvê-lo de forma numérica. Será descrito todo o procedimento utilizado na obtenção das expressões para os EXIT *charts* do canal binário equivalente esquerdo; um procedimento análogo foi utilizado para o canal binário equivalente direito.

O bit 1 é o transmitido pelo canal binário equivalente esquerdo quando é enviado através do canal PS um bloco de cor amarelo ou preto. Com base nesta observação, a LLR do canal binário equivalente esquerdo, L_E, é dada por:

$$L_E = \ln\left[\frac{\frac{1}{\sqrt{|\boldsymbol{C}_3|}}e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_3)^T\boldsymbol{C}_3^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_3)}+\frac{1}{\sqrt{|\boldsymbol{C}_4|}}e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_4)^T\boldsymbol{C}_4^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_4)}}{\frac{1}{\sqrt{|\boldsymbol{C}_1|}}e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_1)^T\boldsymbol{C}_1^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_1)}+\frac{1}{\sqrt{|\boldsymbol{C}_2|}}e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_2)^T\boldsymbol{C}_2^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_2)}}\right], \quad (6.12)$$

em que os subscritos 1, 2, 3 e 4 referem-se às quatro cores (ver Seção 3.4).

Dada a intratabilidade matemática desta expressão, foram levantados histogramas das LLRs L_E e L_D para ambos os canais (direito e esquerdo). Estes histogramas foram levantados com grande precisão, em intervalos de $0,1$ unidade, numa faixa de valores para a LLR que se estendeu de -800 a $+800$. Como exemplo, a LLR L_E para uma SNR de 0 dB é apresentada na Figura 6.3. Observe que os histogramas são equivalentes às pdfs condicionadas $p(L_E|X=1)$ e $p(L_E|X=0)$.

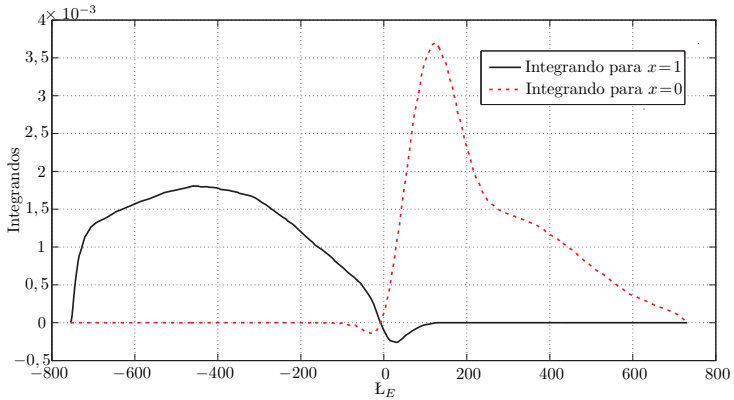


Figura 6.4: Gráfico dos integrandos da equação 6.13.

Aplicando a equação (5.11), obtem-se a expressão da informação

mútua $I(X;L_E)$:

$$\begin{aligned} I(X;L_E) &= \frac{1}{2}\sum_{x=1,0}\int_{-\infty}^{\infty} p(L_E|X{=}x)\log_2\frac{2p(L_E|X{=}x)}{p(L_E|X{=}0){+}p(L_E|X{=}1)}dL_E \\ &= \frac{1}{2}\int_{-\infty}^{\infty} p(L_E|X{=}1)\log_2\frac{2p(L_E|X{=}1)}{p(L_E|X{=}0){+}p(L_E|X{=}1)}dL_E \\ &+ \frac{1}{2}\int_{-\infty}^{\infty} p(L_E|X{=}0)\log_2\frac{2p(L_E|X{=}0)}{p(L_E|X{=}0){+}p(L_E|X{=}1)}dL_E. \quad (6.13) \end{aligned}$$

Deve-se notar que as densidades $p(L_E|X=0)$ e $p(L_E|X=1)$ foram obtidas numericamente, através dos histogramas da Figura 6.3. Com isso, os integrandos das duas integrais em (6.13) podem ser obtidos numericamente. Na Figura 6.4 são apresentadas as curvas referentes aos integrandos em função de L_E. Por fim, a solução numérica para a informação mútua $I(L_E;X)$ em (6.13), para SNR igual a 0 dB, é obtida calculando-se a média das áreas sob as curvas da Figura 6.4.

O procedimento descrito acima para o canal binário equivalente esquerdo e para SNR=0 dB foi repetido para ambos os canais binários equivalentes, partindo de uma SNR de -30 dB até 10 dB, com passos de $0,01$ dB. Os resultados são apresentados na Figura 6.5.

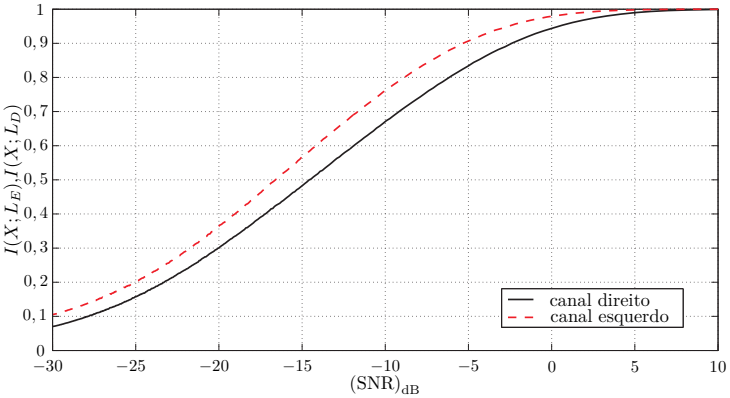


Figura 6.5: Curvas para $I(X;L_E)$ e $I(X;L_D)$ em função de SNR (dB).

A partir das curvas apresentadas na Figura 6.5 é possível calcular numericamente para ambos canais binários equivalentes o valor inicial da informação mútua utilizada nos EXIT *charts*. Quando calcula-se as funções de transferência para os EXIT *charts*, considera-se o canal de comunicação e o canal extrínseco. Na Figura 6.6 é apresentado

um diagrama de blocos para o canal PS considerando o mecanismo de decodificação dos códigos LDPC e os respectivos canais envolvidos. O canal extrínseco é o canal que existe entre os dois blocos do decodificador LDPC, enquanto que o canal de comunicação é o canal utilizado na transmissão da informação.

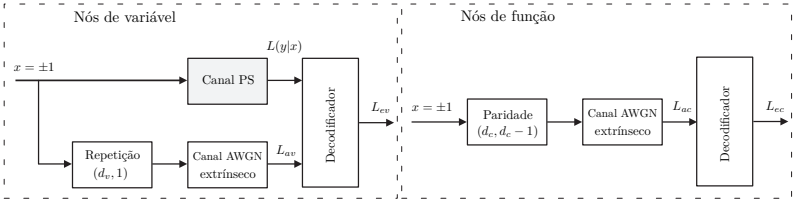


Figura 6.6: Diagrama de blocos para os códigos LDPC no canal PS.

Assim, utilizando-se as curvas para $I(X; L_E)$ e $I(X; L_D)$ e considerando-se válidas as propriedades da dualidade entre códigos de repetição e de paridade simples bem como a aproximação Gaussiana, como já mencionado na Seção 5.3.2, de acordo com [62] as funções de transferência relativas aos nós de variável e aos nós de função para canais PS são dadas respectivamente por:

$$\begin{aligned} I(X; L_{ev}) &= I_{E_v}(I_{A_v}, \lambda(x), \mathrm{SNR}) \\ &= \sum_{i=1}^{D_v} \lambda_i J\Big(\sqrt{(i-1)[J^{-1}(I_{A_v})]^2 + J^{-1}(I_o)^2}\Big), \end{aligned} \quad (6.14)$$

$$\begin{aligned} I(X; L_{ec}) &= I_{E_c}(I_{A_c}, \rho(x)) \\ &\approx \sum_{j=1}^{D_c} \rho_j \left[1 - J\left(\sqrt{(j-1)} J^{-1}(1 - I_{A_c})\right)\right], \end{aligned} \quad (6.15)$$

em que I_o corresponde ao valor da informação mútua ($I(X; L_E)$ ou $I(X; L_D)$) para a SNR do canal binário equivalente e a função $J(\cdot)$ é a apresentada na equação (5.14) obtida para o cálculo da informação mútua entre entre uma variável binária e uma variável aleatória Gaussiana consistente.

