



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Lucas Lui Motta

**Um Método de Compressão de Áudio Baseado  
na Decomposição do Sinal em Sub-bandas  
Wavelet e Codificação dos Coeficientes  
Wavelet Mais Expressivos**

Campinas

2019

Lucas Lui Motta

# **Um Método de Compressão de Áudio Baseado na Decomposição do Sinal em Sub-bandas Wavelet e Codificação dos Coeficientes Wavelet Mais Expressivos**

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Telecomunicações e Telemática.

Orientador: Prof. Dr. Max Henrique Machado Costa

Co-orientador Prof. Dr. Rafael Santos Mendes

Este exemplar corresponde à versão final da dissertação do aluno Lucas Lui Motta, e orientada pelo Prof. Dr. Max Henrique Machado Costa

---

Campinas

2019

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

M858m Motta, Lucas Lui, 1992-  
Um método de compressão de áudio baseado na decomposição do sinal em sub-bandas wavelet e codificação dos coeficientes wavelet mais expressivos / Lucas Lui Motta. – Campinas, SP : [s.n.], 2019.

Orientador: Max Henrique Machado Costa.

Coorientador: Rafael Santos Mendes.

Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Compressão de dados (Telecomunicações). 2. Wavelets (Matemática) - Processamento de dados. 3. Análise de regressão - Processamento de dados. 4. MP3 (Padrão de codificação de áudio). 5. Processamento de Sinais. I. Costa, Max Henrique Machado, 1950-. II. Mendes, Rafael Santos, 1957-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** An audio compression method based on the signal decomposition into wavelet sub-bands and the encoding of expressive wavelet coefficients

**Palavras-chave em inglês:**

Data compression (Telecommunication)

Wavelets (Mathematics)

Regression analysis - Data processing

Algorithms - Data processing

**Área de concentração:** Telecomunicações e Telemática

**Titulação:** Mestre em Engenharia Elétrica

**Banca examinadora:**

Max Henrique Machado Costa [Orientador]

Fernando Oscar Runstein

Tiago Fernandes Tavares

**Data de defesa:** 30-07-2019

**Programa de Pós-Graduação:** Engenharia Elétrica

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0003-2682-6700>

- Currículo Lattes do autor: <http://lattes.cnpq.br/3578130991399214>

## COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

**Candidato:** Lucas Lui Motta RA: 192701

**Data da Defesa:** 30 de julho de 2019

**Título da Tese:** “Um Método de Compressão de Áudio Baseado na Decomposição do Sinal em Sub-bandas Wavelet e Codificação dos Coeficientes Wavelet Mais Expressivos”.

Prof. Dr. Max Henrique Machado Costa (Presidente)

Prof. Dr. Fernando Oscar Runstein

Prof. Dr. Tiago Fernandes Tavares

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

# Agradecimentos

A Deus por todas as oportunidades proporcionadas em minha vida.

Agradeço a minha querida família, especialmente aos meus pais e irmão.

Ao meu pai, agradeço apoio e incentivo em todos os dias de estudo, lembrando-me sempre a importância daquilo que eu estava fazendo, e se esforçando para manter a família bem.

A minha mãe, agradeço a sua força de vontade em se permanecer feliz diante das dificuldades que a cercam, muitas vezes se sacrificando para não me interromper durante os estudos, e seu carinho nos momentos em que estive ao seu lado.

A meu irmão, o companheirismo no dia-a-dia e também o suporte aos meus pais nos momentos em que estive ausente, auxiliando-os física e psicologicamente.

Tenho muito a agradecer ao meu orientador, Prof. Max, e co-orientador, Prof. Rafael, que me apoiaram no desenvolvimento desse trabalho, de maneira competente e aplicada.

Gostaria de agradecer a minha amada Cynthia, que sem seu carinho e apoio não teria conseguido ultrapassar alguns obstáculos durante a confecção desse trabalho.

Agradeço também aos amigos e colegas que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Por fim, agradeço a toda comunidade de servidores da Unicamp o suporte que me foi dado.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Lucas Lui Motta  
Campinas, 30 de julho de 2019

*For me, data compression is more than a manipulation of numbers;  
it is the process of discovering structures that exist in the data”  
(Sayood, 2012)*

# Resumo

Nesta dissertação propõe-se um método de compressão de áudio que se baseia na decomposição do sinal de entrada em sub-bandas wavelets e na codificação dos coeficientes mais expressivos. O esquema de codificação é composto principalmente da ordenação e aproximação polinomial dos coeficientes, sendo a ordenação feita de tal forma a possibilitar uma adequada aproximação polinomial. Além disso, o método faz uso de codificação de entropia para melhorar a eficiência do codificador e também de um sistema adaptativo de controle de taxa versus distorção para obter uma taxa de bits aproximadamente constante na saída do codificador. Discute-se ainda sobre a flexibilidade do codificador proposto com diferentes configurações e os melhores resultados são comparados com outros codificadores de áudio em termos da relação sinal-ruído e qualidade de áudio perceptiva.

**Palavras-chaves:** Compressão de Áudio; Sub-bandas Wavelet; Ordenação de Coeficientes; Aproximação Polinomial; Codificação de Entropia; Controle de Taxa Versus Distorção.

# Abstract

This dissertation proposes an audio compression method based on the decomposition of input signal into wavelet subbands and the coding of most expressive coefficients. The coding scheme is composed mainly by the ordering and polynomial approximation of coefficients, the ordering being made for a suitable polynomial approximation. In addition, the method makes use of entropy coding to improve the efficiency of the encoder and also of an adaptive system of rate versus distortion control to obtain an approximate constant bit rate at the encoder output. It also discusses the flexibility of the proposed encoder with different configurations and the best results are compared with other audio encoders in terms of signal-to-noise ratio and perceptual audio quality.

**Keywords:** Audio Compression; Wavelet Sub-bands; Coefficients Ordering; Polynomial Approximation; Entropy Coding; Rate Versus Distortion Control.