Considerar o canal extrínseco como sendo Gaussiano é justificado pelo próprio mecanismo de troca de mensagens do decodificador dos códigos LDPC, como mencionado na Seção 5.3.2. Como será demonstrado a seguir esta consideração produz bons resultados também para o canal PS. Cabe mencionar que, em [62], a mesma consideração

foi feita para a obtenção das funções EXIT para o canal com desvanecimento Rayleigh e com múltiplas antenas (MIMO).

O projeto dos códigos LDPC para o canal PS utilizando EXIT *charts* consiste em uma caminhada aleatória (do inglês *random walk*) "inteligente" através do espaço paramétrico, ou seja, nas distribuições de graus do código LDPC. O ponto-chave dessa técnica de otimização baseia-se na análise dos EXIT *charts*, que permite levar em conta deficiências do canal, que são normalmente negligenciadas por outros métodos de análise. Além disso, sua simplicidade garante uma baixa complexidade computacional.

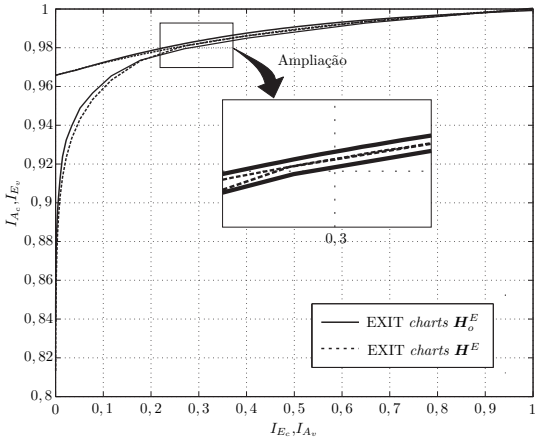


Figura 6.7: EXIT *charts* para o canal binário equivalente esquerdo.

Seguindo o algoritmo descrito na seção 5.3.3 foram realizadas otimizações de códigos LDPC para os canais binários equivalentes. Na Figura 6.7 são apresentados os EXIT *charts* para o canal binário equivalente esquerdo para uma SNR de $-1,2$ dB. Podem ser observados dois pares de curvas que correspondem aos EXIT *charts* para os códigos LDPC das matrizes $\boldsymbol{H}^E$ e $\boldsymbol{H}_o^E$ (otimizada).

Na Figura 6.8 são também apresentados os EXIT *charts*, mas para o canal binário equivalente direito para uma SNR de $-1,4$ dB. São apresentados dois pares de curvas que correspondem aos EXIT *charts* para os códigos LDPC das matrizes $\boldsymbol{H}^D$ e $\boldsymbol{H}_o^D$ (otimizada). Em ambas figuras é mostrada uma ampliação no ponto em que o "tunel" se fecha para o caso das matrizes não otimizadas. Isso significa que, se forem adotas as matrizes otimizadas, a SNR pode ser reduzida ainda mais até o ponto em que o túnel se feche.

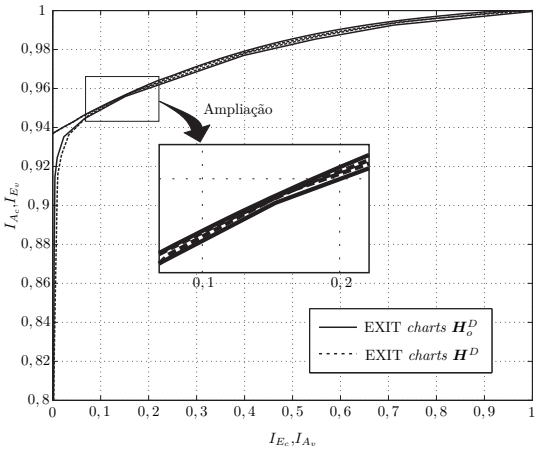


Figura 6.8: EXIT *charts* para o canal binário equivalente direito.

6.4 Resultados de simulações

Nesta seção são apresentados os resultados de simulação da probabilidade de erro de bit (BER) de alguns códigos LDPC projetados para o canal PS. A fim de permitir uma comparação entre os diferentes sistemas codificados, foi adotada para todos os sistemas uma taxa de armazenamento média de $0,95$ bits/bit de informação, que corresponde a uma taxa pré-determinada de $1,9$ bits/bloco de informação. Além disso, os códigos LDPC envolvidos foram projetados com comprimento de bloco igual a 19200 ($n = 19200$). O número de iterações dos decodificadores foi limitado a 10 e o valor da BER para cada SNR simulada foi obtido depois de encontradas 100 palavras-código decodificadas erroneamente.

Para se ter uma idéia dos ganhos conseguidos com a abordagem de codificação independente de canais binários equivalentes esquerdo e direito, proposta nesta tese, bem como dos ganhos obtidos a partir da otimização via EXIT *charts*, também uma contribuição desta tese, foram considerados três cenários distintos nas simulações.

No primeiro cenário, os bits de informação foram codificados por um único codificador LDPC binário de taxa $0,95$. Os bits codificados foram então agrupados de dois em dois para a formação dos blocos (cores) a serem submetidos ao canal PS, segundo o rotulamento proposto na Seção 6.1.2. Nas curvas de BER, este sistema será identificado por $\boldsymbol{H}$, em alusão à matriz de verificação de paridade do código LDPC

utilizado.

No segundo cenário, considerou-se a abordagem de codificação independente de canais binários equivalentes esquerdo e direito. Ou seja, a sequência de informação foi particionada em duas sub-sequências (esquerda e direita), cada uma das quais sendo codificada por um codificador LDPC independente. Os blocos (cores) a serem submetidos ao canal PS foram então formados com um bit codificado pelo codificador da esquerda e o outro pelo codificador da direita. O código LDPC do canal esquerdo possui uma taxa igual a $0,9688$, e o do canal direito, $0,9313$. Este sistema será identificado por $\boldsymbol{H}_D$ e $\boldsymbol{H}_E$, em alusão às matrizes de verificação de paridade dos códigos LDPC utilizados. Curvas serão mostradas para o desempenho de cada código (canal) individualmente, bem como o desempenho médio do sistema.

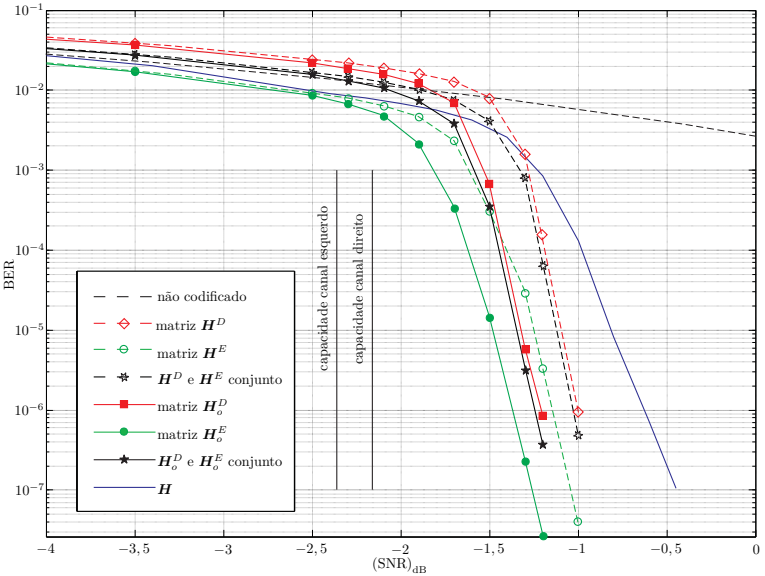


Figura 6.9: Curvas de BER para códigos LDPC submetidos ao canal PS.

O terceiro cenário é idêntico ao segundo, sendo que os códigos LDPC adotados foram otimizados via EXIT *charts*. As matrizes de verificação de paridade dos códigos LDPC otimizados foram obtidas a partir do algoritmo de otimização apresentado na Seção 5.3.3 e possuem as mesmas taxas, densidade de uns, e consequentemente a mesma complexidade de codificação e decodificação que as matrizes originais. Este sistema será identificado e curvas de desempenho serão

mostradas do mesmo modo com foi feito no cenário anterior, sendo que um subscrito "O"(otimizado) foi incluído.