# Lista de ilustrações

Figura 2.1 – Sistema de comunicação: (a) sem codificação (b) com codificação. . . .	19
Figura 2.2 – Fonte de informação (a) na forma original, (b) após a compressão sem perdas e (c) após a compressão com perdas. . . . .	20
Figura 2.3 – Técnicas de codificação classificadas pelo comprimento do dado original e codificado. . . . .	21
Figura 2.4 – Diagrama de desempenho dos sistemas de compressão. . . . .	23
Figura 2.5 – Comparação do ELC da ISO 226:2003 em relação: a) Robinson e Daddison e b) Fletcher e Munson. . . . .	25
Figura 2.6 – Tipos de medição de qualidade de áudio codificado. . . . .	28
Figura 2.7 – A estrutura comum para a medição de qualidade de áudio perceptual. .	29
Figura 2.8 – Efeito de descontinuidades no processo de segmentação do áudio. . . .	31
Figura 2.9 – Janelas frequentemente utilizadas no processamento de sinais. . . . .	32
Figura 2.10–Caixas de Heisenberg do (a) domínio do tempo, (b) transformada de Fourier, (c) STFT com uma janela estreita e (d) STFT com uma janela ampla. . . . .	34
Figura 2.11–Caixas de Heisenberg da (a) transformada de Wavelet padrão e, da decomposição de pacotes wavelets: (b) incompleta e (c) completa. . . .	35
Figura 2.12–Esquema de: (a) decomposição e (b) reconstrução de um sinal pela FWT.	39
Figura 2.13–Ilustração das respostas em amplitude e a seletividade dos filtros. . . .	40
Figura 2.14–Esquema de decomposição para multi-níveis de transformada. . . . .	40
Figura 2.15–Ilustração das respostas em amplitude para multi-níveis de transformada.	41
Figura 2.16–Estrutura da árvore de decomposição DWT. . . . .	41
Figura 2.17–Estrutura da árvore WPD do tipo: (a) incompleto (ou personalizada) e (b) árvore balanceada (ou árvore binária cheia). . . . .	42
Figura 2.18–Exemplo de regressão: (a) dados originais e (b) dados aproximados com regressão. . . . .	47
Figura 3.1 – Fluxograma do codificador proposto. . . . .	48
Figura 3.2 – Ilustração do processo de segmentação de áudio do codificador. . . . .	50
Figura 3.3 – Coeficientes dos filtros de decomposição e reconstrução do codificador.	52
Figura 3.4 – Processo de decomposição por pacotes wavelet do codificador. . . . .	53
Figura 3.5 – Exemplo das faixas de frequências das sub-bandas geradas na WPD. .	54
Figura 3.6 – Comparação do limiar auditivo definido para o codificador proposto e o padronizado pela ISO 226:2003. . . . .	55
Figura 3.7 – A decomposição da curva do limiar auditivo nas sub-bandas wavelets. .	56

Figura 3.8 – Exemplo de uma sub-banda (a) eliminada e uma (b) preservada. . . . .	57
Figura 3.9 – Exemplo do processo de eliminação dos coeficientes em uma sub-banda preservada: (a) antes e (b) depois. . . . .	57
Figura 3.10–Exemplo de ordenação dos coeficientes feita de forma crescente de valor. . . . .	59
Figura 3.11–Exemplo de ordenação dos coeficientes feita de forma crescente de valor. . . . .	59
Figura 3.12–Posições a serem armazenadas: (a) antes e (b) depois da eliminação. . . . .	60
Figura 3.13–Pontos críticos considerados na aproximação polinomial pelo MMQP. . . . .	61
Figura 3.14–Um exemplo da curva estimada por aproximação polinomial. . . . .	62
Figura 3.15–Ilustração das categorias do código de entropia. . . . .	63
Figura 3.16–As duas estruturas de árvores de Huffman possíveis para 4 símbolos. . . . .	64
Figura 3.17–Exemplo da 1ª e 10ª combinação de folhas para sinalização das categorias. . . . .	65
Figura 3.18–Exemplo de contagem dos coeficientes preservados por categoria. . . . .	67
Figura 3.19–Quantização utilizada na codificação dos coeficientes pelo método B. . . . .	68
Figura 3.20–Fluxograma do RDC incorporado ao codificador proposto. . . . .	70
Figura 3.21–Diagrama de estados do sistema RDC. . . . .	71
Figura 3.22–O formato do <i>bitstream</i> para cada bloco de áudio codificado. . . . .	73
Figura 3.23–Detalhamento dos dados e suas representações em cada trecho do <i>bits-</i> <i>tream</i> . . . . .	74
Figura 3.24–Fluxograma do processo de decodificação. . . . .	75
Figura 4.1 – Desempenho do codificador para diferentes níveis de profundidade $u$ . . . . .	77
Figura 4.2 – Resolução do limiar de audição nos diferentes níveis de profundidade. . . . .	78
Figura 4.3 – Desempenho do codificador para diferentes tamanhos de sub-bandas $v$ . . . . .	79
Figura 4.4 – Comparação de desempenho de 100 sinais de áudio de 32 a 196 kbps. . . . .	80
Figura 4.5 – Comparação de SNR usando amostras de áudio por gênero. . . . .	81
Figura 4.6 – Comparação de desempenho de 100 amostras de áudio de 32 a 196 kbps. . . . .	82
Figura 4.7 – Comparação de desempenho de 100 sinais de áudio de 32 a 196 kbps. . . . .	83

# Lista de tabelas

Tabela 2.1 – Métrica ODG do método objetivo de qualidade de áudio percebido. . .	29
Tabela 2.2 – Qualidade de som versus representação digital do sinal de áudio. . . .	30
Tabela 3.1 – A Lista de Polinômios para a Aproximação Polinomial . . . . .	61
Tabela 3.2 – Um exemplo das categorias formadas pelo código de entropia. . . . .	64
Tabela 3.3 – Nomes dados as folhas da árvore com tamanho variado de código. . . .	64
Tabela 3.4 – Tabela com as 12 combinações das categorias nas folhas. . . . .	65
Tabela 3.5 – O tamanho total das palavras códigos usando cada combinação. . . .	66
Tabela 3.6 – Códigos para o decodificar identificar a combinação de folhas utilizada.	67

# Lista de Acrônimos

<b>AAC</b>	Advanced Audio Coding
<b>ABR</b>	Taxa de Bits Média, do inglês, <i>Average Bit Rate</i>
<b>AC</b>	Audio Coding
<b>CB</b>	Consumo de Bits
<b>CBR</b>	Taxa de Bits Constante, do inglês, <i>Constant Bit Rate</i>
<b>CD</b>	Disco Compacto, do inglês, <i>Compact Disk</i>
<b>CF</b>	Fator de Compressão, do inglês, <i>Compression Factor</i>
<b>CR</b>	Taxa de Compressão, do inglês, <i>Compression Rate</i>
<b>CFT</b>	Transformada Contínua de Fourier, do inglês, <i>Continuous Fourier Transform</i>
<b>CWT</b>	Transformada Contínua de Wavelet, do inglês, <i>Continuous Wavelet Transform</i>
<b>DFT</b>	Transformada Discreta de Fourier, do inglês, <i>Discrete Fourier Transform</i>
<b>DVD</b>	Digital Video Disc
<b>DWT</b>	Transformada Discreta de Wavelet, do inglês, <i>Discrete Wavelet Transform</i>
<b>ELC</b>	Contorno de Igual Sonoridade, do inglês, <i>Equal-loudness Contour</i>
<b>HDTV</b>	High Definition Television
<b>ISO</b>	International Organization for Standardization
<b>ITU</b>	International Telecommunication Union
<b>JPEG</b>	Joint Photographic Experts Group
<b>MMQ</b>	Método dos Mínimos Quadrados Ordinários
<b>MMQP</b>	Método dos Mínimos Quadrados Ordinários Ponderado
<b>MPEG</b>	Moving Picture Expert Group
<b>ODG</b>	Grau de Diferença Objetivo, do inglês, <i>Objective Difference Grade</i>
<b>PSNR</b>	Relação Sinal-Ruído de Pico, do inglês, <i>Peak Signal-to-Noise Ratio</i>
<b>PEAQ</b>	Avaliação Perceptual da Qualidade do Áudio, do inglês, <i>Perceptual Evaluation of Audio Quality</i>
<b>RDC</b>	Controle de Taxa Versus Distorção, do inglês, <i>Rate Versus Distortion Control</i>
<b>RLC</b>	Codificação de Corrida-Comprimento, do inglês, <i>Run-length Coding</i>
<b>SNR</b>	Relação Sinal-Ruído, do inglês, <i>Signal-to-Noise Ratio</i>
<b>SPL</b>	Nível de Pressão Sonora, do inglês, <i>Sound Pressure Level</i>
<b>STFT</b>	Transformada de Fourier de Tempo Curto, do inglês, <i>Short-Time Fourier Transform</i>
<b>VBR</b>	Taxa de Bits Variável, do inglês, <i>Variable Bit Rate</i>

**WPD**      Decomposição de Pacotes Wavelet, do inglês, *Wavelet Packet Decomposition*

# Sumário

<b>1 Aspectos Introdutórios . . . . .</b>	<b>15</b>
1.1 Introdução . . . . .	15
1.2 Objetivos do Trabalho . . . . .	18
1.3 Organização do Texto . . . . .	18
<b>2 Conceitos Fundamentais . . . . .</b>	<b>19</b>
2.1 Compressão de Dados em Geral . . . . .	19
2.2 Compressão de Áudio com Perdas e Limiar de Audição . . . . .	24
2.3 Medidas de Desempenho e Modos de Compressão de Áudio . . . . .	26
2.4 Processamento de Áudio . . . . .	30
2.5 Análise em Tempo-Frequência e Transformada de Wavelet . . . . .	32
2.6 Transformada Discreta de Wavelet e de Pacotes Wavelet . . . . .	35
2.7 Regressão Linear e Aproximação Polinomial . . . . .	42
<b>3 Codificador de Áudio Proposto . . . . .</b>	<b>48</b>
3.1 Segmentação do Áudio . . . . .	49
3.2 Decomposição de Pacotes Wavelet . . . . .	51
3.3 Eliminação de Sub-bandas e Coeficientes Menos Expressivos . . . . .	54
3.4 Métodos de Codificação para os Coeficientes Mais Expressivos . . . . .	58
3.4.1 Ordenação e Aproximação Polinomial (Método A) . . . . .	58
3.4.2 Código de Entropia (Método A) . . . . .	62
3.4.3 Quantização Direta dos Coeficientes (Método B) . . . . .	68
3.5 Sistema Adaptativo de Controle de Taxa Versus Distorção . . . . .	69
3.6 Formato do <i>bitstream</i> e Processo de Decodificação . . . . .	72
<b>4 Resultados . . . . .</b>	<b>76</b>
4.1 Estudo dos Hiper Parâmetros . . . . .	76
4.2 Teste de Desempenho . . . . .	80
<b>5 Considerações Finais . . . . .</b>	<b>84</b>
5.1 Conclusões Finais . . . . .	84
5.2 Propostas de Trabalhos Futuros . . . . .	85
<b>Referências . . . . .</b>	<b>86</b>

---

# Capítulo 1

---

---

## Aspectos Introdutórios

---

### 1.1 Introdução

O crescimento no volume de dados a serem transmitidos e armazenados tem apresentado uma explosão nos últimos anos [1]. Este crescimento ocorreu devido o aumento da conectividade com a Internet e à ampla utilização de aplicativos de multimídia pela população. Segundo estudos, esse aumento era previsto e um aumento ainda maior ocorrerá no futuro [2].

Embora existam melhorias nas tecnologias de comunicação como as novas gerações de sistemas sem fio, os avanços na tecnologia das fibras ópticas e a miniaturização dos circuitos eletrônicos, a crescente demanda de serviços e o elevado custo de espaço nos meios de transmissão fazem cada vez mais necessária a utilização de sistemas de compressão, capazes de adequar os dados às limitações de um canal de transmissão ou armazenamento.

Entre os tipos de dados mais necessários de serem comprimidos estão os de áudio. Imagina-se que vários destes são transmitidos e armazenados a todo tempo devido sua presença em quase todas funções básicas de comunicação e entretenimento como, por exemplo, na comunicação por voz nos celulares, na difusão de rádio e televisão digital, ou nos diversos tipos de aplicações encontrados na Internet.

Além disso, sons codificados digitalmente com alta qualidade podem ser um problema se forem necessários de transmitir em canais sem fio, porque estes canais de transmissão compartilham sua capacidade de transmitir informações digitais entre vários sinais. Também é um problema se for desejado armazenar a gravação digital em alta qualidade em um meio com capacidade limitada, como um reproduutor de áudio portátil.

Nestes casos, alguns sistemas eficientes de codificação de áudio podem, reduzir a taxa de bits ou tamanho de arquivo dos áudios com alta qualidade em torno de 50% se não ocasionarem perda de informações do áudio (e.g, FLAC [3]), ou em volta de 90% se considerarem perdas de informação de maneira controlada (e.g, MP3 [4]). A escolha entre os dois tipos de sistemas depende da finalidade do áudio.

Assim, a compressão de áudio tem sido uma proposta atraente para fornecer áudio de alta qualidade a taxas razoáveis de transmissão e armazenamento. Deste modo,

continuamente são propostos novos métodos de compressão de áudio pelas instituições de pesquisas, grupos comerciais e comunidades de entusiastas, sendo que, muitos destas propostas viram até padrões em diferentes aplicações de serviços e dispositivos.