Na Figura 6.9 são apresentadas as curvas de BER para todas as simulações realizadas. Além disso, são apresentados também o desempenho do canal para o caso não codificado e as máximas taxas de armazenamento (capacidades) dos canais binários equivalentes. Os polinômios de distribuição de graus para todas as matrizes utilizadas são apresentados no Apêndice D.

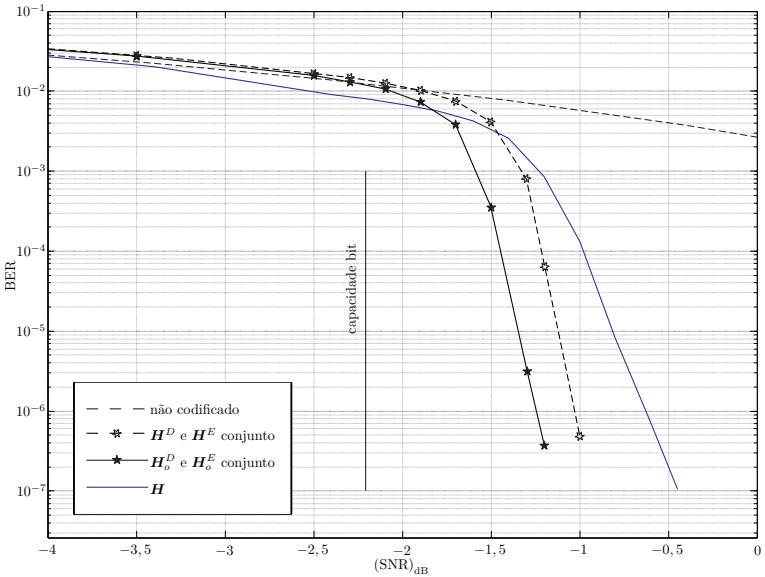


Figura 6.10: Comparação de BER para taxa total de 1,9 bits/bloco.

Na Figura 6.10, por conveniência, são reproduzidas as curvas de BER referentes a $\boldsymbol{H}$, $\boldsymbol{H}^D$ e $\boldsymbol{H}^E$ conjunto e $\boldsymbol{H}_o^D$ e $\boldsymbol{H}_o^E$ conjunto. Estas três curvas podem ser comparadas de forma justa, pois todas são equivalentes a uma taxa total de 1,9 bits/bloco.

O código que possui o "pior" desempenho é aquele da matriz $\boldsymbol{H}$, que para uma BER $= 10^{-5}$ está a 1,3 dB da capacidade de canal (considerando a curva "capacidade bit"). Os códigos das matrizes $\boldsymbol{H}^D$ e $\boldsymbol{H}^E$, considerando o desempenho conjunto, estão a 1,1 dB da SNR limite associada à capacidade de canal. E por fim os códigos das matrizes $\boldsymbol{H}_o^D$ e $\boldsymbol{H}_o^E$, que correspondem às matrizes otimizadas para os canais binários equivalentes, ficaram a apenas 0,9 dB da SNR limite associada à capacidade de canal. Fazendo uma análise

preliminar é possível afirmar que a técnica de modulação codificada MLC/MSD permitiu um melhor aproveitamento do canal, resultando num ganho de SNR de aproximadamente $0,2$ dB. Nota-se também que com a otimização dos códigos foi possível se aproximar mais $0,2$ dB da capacidade.

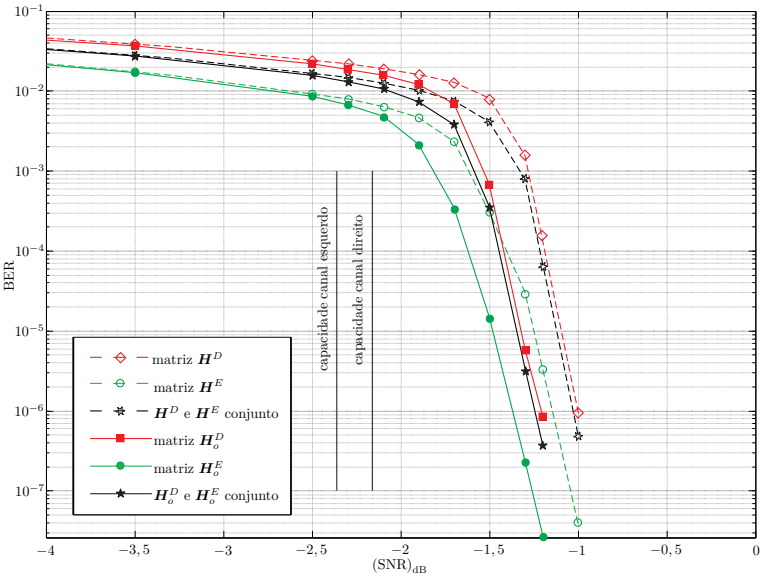


Figura 6.11: Comparação de BER dos códigos LDPC otimizados e não otimizados.

Na Figura 6.11, por conveniência, são reproduzida as curvas da Figura 6.9 excluindo-se as curvas para a matriz $\boldsymbol{H}$ e o caso não codificado. Considerando as curvas referentes ao canal esquerdo, é possível observar para a BER $= 10^{-5}$ um ganho de aproximadamente $0,2$ dB em relação ao não otimizado resultando numa distância de aproximadamente $0,9$ dB da capacidade de canal previamente estabelecida. Para o canal direito para a BER $= 10^{-5}$ coincidentemente o resultado é o mesmo, tem-se um ganho de aproximadamente $0,2$ dB em relação ao não otimizado resultando numa distância de aproximadamente $0,9$ dB da SNR limite associada à capacidade de canal.

Em todas as simulações procurou-se avaliar se durante a decodificação dos códigos LDPC ocorriam erros no decodificador, ou seja, verificar se o decodificador convergia para uma palavra-código

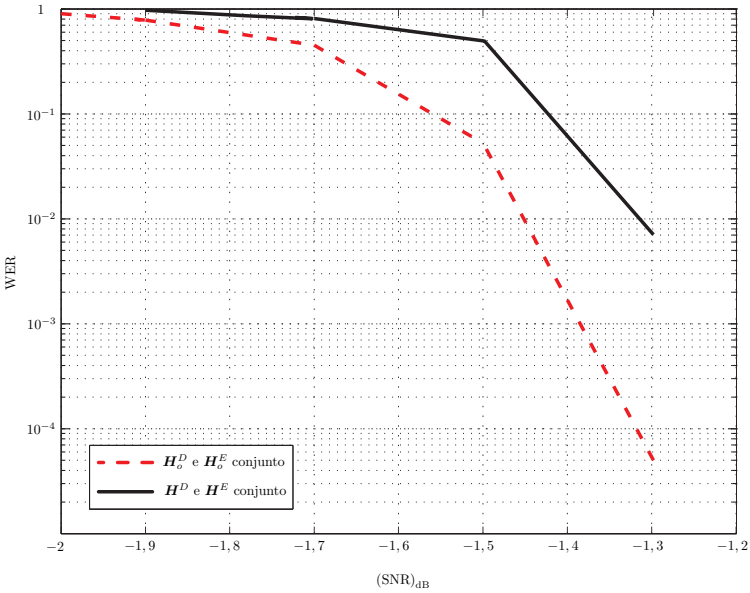


Figura 6.12: WER dos códigos LDPC otimizados e não otimizados.

diferente da transmitida. Este tipo de erro não aconteceu para nenhum dos códigos, em nenhuma das relações sinal ruído simuladas. Na Figura 6.12 são apresentados os resultados de probabilidade de erros de palavra-código. Toda vez que o decodificador não converge para uma palavra-código é contabilizado um erro. Estes erros são somados e divididos pelo número total de palavras-código transmitidas, resultando na probabilidade de erro de palavra-código (WER). Para uma aplicação prática esta é uma avaliação muito importante, porque sistemas práticos de códigos de barra 2D exigem erro zero. Para o código otimizado foi obtida uma WER de aproximadamente 5×10^{-5}, suficientemente baixa para sua aplicação em um sistema prático.