As diversas aplicações definem os algoritmos de compressão utilizados. Existem aquelas que exigem ou não um bom desempenho em relação a quantidade de compressão e qualidade do áudio enquanto que, outras requerem ou não que as operações de compressão sejam feitas em tempo real. Portanto, quanto maior a exigência, maior o grau de complexidade dos algoritmos, podendo ser de baixa, intermediária e alta complexidade.

Pode-se considerar que, quando se trata de compressão de áudio, importantes contribuições foram feitas pela família de codificadores do grupo MPEG (*Motion Picture Expert Group*), da ITU (*International Telecommunication Union*), da companhia Dolby Laboratories e a fundação Xiph.org quais desenvolveram e continuam desenvolvendo uma ampla gama de diferentes tipos de compressores para diferentes aplicações [4–10].

No grupo MPEG, os primeiros codificadores desenvolvidos são aqueles da linha MPEG-1 que de acordo com o nível de complexidade, foram classificados em três camadas diferentes. Na primeira camada, o codificador era bastante simples e, deste modo, adequado para sistemas que exigiam baixa complexidade como as fitas cassetes digitais da época. Na segunda camada, especificou-se um formato de compressão de intermediária complexidade e adequado para taxa de bits de aproximadamente 128 kbit/s sendo geralmente usado em aplicações de difusão de rádio digital. Por fim, na camada 3, caracterizou-se o comumente chamado de MP3 [5] qual representa um algoritmo de alta complexidade e alto desempenho muito reconhecido na transmissão de arquivos de áudio pela Internet e em reprodutores de áudio portátil até hoje.

Posteriormente, o grupo MPEG lançou o padrão MPEG-2 que continha, além de outras ferramentas, algumas extensões nos codificadores da linha MPEG-1 através da adição de sistema de som *surround* (i.e., suporte de multi-canais 5.1 onde são utilizado 5 canais de áudio mais um canal de baixas frequências *subwoofer*) e mais possibilidades de escolha para as taxas de bits possíveis na saída do codificador [4].

Na mesma linha MPEG-2 o grupo também lançou um novo codificador chamado de AAC (Advanced Audio Coding) [11] para ser o sucessor do famoso MP3 com uma técnica diferente de compressão. Este foi posteriormente melhorada na linha MPEG-4 e virou referência de sistema eficiente para difusão de televisão em alta definição (*high definition television*, HDTV) e filmes no DVD (*Digital Video Disc*) e cinema.

Assim como a ISO (*International Organization for Standardization*) formou o grupo MPEG para definir padrões para a compressão e transmissão de áudio e vídeo, a ITU também padronizou diferentes codificadores de áudio os quais foram nomeados com os



códigos G.7xx como, por exemplo, o G.719 [6]. Estes existem para vários níveis de complexidade e foram bastante focados para aplicações de conversação como a comunicação digital telefônica, videoconferência e teleconferência.

Outros codificadores importantes são aqueles pertencentes aos Dolby Laboratories que têm aprimorado seus codificadores notavelmente através das versões AC-1 (*Audio Coding 1*), AC-2, AC-3 e AC-4. O codificador Dolby AC-3 [7] é bastante reconhecido pela sua inclusão no sistema HDTV nos Estados Unidos da América na época e, o AC-4 [8] como um forte concorrente do AAC para os novos sistemas HDTV, DVDs e cinemas.

Por fim, é notado que há uma série de codificadores populares como os agrupados pela fundação Xiph.Org que tem mostrado sua grande importância por fornecer codificadores de alta qualidade sobre licenças livres de royalties. Entre seus codificadores mais conhecidos estão o Opus [10] qual é utilizado atualmente na plataforma de compartilhamento de vídeos Youtube e seu antecessor Vorbis [9] da mesma época que o MP3.

Todos estes sistemas variam em seu modo exato de operação, mas funcionam aproximadamente da mesma maneira. Se tratando de codificadores de áudio que toleram perdas de informação, estes se aproveitam das propriedades perceptuais do sistema auditivo humano para realizar compressão. Com estes sistemas, o processo de compressão produz distorções consideráveis no sinal mas que são imperceptíveis para os ouvintes humanos.

As propriedades perceptuais estão presentes no domínio do tempo e da frequência, surgindo a necessidade de transformadas ou técnicas com estas perspectivas. Os compressores utilizam então uma de diversas técnicas para obter essa visão, como por exemplo, a técnica por banco de filtros utilizada nos codificadores MPEG camada 1 [5] e 2 [4] que filtra o sinal de entrada em 32 sub-bandas.

Nos repositórios de trabalhos científicos são comumente encontrados trabalhos sobre compressão de áudio baseados em técnica de análise em tempo-frequência como, por exemplo, a chamada de Decomposição de Pacotes Wavelet (*Wavelet Packet Decomposition*, WPD) que é considerada uma variação (ou generalização) da conhecida Transformada Discreta Wavelet (*Discrete Wavelet Transform*, DWT) [12–16].

O enfoque da transformada de wavelets através da teoria da análise de multiresolução tem possibilitado diferentes técnicas de compressão as quais, muitas das vezes, são baseadas na eliminação das componentes de detalhe do sinal [17–19]. Para o caso de áudio, esse efeito se traduz na decomposição de sinais em sub-bandas que serão codificadas de acordo as propriedades perceptuais.

Grande parte dos trabalhos encontrados usando a WPD para compressão de áudio são baseados nas sofisticações dos modelos das propriedades perceptuais e das técnicas de quantização utilizadas nos codificadores mais reconhecidos (e.g, MP3 e AAC) [12–16].

Neste trabalho, o codificador proposto segue outros caminhos de codificação, com técnicas inspiradas em outros tipos de problemas, mas que contribuem ao estado da arte pela sua forma de inovação na codificação das componentes do áudio.

O codificador proposto é baseado originalmente no trabalho de Luvizotto [20] sobre modelos de sinais musicais. Nesse os coeficientes das sub-bandas WPD eram ordenados e posteriormente representados pelos parâmetros de um polinômio encontrado por regressão linear. Esse tipo de representação foi inicialmente, porém superficialmente, explorada para compressão de áudio no trabalho de Fabbri [21].

Nesta dissertação é descrita de maneira detalhada, e com vários adendos em relação ao trabalho de Fabbri, uma técnica de compressão de áudio com perdas utilizando os seguintes recursos: a WPD para a decomposição do sinal em sub-bandas; uma propriedade perceptual para a eliminação dos coeficientes menos expressivos do sinal; e, principalmente, o processo de ordenação e aproximação polinomial do trabalho de Luvizotto para a codificação dos coeficientes mais expressivos.