Capítulo 7

Considerações finais

Nesta tese de doutorado, foram projetados códigos LDPC para aplicações em códigos de barra 2D coloridos. Um novo codificador LDPC com complexidade de codificação linear no comprimento da palavra-código foi proposto. Além disso, descreveu-se todo o procedimento adotado no processo de otimização de códigos LDPC baseado em EXIT *charts*. Também obtiveram-se matrizes de verificação de paridade otimizadas para o modelo de canal considerado, o canal *print and scan*.

O método de codificação desenvolvido nesta tese possui caracteristicas bastante atrativas: complexidade de codificação linearmente proporcional ao comprimento da palavra código, *girth* maior que 4 e considerável liberdade de projeto para as matrizes de paridade do código. Apesar de serem encontradas técnicas de codificação semelhantes na literatura, o processo de codificação que foi proposto nesta tese se diferencia por fazer a utilização de uma matriz esparsa na codificação.

O procedimento de otimização de códigos LDPC apresentado foi utilizado especificamente para a obtenção de matrizes ótimas para os canais binários equivalentes, que foram os modelos de canais adotados nesta tese para representar o canal PS nesta aplicação em códigos de barra 2D coloridos. Entretanto, todos os conceitos e métodos utilizados podem ser aplicados a quaisquer outros modelos de canais.

Devido à alta complexidade computacional exigida para a obtenção das curvas de BER, o número de iterações do decodificador SPA foi limitado a apenas 10 iterações. Entretanto, em sistemas práticos, apenas uma decodificação é realizada por código de barras, em

contraste com centenas de milhares de realizações para se ter uma estimativa precisa da BER numa simulação de Monte Carlo. Na prática, para um único código de barras, uma decodificação de um código LDPC com, por exemplo, 200 iterações, seria bastante razoável, e produziria uma taxa de erro que se aproximaria de zero.

No sistema de códigos de barra 2D coloridos apresentado, às cores usadas nos símbolos 2D são limitadas as cores "primárias" utilizadas pelas impressoras comerciais. Com o uso de *halftoning*, ou de impressoras que utilizam mais cores, um número maior de simbolos 2D poderiam ser incorporados ao sistema, certamente aumentando sua capacidade de armazenamento.

Por último, o codificador proposto nesta tese é baseado numa matriz de verificação de paridade composta de duas submatrizes esparsas, uma das quais é inversível. Buscas por outras matrizes com estas mesmas características possibilitariam uma maior liberdade na construção e na busca por códigos LDPC ótimos.

Apêndice A

Algoritmo SPA

A.1 Inicialização

O algoritmo é inicializado enviando cada uma das mensagens Q_{ij}^a dos nós de variável x_j para os nós de função (f_i), como a probabilidade *a priori* de o j-ésimo nó de variável ser o símbolo a. Para facilitar o entendimento do algoritmo, considere a Figura A.1 que representa um grafo bipartido trocando mensagens entre os nós, lembrando que um grafo bipartido é a representação gráfica de uma matriz de verificação de paridade.

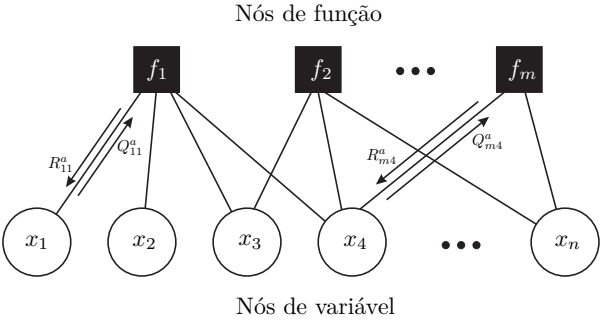


Figura A.1: Troca de mensagens em um grafo bipartido.

A.2 Atualizando os R_{ij}^a

A mensagem R_{ij}^a que o nó de função i envia para o nó de variável vizinho j corresponde à probabilidade de o nó de função i "ser satisfeito" dado que o nó de variável vizinho j está no estado a. O termo "ser satisfeito" significa que a soma de todos os nós de variável conectados ao nó de função é igual ao símbolo de síndrome (ou paridade) z_i modulo q. No decodificador de síndrome, para o caso não binário, o símbolo z_i não é necessariamente zero. Denote o evento "nó de função i "ser satisfeito"" por ζ_i. Então, a mensagem R_{ij}^a seria dada por:

$$R_{ij}^a = p(\zeta_i \mid x_j = a)$$

que, pelo teorema das probabilidades totais, poderia ser escrita como:

$$R_{ij}^a = \sum_{\boldsymbol{x}:x_j=a} p(\zeta_i \mid \boldsymbol{x}) p(\boldsymbol{x} \mid x_j = a) \tag{A.1}$$

A probabilidade $P(\zeta_i \mid \boldsymbol{x})$ em (A.1) de o nó de função i ser satisfeito é 0 ou 1 para uma dada configuração $\boldsymbol{x}$, e pode ser obtida a partir da matriz de verificação de paridade ou, equivalentemente, a partir do grafo de Tanner. Na verdade, pode-se perceber que o evento ζ_i não depende de toda a configuração $\boldsymbol{x}$, mas apenas dos x_k's relacionados aos nós de variável conectados ao nó de função i. Por outro lado, a probabilidade $p(\boldsymbol{x} \mid x_j = a)$ depende da configuração $\boldsymbol{x}$ inteira, e por isso não pode ser obtida localmente no nó de função i uma vez que numa dada iteração do algoritmo esse nó só tem acesso aos, ou só tem informação dos, nós de variável a ele conectados. Sendo assim, o algoritmo SPA realiza uma aproximação e gera, no nó de função i, um valor que apenas representa a probabilidade $p(\boldsymbol{x} \mid x_j = a)$. E isso é obtido a partir das mensagens Q que o nó de função i recebeu dos seus vizinhos variáveis no passo anterior, excluindo-se aquela que ele recebeu do próprio nó de variável j. Assim, a mensagem R_{ij}^a passa a ter o seguinte valor (aproximado):

$$R_{ij}^a = \sum_{\boldsymbol{x}:x_j=a} p(\zeta_i \mid \boldsymbol{x}) \prod_{k \in \mathcal{N}(i) \setminus j} Q_{ik}^{x_k} \tag{A.2}$$

em que $\mathcal{N}(i)$ denota o conjunto de índices de nós de variável vizinhos ao nó de função i e $\mathcal{N}(i) \setminus j$ denota os índices de todos os nós de variável vizinhos menos o nó j.

A.3 Atualizando os Q_{ij}^{a}

A mensagem Q_{ij}^{a} que o nó de variável j envia para o nó de função vizinho i representa o quanto ele acredita que seu valor corresponde ao símbolo a, supondo que todos os outros nós de função vizinhos "sejam satisfeitos", ou seja,

$$Q_{ij}^{a} = p\left(x_j = a \mid \bigcap_{i \in \mathcal{M}(j)\setminus i} \zeta_i\right) \tag{A.3}$$

em que $\mathcal{M}(j)\setminus i$ denota o conjunto de indices de nós de função vizinhos ao nó de variável j menos o nó i. Aplicando o teorema de Bayes em (A.3), e desprezando o termo que independe de a, tem-se o seguinte valor representativo para a mensagem:

$$\begin{aligned} Q_{ij}^{a} &= p(x_j = a)p\left(\bigcap_{i \in \mathcal{M}(j)\setminus i} \zeta_i \mid x_j = a\right) \\ &= \alpha_{ij} f_j^a \prod_{k \in \mathcal{M}(j)\setminus i} R_{kj}^{a} \end{aligned} \tag{A.4}$$

em que f_j^a denota a probabilidade *a priori* de o nó de variável j corresponder ao símbolo a e a constante α_{ij} é introduzida para fins de normalização de modo a garantir que $\sum_a Q_{ij}^{a} = 1$.

A.4 Tentativa de decodificação

Após o cálculo das atualizações das mensagens Q e R, para cada índice $j \in \{1, \ldots, n\}$ e possíveis estados a, tem-se:

$$\hat{n}_j = \text{argmax}_a f_j^a \prod_{k \in \mathcal{M}(j)} R_{kj}^{a} \tag{A.5}$$

O vetor $\hat{\boldsymbol{n}}$ é a tentativa de decodificação. Se ele satisfizer a equação de síndrome, $\boldsymbol{H}\hat{\boldsymbol{n}} = \boldsymbol{z}$, então a decodificação estará terminada. Caso o vetor $\hat{\boldsymbol{n}}$ não satisfaça a equação de síndrome, então uma nova iteração será necessária, e os passos descritos nas Seções A.2 e A.3 devem ser repetidos até a equação de síndrome ser satisfeita. Na prática, para evitar que o algoritmo SPA não convirja corretamente, um numero máximo de iterações é fixado.

Apêndice B

Cálculo das taxas de informação

Este apêndice tem como objetivo descrever o método utilizado para o cálculo das taxas de informação apresentadas no Capítulo 6 desta tese.

B.1 Entropia

A entropia para uma variável aleatória X composta por um alfabeto discreto é definida como:

$$H(X) = -\sum_{i=1}^{M} p(X_i) \log_b p(X_i) = E[-\log_b p(X)], \tag{B.1}$$

em que b (base do logaritmo) determina a unidade utilizada pela entropia. Em comunicações são mais utilizadas a base e correspondendo à unidade "nats" e a base 2 que corresponde à unidade "bits". A base e é muito mais conveniente considerando aplicações matemáticas, enquanto que a base 2 fornece um número mais intuitivo e direcionado à aplicação.

B.1.1 Entropia Diferencial

A entropia pode ser estendida para o caso de variáveis aleatórias contínuas. Para uma variável aleatória X contínua a entropia diferencial é definida como:

$$H(X) = -\int_{\mathbb{R}} p(x) \log_b p(x) = E[-\log_b p(X)]. \tag{B.2}$$

B.2 Entropia condicionada

A entropia condicionada de X dado um certo Y é:

$$H(X|Y) = H(X,Y) - H(Y). \tag{B.3}$$

O nome de entropia condicionada vem do fato de que

$$H(X|Y) = -\sum_{x,y} p(x,y) \log_b p(x|y), \tag{B.4}$$

em que $p(x,y)$ é a pdf conjunta para (X,Y) e $p(x|y) = p(x,y)/p(y)$ é a pdf condicionada. A equação (B.4) pode ser reescrita como:

$$H(X|Y) = -\sum_{y,x} p(x|y)p(y) \log_b p(x|y) = \sum_{y} p(y)H(X|Y=y), \tag{B.5}$$

em que

$$H(X|Y=y) = -\sum_{x} p(x|y) \log_b p(x|y). \tag{B.6}$$

B.2.1 Entropia diferencial condicionada

A entropia condicionada pode ser estendida para o caso de variáveis aleatória continuas. Para uma variável aleatória X continua a entropia diferencial condicionada é definida como:

$$H(X|Y) = -\int_{\mathbb{R}} p(x|y)p(y) \log_b p(x|y) = \int_{\mathbb{R}} p(y)H(X|Y=y). \tag{B.7}$$

B.3 Informação mútua

A informação mútua entre Y e X é dada por:

$$\begin{aligned} I(X;Y) &= H(Y) + H(X) - H(X,Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) - H(X|Y), \end{aligned} \tag{B.8}$$

e é a medida da quantidade de informação que a variável aleatória Y contém acerca da variável aleatória X, ou seja redução da incerteza de X por conhecimento de Y.

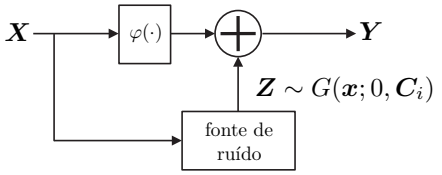


Figura B.1: Modelo simplificado do canal PS.

B.4 Cálculo das taxas de transmissão para os canais PS

Na Figura B.1 é apresentado um modelo simplificado do canal PS, em que os vetores aleatórios $\boldsymbol{X}$, $\boldsymbol{Z}$ e $\boldsymbol{Y}$ representam, respectivamente: o símbolo de entrada, o ruído e o símbolo de saída do canal.

Para se obter as taxas de transmissão no canal é necessário calcular a informação mútua para uma dada configuração do canal:

$$I(\boldsymbol{X};\boldsymbol{Y}) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X}). \tag{B.9}$$

Se X é conhecido, pode-se reescrever $H(\boldsymbol{Y}|\boldsymbol{X})$ como $H(\boldsymbol{Z})$, em que $\boldsymbol{Z}$ corresponde ao ruído do canal PS. Portanto, obtém-se:

$$I(\boldsymbol{X};\boldsymbol{Y}) = H(\boldsymbol{Y}) - H(\boldsymbol{Z}). \tag{B.10}$$

Seguindo o modelo do canal PS, tem-se:

$$H(\boldsymbol{Y}) = -\int_{\mathbb{R}^3} \sum_{k=1}^{N_G} G(\boldsymbol{y}, \boldsymbol{\mu}_k, \boldsymbol{C}_k) p(\boldsymbol{X} = \boldsymbol{x}_k) \log_b \left[\sum_{j=1}^{N_G} G(\boldsymbol{y}, \boldsymbol{\mu}_j, \boldsymbol{C}_j) p(\boldsymbol{X} = \boldsymbol{x}_j) \right], \tag{B.11}$$

$$H(\boldsymbol{Z}) = -\int_{\mathbb{R}^3} G(\boldsymbol{y} - \boldsymbol{\mu}_i; \boldsymbol{0}, \boldsymbol{C}_i) \log_b \left[G(\boldsymbol{y} - \boldsymbol{\mu}_i; \boldsymbol{0}, \boldsymbol{C}_i) \right], \tag{B.12}$$

em que

$$G(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{C}) = \frac{1}{(2\pi)^{3/2} |\boldsymbol{C}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{C}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}. \tag{B.13}$$

Para se obter as taxas máximas de transmissão para os canais PS é necessário calcular essas entropias. O procedimento adotado foi baseado em simulação de Monte Carlo, uma vez que a entropia pode ser

calculada pela equação (B.1). Como, pela Lei dos Grandes Números,

$$\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i) \;\rightarrow\; E[f(\boldsymbol{X})] \;\text{ para }\; N \rightarrow \infty, \tag{B.14}$$

tem-se que $H(\boldsymbol{X})$ pode ser obtido por

$$H(\boldsymbol{X}) = -\frac{1}{N}\sum_{i=1}^{N} \log_b \left[p(\boldsymbol{x}_i)\right]. \tag{B.15}$$

Para a simulação de Monte Carlo ($N = 10^6$) as equações utilizadas foram:

$$H(\boldsymbol{Y}) = -\frac{1}{N}\sum_{i=1}^{N} \log_b \left[\sum_{j=1}^{N_G} G(\boldsymbol{y}_i; \boldsymbol{\mu}_j, \boldsymbol{C}_j) p(\boldsymbol{X} = \boldsymbol{x}_j)\right], \tag{B.16}$$

$$H(\boldsymbol{Z}) = -\frac{1}{N}\sum_{i=1}^{N} \log_b \left[G(\boldsymbol{y}_i - \boldsymbol{x}_i; \boldsymbol{0}, \boldsymbol{C}_i)\right]. \tag{B.17}$$

Apêndice C

Distribuição Gaussiana generalizada

Uma variável aleatória X possui uma distribuição Gaussiana generalizada (*Generalized Gaussian distribution*) se a sua função de densidade de probabilidade (pdf) for dada por:

$$GGD(x;\mu,\sigma,\gamma) = \frac{1}{2\Gamma(1+1/\gamma)A(\gamma,\sigma)} e^{-\left|\frac{x-\mu}{A(\gamma,\sigma)}\right|^{\gamma}}, \quad x \in \mathbb{R} \qquad \text{(C.1)}$$

em que $\mu \in \mathbb{R}$, $\gamma, \sigma > 0$ e $A(\gamma,\sigma) = \left[\frac{\sigma^2\Gamma(1/\gamma)}{\Gamma(3/\gamma)}\right]^{1/2}$. O parâmetro μ é a média, a função $A(\gamma,\sigma)$ é um fator de escala que permite que $\mathrm{Var}(X) = \sigma^2$, e γ é o fator de forma. Quando $\gamma = 1$, a GGD corresponde a distribuição Laplaciana ou dupla exponencial e se $\gamma = 2$ a GGD corresponderá a uma distribuição Gaussiana.