## 1.2 Objetivos do Trabalho

O principal objetivo desse trabalho é a apresentação detalhada de um novo esquema de compressão de áudio. Para isso, o trabalho tem os seguintes objetivos específicos:

- Propor um codificador de áudio baseado na codificação das sub-bandas *wavelet*.
- Adaptar esquemas de codificação de áudio para o domínio das sub-bandas.
- Implementar técnicas de codificação que permitam obter ganhos de compressão.
- Estudar o desempenho do codificador para diferentes configurações do codificador.
- Comparar o desempenho do codificador proposto com outros codificadores.

## 1.3 Organização do Texto

O Capítulo 2 apresenta conceitos fundamentais da compressão e das técnicas de codificação utilizadas. O Capítulo 3 introduz o esquema de compressão de áudio proposto. O Capítulo 4 apresenta uma discussão sobre o desempenho do codificador para diferentes configurações de parâmetros e apresenta os resultados do codificador proposto comparando-os com o de outros codificadores. Finalmente, o Capítulo 5 apresenta as conclusões do trabalho e propostas para possíveis trabalhos futuros.

---

## Capítulo 2

---

---

# Conceitos Fundamentais

---

Neste capítulo há uma revisão bibliográfica sobre conceitos fundamentais da codificação de áudio e sobre as várias técnicas de codificação utilizadas, a fim de brindar um conhecimento básico ao leitor no assunto deste trabalho de dissertação. Dentre os conceitos apresentados estão os de compressão de dados em geral (Seção 2.1); compressão de áudio com perdas e o limiar de audição (Seção 2.2); medidas de desempenho para compressão de áudio (Seção 2.3); processamento de áudio (Seção 2.4). Além disso, também são apresentados o conceito de análise em tempo-frequência e Transformada de Wavelet (Seção 2.5), Transformada Discreta de Wavelet e de Pacotes Wavelet (Seção 2.6) e uma revisão de regressão linear e aproximação polinomial (Seção 2.7).

### 2.1 Compressão de Dados em Geral

Para definir compressão, considere-se um sistema de comunicação simples tal como o da Fig. 2.1a, com uma fonte de informações arbitrária, um canal de transmissão (ou armazenamento) e um usuário final para usufruir das informações. Se adicionados dois blocos, conforme Fig. 2.1b, um que transforma os símbolos da fonte em outra forma, e outro que faz o processo inverso após o canal, a esses respectivamente se dá o nome de codificador e decodificador e ao processo de codificação e decodificação.

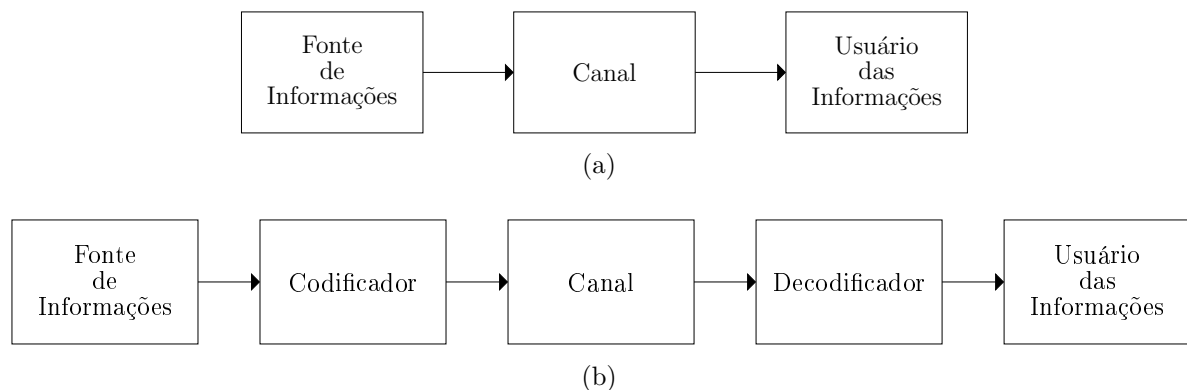


Figura 2.1 – Sistema de comunicação: (a) sem codificação (b) com codificação.

Uma das formas de codificação e decodificação são aquelas feitas para minimizar a taxa média de bits da fonte. Nesse caso, o intuito é transmitir (ou armazenar) a informação com menor custo de representação. Dentro dessa forma, existem ainda as versões que

toleram ou não perda de informação, sendo respectivamente nomeadas como compressão e compactação ou simplesmente compressão com e sem perdas.

Na teoria da informação, desenvolvida por Shannon [22] e com boa apresentação em Cover e Thomas [23], há fundamentos e limitantes sobre o processo de compressão com e sem perdas. O termo entropia definido na teoria, virou sinônimo de informação enquanto que, o termo redundância é designado aos dados que não trazem informação nova. A compressão sem perdas se baseia então em retirar a redundância da fonte, sem alterar sua entropia, enquanto que, a compressão com perdas retira a redundância e reduz a entropia de maneira controlada, conforme ilustração da Fig. 2.2.

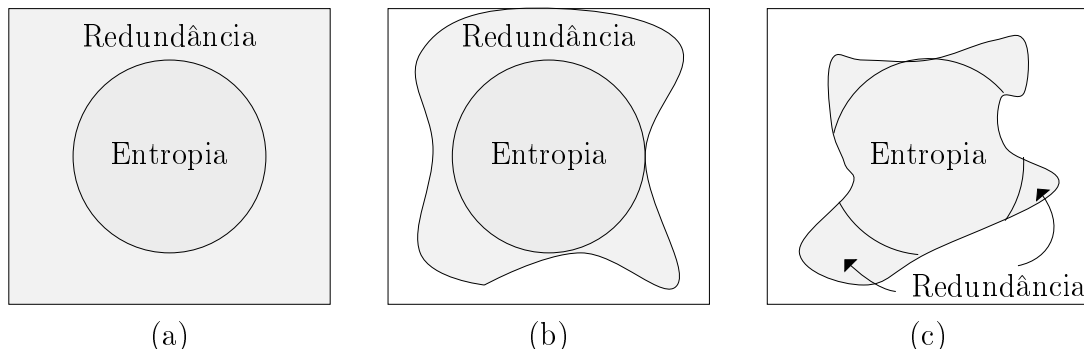


Figura 2.2 – Fonte de informação (a) na forma original, (b) após a compressão sem perdas e (c) após a compressão com perdas.