Na Figura C.1 são plotadas diferentes distribuições Gaussianas generalizadas com media igual a zero ($\mu = 0$) e desvio padrão igual a um ($\sigma = 1$), mas são atribuídos diferentes valores para o fator de forma γ, os valores de γ para cada uma das curva esta apresentado na legenda da figura.

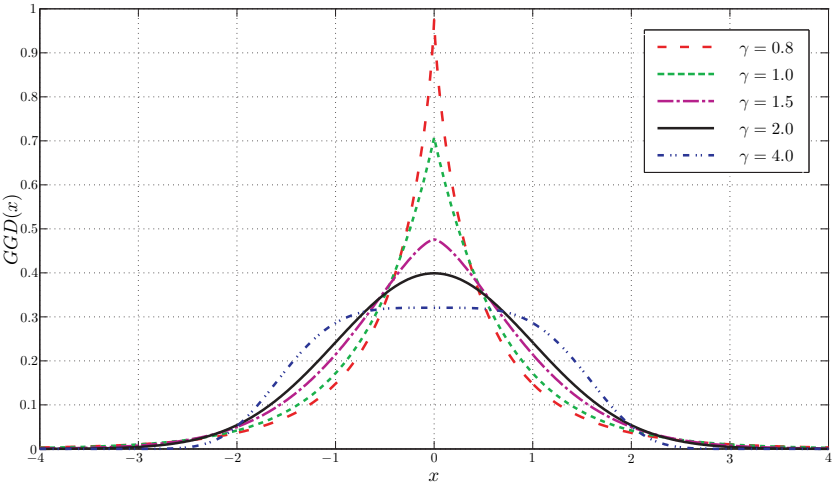


Figura C.1: Distribuição Gaussiana generalizada para diferentes valores de γ.

Apêndice D

Distribuição de graus utilizadas

Neste apêndice são apresentadas as funções de distribuição de graus para todas as matrizes de verificação de paridade utilizadas nas simulações apresentadas na Seção 6.4. A seguir são apresentados os polinômios λ e ρ, que respectivamente representam os polinômios de distribuição de graus para os nós de variável e para os nós de função. O par de polinômios a seguir é referente ao código LDPC de taxa $R = 0,95$, com comprimento de bloco de 19200 bits, a matriz obtida com esta distribuição foi apresentada nas simulações como $\boldsymbol{H}$.

$$
\begin{aligned}
\lambda(x) &= 0,01301x + 0,49901x^2 + 0,46899x^3 + 0,01898x^4, \\
\rho(x) &= 0,00066x^{43} + 0,00069x^{45} + 0,00070x^{46} + 0,00306x^{50} \\
&+ 0,00156x^{51} + 0,00478x^{52} + 0,00730x^{53} + 0,00496x^{54} \\
&+ 0,00505x^{55} + 0,01286x^{56} + 0,01658x^{57} + 0,02042x^{58} \\
&+ 0,01806x^{59} + 0,03306x^{60} + 0,03455x^{61} + 0,03322x^{62} \\
&+ 0,03761x^{63} + 0,03135x^{64} + 0,04477x^{65} + 0,05859x^{66} \\
&+ 0,05641x^{67} + 0,04892x^{68} + 0,06865x^{69} + 0,06429x^{70} \\
&+ 0,04673x^{71} + 0,04629x^{72} + 0,03576x^{73} + 0,03964x^{74} \\
&+ 0,04707x^{75} + 0,04188x^{76} + 0,02121x^{77} + 0,02149x^{78} \\
&+ 0,02056x^{79} + 0,00857x^{80} + 0,01984x^{81} + 0,00753x^{82} \\
&+ 0,00635x^{83} + 0,00900x^{84} + 0,00520x^{85} + 0,00395x^{86} \\
&+ 0,00266x^{87} + 0,00539x^{88} + 0,00136x^{89} + 0,00142x^{93}
\end{aligned}
$$

A seguir são apresentados o par de polinômios de distribuição de graus para o código LDPC utilizado no canal binário equivalente

direito. O código possui taxa $0,9313$, com comprimento de bloco de 19200 bits, a matriz obtida com esta distribuição foi apresentada nas simulações como $\boldsymbol{H}^D$.

$$\begin{aligned}
\lambda(x) &= 0,01789x + 0,70029x^2 + 0,02105x^3 + 0,03263x^4 + 0,22176x^5 \\
&+ 0,00639x^6, \\
\rho(x) &= 0,00046x^{28} + 0,00047x^{29} + 0,00049x^{30} + 0,00100x^{31} \\
&+ 0,00155x^{32} + 0,00602x^{34} + 0,00394x^{35} + 0,00867x^{36} \\
&+ 0,00890x^{37} + 0,01764x^{38} + 0,02307x^{39} + 0,02428x^{40} \\
&+ 0,03728x^{41} + 0,03346x^{42} + 0,04860x^{43} + 0,04198x^{44} \\
&+ 0,05933x^{45} + 0,05549x^{46} + 0,06186x^{47} + 0,06388x^{48} \\
&+ 0,05740x^{49} + 0,06090x^{50} + 0,05965x^{51} + 0,05010x^{52} \\
&+ 0,04685x^{53} + 0,05536x^{54} + 0,03988x^{55} + 0,03882x^{56} \\
&+ 0,02602x^{57} + 0,01734x^{58} + 0,01392x^{59} + 0,00754x^{60} \\
&+ 0,00575x^{61} + 0,00292x^{62} + 0,00494x^{63} + 0,00903x^{64} \\
&+ 0,00102x^{65} + 0,00207x^{66} + 0,00210x^{67}.
\end{aligned}$$

A seguir são apresentados o par de polinômios de distribuição de graus para o código LDPC otimizado utilizado no canal binário equivalente direito. O código possui taxa $0,9313$, com comprimento de bloco de 19200 bits, a matriz obtida com esta distribuição foi apresentada nas simulações como $\boldsymbol{H}_o^D$.

$$\begin{aligned}
\lambda(x) &= 0,01606x + 0,70302x^2 + 0,02105x^3 + 0,03263x^4 + 0,22724x^5, \\
\rho(x) &= 0,10005x^{44} + 0,21375x^{45} + 0,11754x^{46} + 0,08496x^{47} \\
&+ 0,02053x^{48} + 0,12489x^{49} + 0,07988x^{50} + 0,11366x^{51} \\
&+ 0,14472x^{53}.
\end{aligned}$$

A seguir são apresentados o par de polinômios de distribuição de graus para o código LDPC utilizado no canal binário equivalente esquerdo. O código possui taxa $0,9688$, com comprimento de bloco de 19200 bits, a matriz obtida com esta distribuição foi apresentada nas

simulações como $\boldsymbol{H}^E$.

$$
\begin{aligned}
\lambda(x) &= 0,00969x + 0,66548x^2 + 0,17034x^3 + 0,14794x^4 + 0,00656x^5, \\
\rho(x) &= 0,00136x^{85} + 0,00275x^{86} + 0,00278x^{87} + 0,00281x^{88} \\
&+ 0,00711x^{89} + 0,00144x^{90} + 0,00872x^{91} + 0,01028x^{92} \\
&+ 0,01187x^{93} + 0,01350x^{94} + 0,01515x^{95} + 0,02449x^{96} \\
&+ 0,03248x^{97} + 0,02343x^{98} + 0,03629x^{99} + 0,04302x^{100} \\
&+ 0,04988x^{101} + 0,07636x^{102} + 0,06561x^{103} + 0,05630x^{104} \\
&+ 0,04847x^{105} + 0,06074x^{106} + 0,04768x^{107} + 0,04811x^{108} \\
&+ 0,04162x^{109} + 0,03149x^{110} + 0,04590x^{111} + 0,04986x^{112} \\
&+ 0,03054x^{113} + 0,01631x^{114} + 0,02010x^{115} + 0,01659x^{116} \\
&+ 0,01487x^{117} + 0,01312x^{118} + 0,00378x^{119} + 0,00762x^{120} \\
&+ 0,00576x^{121} + 0,00387x^{122} + 0,00195x^{123} + 0,00394x^{124} \\
&+ 0,00203x^{128}.
\end{aligned}
$$

A seguir são apresentados o par de polinômios de distribuição de graus para o código LDPC otimizado utilizado no canal binário equivalente esquerdo. O código possui taxa $0,9313$, com comprimento de bloco de 19200 bits, a matriz obtida com esta distribuição foi apresentada nas simulações como $\boldsymbol{H}_o^E$.

$$
\begin{aligned}
\lambda(x) &= 0,00750x + 0,66876x^2 + 0,17034x^3 + 0,15340x^4, \\
\rho(x) &= 0,08045x^{101} + 0,12672x^{102} + 0,09678x^{103} + 0,27322x^{104} \\
&+ 0,07689x^{105} + 0,10629x^{106} + 0,10557x^{107} + 0,05327x^{108} \\
&+ 0,00520x^{109} + 0,03674x^{110} + 0,03530x^{111} + 0,00356x^{112}.
\end{aligned}
$$

Apêndice E

Algoritmo de otimização de códigos LPDC via EXIT *charts*

Neste apêndice são apresentados maiores detalhes sobre o passo fundamental e também o fluxograma E.2, referentes ao algoritmo descrito na Seção 5.3.3.

E.1 Adaptação dos perfis de graus

O passo fundamental corresponde à menor alteração que se pode realizar na distribuição dos graus sem alterar o número total de conexões (número de 1's na matriz $\boldsymbol{H}$) entre os nós de variável e função. Para facilitar o entendimento, o conceito de passo fundamental será mostrado através de um exemplo.

Na Figura E.1 é apresentado um exemplo utilizando nós de função. Nesse caso, o passo fundamental é realizado com o objetivo de alterar o polinômio $\rho(x)$ na forma de um incremento do número de nós de função de grau i, permanecendo fixo o polinômio $\lambda(x)$. A cada passo, como nesse do exemplo, no mínimo quatro nós são afetados, consequentemente quatro coeficientes do polinômio $\rho(x)$ são modificados.

Note que no exemplo são apresentados quatro grupos de nós. Antes do passo fundamental existiam 15 nós de grau i, 20 de grau $i+1$, 10 de grau j e 25 de grau $j+1$. A modificação realizada consiste em desligar uma das conexões de um dos nós de grau $i+1$ e conectá-la a um outro nó qualquer do grupo de nós com grau j. No novo arranjo, passá-se a ter 16 nós de grau i, 19 nós de grau $i+1$, 9 de grau j e 26 de grau $j+1$.

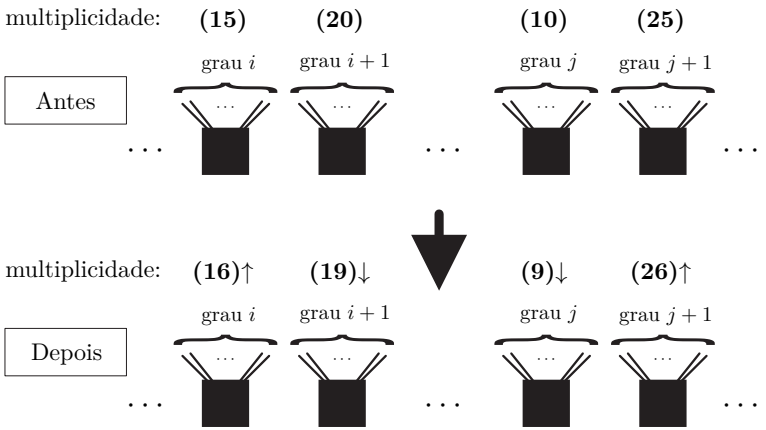


Figura E.1: Exemplo de um passo fundamental qualquer.

Para alterar o polinômio $\lambda(x)$, o mesmo procedimento apresentado para os nós de função deve ser aplicado aos nós de variável.

E.2 Fluxograma

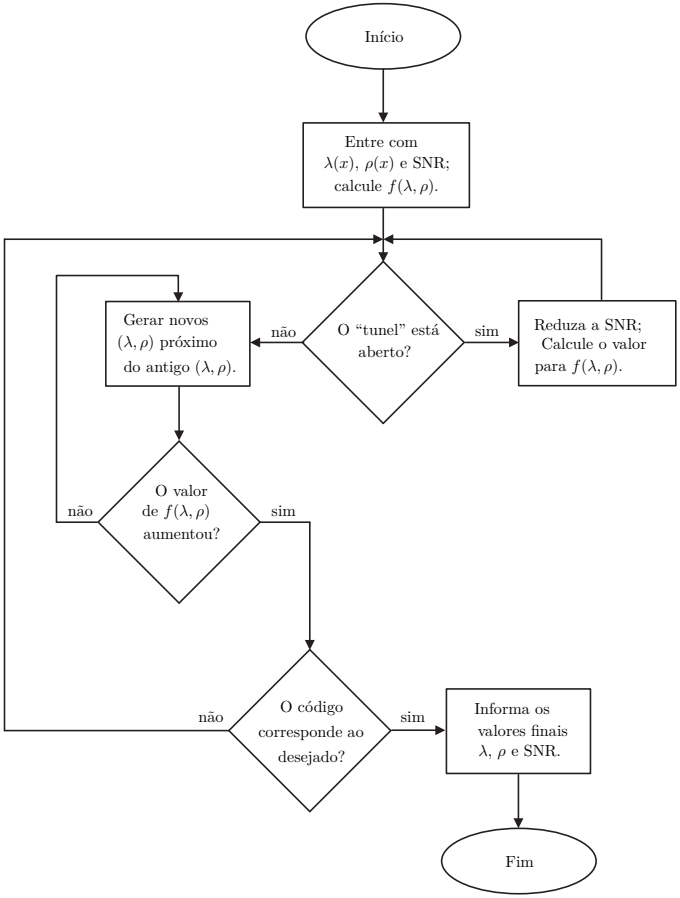


Figura E.2: Fluxograma referente ao algoritmo de otimização de LDPC baseado em EXIT *charts*.

Referências Bibliográficas

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 379 – 423 and 623 – 656, 1948.

[2] T. M. Cover and J. A. Tomas, *Elements of Information Theory*. Nova Iorque: John Wiley & Sons, 1991.

[3] R. G. Gallager, *Low-Density Parity-Check Codes*. Massachusetts: MIT Press, 1963.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)," *IEEE International Conference on Communications (ICC)*, vol. 2, Maio 1993.

[5] C. B. Schlegel and L. C. Pérez, *Trellis and Turbo Coding*. IEEE Press, Wiley-Interscience, 2004.

[6] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645 – 1646, Agosto 1996.

[7] T. J. Richardson and R. Urbanke, "The capacity of low-density paritycheck codes under message-passing decoding," *IEEE Trans. Inform.Theory*, vol. 47, Fevereiro 2001.

[8] N. Degara-Quintela and F. Pérez-González, "Visible encryption: Using paper as a secure channel," *Proceedings of SPIE*, vol. 5020, pp. 413 – 422, 2003.

[9] P. V. K. Borges and J. Mayer, "Text luminance modulation for hardcopy watermarking," *Elsevier - Sinal Processing*, vol. 87, pp. 1754 – 1771, 2007.

[10] P. V. K. Borges, J. Mayer, and E. Izquierdo, "Robust and transparent color modulation for text data hiding," *IEEE Transactions on Multimidia*, vol. 10, no. 8, pp. 1479 – 1489, Dezembro 2008.

[11] K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, and I. El-Khalil, "'print and scan' resilient data hiding in images," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, pp. 464 – 478, Dezembro 2006.

[12] X. Liu, "Analysis and reduction of moire patterns in scanned halftone pictures," Ph.D. dissertation, Virginia Polytechnic, Maio 1996.

[13] E. J. Stollnitz, V. Ostromoukhov, and D. H. Salesin, "Reproducing color images using custom inks," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1998, pp. 267 – 274.

[14] H. R. Kang, *Digital Color Halftoning*. Bellingham, WA, USA: Society of Photo-Optical Instrumentation Engineers (SPIE), 1999.

[15] D. L. Lau and G. R. Arce, *Modern Digital Halftoning, Second Edition*. Boca Raton, FL, USA: CRC Press, Inc., 2007.

[16] R. Levien, "Output dependent feedback in error diffusion halftoning," pp. 280 – 282, 1992.

[17] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall Information and System Sciences Series, 1989.

[18] F. T. R. . D, "M-commerce, an emerging sector," *http://www.francetelecom.com/sirius/rd/en/ddm/en/technologies/ddm200405/techfiche4.php*, Maio 2004.