A teoria de Shannon fornece os limites de compressão, mas não a técnica para alcançá-los, o que torna a compressão uma área própria de estudos. Podemos definir essa área, como aquela que envolve identificar as estruturas existentes nos dados e realizar modelos destas para obter uma compacta representação dos dados. Em grande parte, não são os dados brutos, mas a estrutura dos dados que contém as informações. Logo, a essência da compressão envolve descobrir essas estruturas [17].

Em alguns casos, o tipo de estrutura ou redundância que existe nos dados segue uma simples regra, facilmente reconhecida, ou pode requerer uma abordagem mais abstrata para ser compreendida. Outras vezes a estrutura ou redundância nos dados se torna mais evidente quando os dados são analisados em grupos. Há também situações onde é vantagem decompor esses dados em outros espaços (e.g, transformadas matemáticas) para encontrar estruturas nas novas componentes formadas.

Na literatura sobre compressão de dados [17,18,24–27], os autores tentam classificar as diversas técnicas utilizadas na compressão afim de melhor expor o assunto. Porém, todos esses trabalhos não conseguem abranger os diferentes tipos de técnicas existentes devido a grande variabilidade dessas. Uma classificação interessante é aquela feita por Salomon [24] o qual utiliza o comprimento do dado original e o comprimento do dado

codificado para classificar as técnicas de compressão. Os comprimentos podem ser fixos ou variáveis, conforme Fig. 2.3, e uma breve apresentação dessa classificação é:

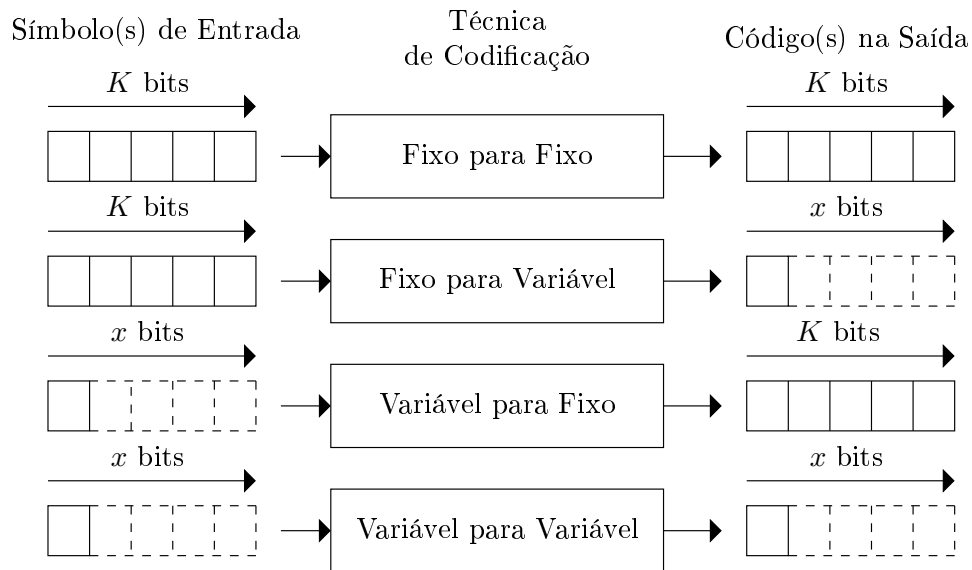


Figura 2.3 – Técnicas de codificação classificadas pelo comprimento do dado original e codificado.

- **Fixo-para-Fixo:** é a classe que codifica uma entrada de tamanho de  $n$  bits em uma saída de mesmo tamanho. Tal processo claramente não realiza compressão, este somente transforma os dados de sua forma original para uma forma onde ele se torna mais fácil comprimir. Assim, esta classe consiste de técnicas como transformadas e predições lineares, que são exemplos de técnicas usadas como primeiro passo na compressão de vários tipos de dados. Entre as propriedades desejadas nessa etapa, estão a concentração dos componentes de energia e decorrelação dos dados [27].
- **Fixo-para-Variável:** é uma classe que codifica uma entrada de tamanho  $n$  em uma saída de tamanho arbitrário. Esse tipo de código também é chamado de código de entropia e utiliza códigos curtos para os símbolos mais frequentes e códigos longos para os menos frequentes. São muito utilizados para eliminar a redundância estatística da fonte. A eficiência desses códigos está diretamente ligada a distribuição dos símbolos da fonte e sua única desvantagem talvez seja a possibilidade de propagação de erro na decodificação. Os códigos dentro dessa categoria ainda se diferenciam pelo método que computam ou estimam as probabilidades dos símbolos da fonte, conforme:
  - **Códigos Estáticos:** caso em que as probabilidades dos símbolos são conhecidas a priori, seja através do assumo de algum modelo, ou devido a análise de

um grande volume de dados. Tanto o codificador quanto decodificador conhecem essas probabilidades. A eficiência desses códigos está diretamente ligada à veracidade da tabela de probabilidade de símbolos dada.

- **Códigos de Dois-Passos:** a ideia é processar os dados duas vezes. A primeira vez para contabilizar a frequência de cada símbolo nos dados e a segunda para realizar a compressão dos dados através dos códigos de tamanho variado feitos com as frequências contadas. A eficiência desse tipo de estratégia é diminuída pela necessidade de sempre ter que enviar essas probabilidades para o decodificador juntamente com os dados.
- **Códigos Adaptativos:** nesta técnica a codificação se inicia com nenhum conhecimento das probabilidades (ou com um palpite) e essas são modificadas a cada dado lido e codificado. Geralmente quanto mais dados lidos, melhor a estimativa das probabilidades e melhor a codificação. É importante que o método de adaptação no codificador permita sua imitação no decodificador. A perda de eficiência nesse código está diretamente relacionada à velocidade de adaptação quando a fonte de informações não demonstra ser estacionária.

Em geral, os códigos de comprimento variado são dados aos símbolos em relação a suas probabilidades individuais. No entanto, em alguns casos é possível melhorar a taxa de compressão com probabilidades de pares, triplas ou  $n$ -uplas de símbolos. O resultado é uma estatística de ordem  $n$  onde os  $n$  símbolos são usados para atribuir uma probabilidade para o símbolo atual.

- **Variável-para-Fixo:** nesse tipo de codificação, os dados são divididos dentro de blocos de vários comprimentos e cada bloco é codificado com tamanho fixo. Os mais importantes membros desse grupo são a codificação de corrida e comprimento (*run-length encoding*, RLE) [28] e os vários codificadores baseados em dicionários (e.g, códigos Lempel-Ziv) [18].
- **Variável-para-Variável:** nesse, os dados são divididos dentro de blocos de comprimento variado e estes são codificados também para uma representação com comprimento variado. Esses códigos geralmente são compostos por pelo menos dois códigos, um do tipo Variável-para-Fixo e outro Fixo-para-Variável (e.g, par: RLE e código de Golomb).

Para as compressões com perdas existe ainda as técnicas que tem a finalidade de reduzir uma parte da entropia através da eliminação dos chamados dados irrelevantes. Neste caso, a relevância tem a ver com a finalidade dos dados e geralmente são impostas

pelas limitações do usuário da informação. Um bom exemplo de técnicas que visam extrair esse conteúdo é o processo de quantização e as técnicas perceptuais de compressão.

A quantização pode ser considerada a técnica mais simples de compressão com perdas pois é baseada em representar um grande conjunto (possivelmente infinito) de valores com um conjunto muito menor. Dentro dessa área, existem estratégias que buscam melhor representar os dados como as quantizações baseadas na distribuição de probabilidade da fonte (não uniforme) ou aquelas baseadas em grupos de dados (quantização vetorial) [29].

As técnicas perceptuais envolvem os fenômenos fisiológicos do usuário da informação. Por exemplo, estas técnicas se aproveitam das limitações fisiológicas visuais e auditivas do usuário para reduzir a quantidade de dados em um sinal de vídeo ou áudio. Os dados irrelevantes encontrados podem ser eliminados, ou aqueles menos expressivos podem ser quantizados de forma mais grosseira.

Por fim, pode-se dizer que os sistemas mais eficientes de compressão tentam diminuir tanto a quantidade de dados redundantes da fonte de informações quanto os dados irrelevantes. Na Fig. 2.4 ilustra-se os efeitos do uso de algumas das técnicas citadas na redução da quantidade de dados redundantes e irrelevantes.

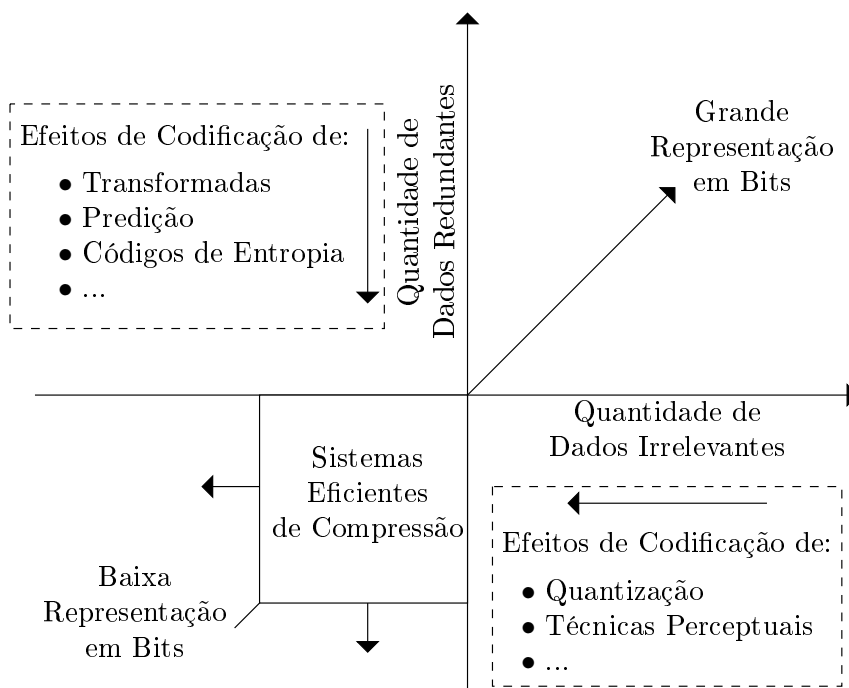


Figura 2.4 – Diagrama de desempenho dos sistemas de compressão.

Nesse trabalho é apresentado um método de compressão de áudio que almeja retirar os dados redundantes e irrelevantes do sinal de áudio através de uma codificação fixo-para-fixo usando uma transformada com propriedades desejadas (WPD), código de entropia que processa os dados em dois-passos, quantização e uma técnica perceptual que será

explicada na próxima seção.

## 2.2 Compressão de Áudio com Perdas e Limiar de Audição

Adotando-se a fonte de informações na Fig. 2.1b como um sinal de áudio e o usuário como um ouvinte, o sistema de codificação anterior se torna de compressão de áudio. Dependendo da finalidade do sinal de áudio, a compressão pode ser de áudio em geral ou também pode ser especializados em determinado tipo de som, como os compressores de voz. Na compressão de áudio em geral, os principais aspectos explorados são:

- Redundância de correlação (ou entre valores de símbolos);
- Redundância de codificação (ou estatística);
- Fatores psicoacústicos (ou sistema auditivo humano).

As redundâncias de correlação e codificação são geralmente exploradas nas compressões de áudio sem perdas, utilizando técnicas como, por exemplo, predição linear e codificação de entropia. Já os fatores psicoacústicos são explorados na compressão de áudio com perdas utilizando as chamadas técnicas perceptuais, por isso, também são chamados de codificadores perceptuais.

A rigor, a psicoacústica é uma área de estudo presente entre a acústica e a fisiologia auditiva que envolve principalmente a sensação auditiva provocada em seres humanos para diferentes características do som. Dentre os estudos desta área, existem aqueles muito explorados nos compressores de áudio para a redução de dados irrelevantes, como:

- Efeito de mascaramento de sons em frequência e no tempo;
- Bandas de percepção do sistema auditivo (bandas em escala mel);
- Sensibilidade variável do ouvido frente às componentes de frequência.

Todos esses fatores indicam componentes do sinal que podem ser quantizados ou até eliminados sem haver uma degradação considerável na qualidade do áudio para o ouvinte. Os efeitos de mascaramento e de banda críticas são melhor vistos em Moore [30] e suas aplicações na compressão de áudio em Bosi [31]. Neste trabalho é explorado a sensibilidade sonora<sup>1</sup> em relação as frequências do sons para obter compressão.

<sup>1</sup> Designa-se sonoridade à componente da percepção auditiva ligada à intensidade acústica. Essa sensação, principalmente ligada à pressão sonora, permite-nos diferenciar um som forte dum som fraco.



Dentre os estudos da sensibilidade sonora, existe uma análise chamada de Contorno de Igual Sonoridade (Equal-loudness Contour, ELC) que consiste do estudo da sensibilidade auditiva do ser humano em função do volume que este escuta para sons com diferentes valores de frequência.