[19] R. Villán, S. Voloshynovskiy, O. J. Koval, and T. Puri, "Multilevel 2d bar codes: towards high capacity storage modules for multimedia security and management," *IEEE on Information Forensics and Security*, vol. 1, no. 4, pp. 405 – 419, Dezembro 2006.

[20] R. Villán, S. Voloshynovskiy, O. J. Koval, and T. Pun, "Multilevel 2d bar codes: toward high-capacity storage modules for multimedia security and management," in *Security, Steganography, and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, E. J. Delp and P. W. Wong, Eds., vol. 5681. SPIE, 2005, pp. 453 – 464.

[21] G. Jancke, "High capacity color barcode technology," http://research.microsoft.com/en-us/projects/hccb/about.aspx, 2009.

[22] D. Parikh and G. Jancke, "Localization and segmentation of a 2d high capacity color barcode," in *IEEE Workdhop on Applications of Computer Vision*, 2008, pp. 1 – 6.

[23] J. Mayer, J. C. M. Bermudez, A. P. Legg, B. F. Uchôa-Filho, D. Mukherjee, A. Said, R. Samadani, and S. Simske, "Design of high capacity 3D print codes aiming for robustness to the PS channel and external distortions," in *Proc. IEEE International Conference on Image Processing*, Cairo, Egito, Novembro 2009.

[24] J. Mayer, J. C. M. Bermudez, A. P. Legg, B. F. Uchôa-Filho, D. Mukherjee, A. Said, S. Simske, and R. Samadani, "Design of high capacity 3D print codes with visual cues aiming for robustness to the PS channel and external distortions," in *Proc. IEEE International Workshop on Multimedia Signal Processing - MMSP'09*, Rio de Janeiro, Brasil, Outubro 2009.

[25] G. Ungerboeck and I. Csajka, "On improving data-link performance by increasing the channel alphabet and introducing sequence coding," in *International Symposium on Information Theory*, Ronneby, Sweden, Junho 1976.

[26] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE transactions of Information Theory*, vol. IT-28, no. 1, pp. 55 – 67, Janeiro 1982.

[27] G. Forney, R. Gallager, C. Lang, F. Longstaff, and S. Quereshi, "Efficient modulation for band-limited channels," *IEEE Jornal of Selected Areas on Communications*, vol. SAC-2, no. 5, pp. 632 – 647, Setembro 1984.

[28] A. R. Calderbanck and N. J. A. Sloane, "An eight-dimensional trellis code," vol. 74, pp. 757 – 759, Maio 1986.

[29] S. Wilson, "Rate 5/6 trellis-coded 8-psk," *IEEE transactions on Communications*, vol. COM-34, no. 10, pp. 1045 – 1049, Outubro 1986.

[30] A. R. Calderbanck and N. J. A. Sloane, "New trellis codes based on lattices and cosets," *IEEE Transactions on Information Theory*, vol. IT-33, no. 2, pp. 177 – 195, Março 1987.

[31] L. Wei, "Trellis-coded modulation with multidimensional constelations," *IEEE Transactions on Information Theory*, vol. IT-33, pp. 483 – 501, Julho 1987.

[32] D. Divsalar and M. Simon, "Multiple trellis coded modulation (mtcm)," *IEEE Transactions on Communications*, vol. 36, pp. 410 – 419, Abril 1988.

[33] M. Rouanne and D. Costello, "A lower bound on the minimum euclidean distance of trellis coded modulation schemes," *IEEE Transactions on Information Theory*, vol. 34, pt. I, pp. 1011 – 1020, Setembro 1988.

[34] S. Pietrobon, R. Deng, A. Lafanechre, G. Ungerboeck, and D. Costello, "Trellis-coded multidimensional phase modulation," *IEEE Transactions on Information Theory*, vol. 36, no. 1, pp. 63 – 89, Janeiro 1990.

[35] G. D. Forney Jr. and G. Ungerboeck, "Modulation and coding for linear gaussian channels," *Information Theory, IEEE Transactions on*, vol. 44, no. 6, pp. 2384 – 2415, Outubro 1998.

[36] U. Wachsmann, R. Fischer, and J. Huber, "Multilevel codes: theoretical concepts and practical design rules," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, pp. 1361 – 1391, Julho 1999.

[37] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *Information Theory, IEEE Transactions on*, vol. 23, no. 3, pp. 371 – 377, Maio 1977.

[38] J. Hou, P. Siegel, L. Milstein, and H. Pfister, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes," *Information Theory, IEEE Transactions on*, vol. 49, no. 9, pp. 2141 – 2155, Setembro 2003.

[39] E. Zehavi, "8-psk trellis codes for a rayleigh channel," *Communications, IEEE Transactions on*, vol. 40, no. 5, pp. 873 – 884, Maio 1992.

[40] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions Information Theory*, vol. IT-8, pp. 21 – 28, Janeiro 1962.

[41] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on information Theory*, vol. 45, pp. 399 – 431, Março 1999.

[42] D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding. 5th IMA Conference ed. Colin Boyd, Lecture Notes in Computer Science*, no. 1025, Springer, Berlim, 1995, pp. 100 – 111.

[43] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs and belief propagation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 585 – 598, 1998.

[44] ——, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 569 – 584, 2001.

[45] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619 – 637, 2001.

[46] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. IT-27, no. 5, pp. 533 – 547, Setembro 1981.

[47] G. Lechner and J. Sayir, "On the convergence of log-likelihood values in iterative decoding," in *Mini-Workshop on Topics in Information Theory*, Essen, Alemanha, Setembro 2002.

[48] Y. Mao and A. H. Banihashemi, "Decoding low-density parity-check codes with probabilistic scheduling," *IEEE Communication Letters*, vol. 5, no. 10, pp. 414 – 416, Outubro 2001.

[49] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 219 – 230, Fevereiro 1998.

[50] *On Iterative Decoding and the Two-Way Algorithm*, Brest, Frana, Setembro 1997.

[51] F. R. Kschischang, B. J. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498 – 519, Fevereiro 2001.

[52] F. R. Kschischang, "Codes defined on graphs," *IEEE Communications Magazine*, pp. 118 – 125, Agosto 2003.

[53] T. Etzion, A. Trachtenberg, and A. Vardy, "Which codes have cycle-free tanner graphs," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2173 – 2181, Setembro 1999.

[54] S. young Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Communications Letters*, vol. 5, pp. 58 – 60, 2001.

[55] T. J. Richardson and R. L. Urbanke, "The capacity of Low-Density Parity Check codes under message-passing decoding," vol. 47, no. 2, pp. 599 – 618, Fevereiro 2001.

[56] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 657 – 670, Fevereiro 2001.

[57] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. on Communications*, vol. 49, no. 4, pp. 1727 – 1737, Outubro 2006.

[58] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *Information Theory, IEEE Transactions on*, vol. 50, no. 11, pp. 2657 – 2673, Novembro 2004.

[59] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *Information Theory, IEEE Transactions on*, vol. 42, no. 2, pp. 429 – 445, Março 1996.

[60] M. Ardakani and F. Kschischang, "A more accurate one-dimensional analysis and design of irregular LDPC codes," *Communications, IEEE Transactions on*, vol. 52, no. 12, pp. 2106 – 2114, Dezembro 2004.

[61] A. Worthen and W. Stark, "Unified design of iterative receivers using factor graphs," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 843 – 849, Fevereiro 2001.

[62] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *Communications, IEEE Transactions on*, vol. 52, no. 4, pp. 670 – 678, Abril 2004.

[63] M. Franceschini, G. Ferrari, and R. Raheli, *LDPC Coded Modulations*. Springer Publishing Company, Incorporated, 2009.

[64] R. Storn and K. Price, "Differential evolutiona simple and efficient heuristic adaptive scheme for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, pp. 341 – 359, 1997.

[65] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1788–1793, 2004.

[66] C. Zhixiong and Y. Jinsha, "A construction of linearly encodable QC-LDPC codes by grouping cyclic shift and block elimination," in *CCCM '08: Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 304–308.

[67] D. Haley, A. Grant, and J. Buetefuer, "Iterative encoding of low-density parity-check codes," in *IEEE Global Telecommunication Conference (Globecom'2002)*, vol. 1, 2002, pp. 1303–1307.