A rigor, o ELC são curvas experimentais dadas por medidas de Nível de Pressão Sonora (*Sound Pressure Level*, SPL) sobre o espectro de frequência, para o qual um ouvinte percebe um volume constante quando apresentado com tons estacionários puros [30]. Tais curvas foram pioneiramente obtidas no trabalho de Fletcher-Munson [32]. Em seu experimento, foram selecionados sujeitos de teste que ouviram tons puros em várias frequências e intensidade. Para cada frequência e intensidade, o sujeito também ouvia um tom de referência a 1 kHz que tinha sua intensidade ajustada até que ele percebesse o mesmo volume que o tom de teste. Se tratando de um experimento subjetivo, foram realizadas as médias das opiniões dos ouvintes que assim produziram as curvas do ELC.

O trabalho de Fletcher-Munson de 1933 foi seguido de muitos outros, como o de Robinson-Dadson [33] de 1956, Suzuki et al. [34] de 1989 e, se encontra atualmente padronizado pela ISO 226:2003 [35] através da média dos resultados de vários desses trabalhos. Os resultados dos trabalhos pioneiros com o padronizada pela ISO é visto na Fig. 2.5 a qual apresenta uma considerável diferença entre os ELC [35].

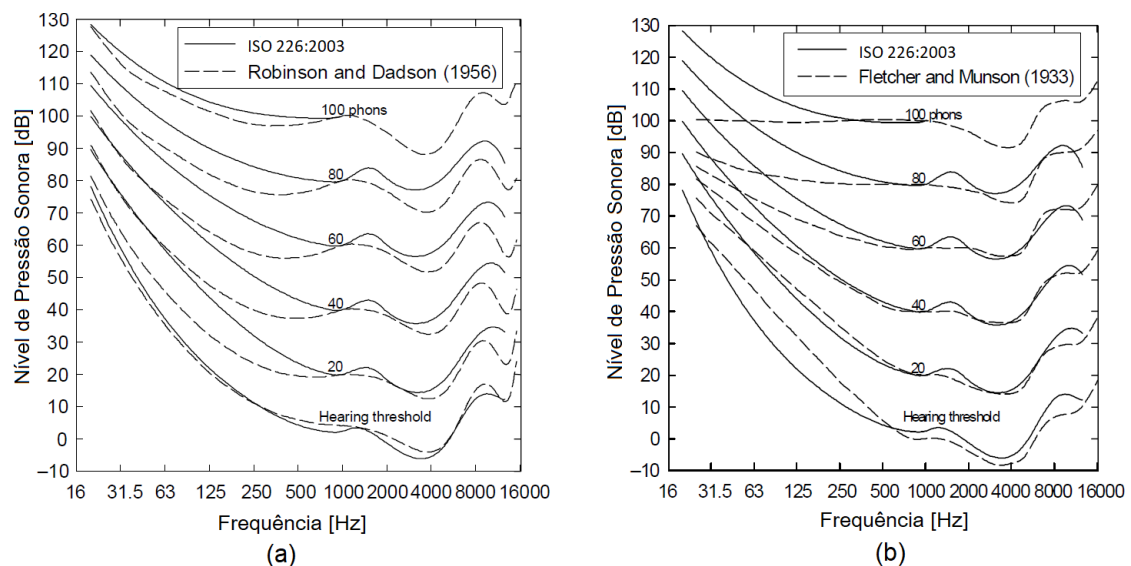


Figura 2.5 – Comparação do ELC da ISO 226:2003 em relação: a) Robinson e Dadson e b) Fletcher e Munson.

As curvas de igual sonoridade na Figura 2.5 são nomeadas por medidas de *phon* quais representam o nível de pressão sonora em decibéis para produzir a mesma sonoridade de um tom de 1 kHz naquela frequência. Sendo assim, nota-se que o ouvido humano é mais sensível para sons em torno de 1 a 5 kHz, sugerindo que um som nessa faixa é percebido mais alto que um som com o mesmo nível de pressão sonora a 100 Hz ou a 10 kHz por

exemplo. Além disso, a curva mais abaixo na figura é denominada de Limiar Auditivo (*Hearing Threshold*) e descreve o mínimo nível de pressão sonora por frequência em que o ser humano percebe algum som.

A parte do ELC que é utilizada no codificador proposto é justamente a curva do Limiar Auditivo. Essa curva é geralmente consultada na construção dos modelos psicoacústicos para vários compressores de áudio [31]. No codificador proposto ela é utilizada como peça chave na eliminação das sub-bandas e coeficientes wavelets menos expressivos, contribuindo com boa parte da compressão alcançada no codificador.

## 2.3 Medidas de Desempenho e Modos de Compressão de Áudio

O desempenho de um sistema de compressão com perdas é medido, geralmente, através do nível de compressão alcançado e da qualidade do sinal decodificado. Acredita-se que um codificador é de bom desempenho quando, para um considerável nível de compressão, a qualidade do sinal resultante é pouco afetada pelos processos de codificação. Desse modo, o propósito de um sistema de compressão de áudio é representar um sinal de áudio com a menor taxa média de bits em um canal de transmissão (ou menor espaço ocupado em um dispositivo de armazenamento) e a melhor fidelidade do áudio.

Para medir o nível de compressão, a principal medida é sem dúvida a taxa de compressão (*compression rate*,  $CR$ ) definida na equação (2.1). Essa medida é também chamada de *bpb* que significa, bit per bit, pois representa uma razão de quantos bits codificados são necessários, na média, para representar um bit do sinal original [18].

$$CR = \frac{\text{tamanho em bits do sinal codificado}}{\text{tamanho em bits do sinal original}} \quad (2.1)$$

Outra medida popularmente conhecida é a inversa da taxa de compressão chamada de fator de compressão (*compression factor*,  $CF$ ) e vista na equação (2.2). Esta medida indica de maneira direta quantas vezes a taxa média de bits (ou tamanho do arquivo) da fonte de áudio diminuiu após a passagem pelo sistema de compressão.

$$CF = \frac{\text{tamanho em bits do sinal original}}{\text{tamanho em bits do sinal codificado}} \quad (2.2)$$

Existem sistemas de compressão de áudio que produzem uma taxa de bits variável (*variable bit rate*, VBR) na saída do codificador. Tais sistemas almejam manter a qualidade do áudio codificado, diminuindo a taxa de bits em trechos mais calmos do áudio, ou seja, que precisam de menos informações, e aumentando-a em trechos mais agitados que precisem de uma maior quantidade de dados.

Outros sistemas apresentam uma taxa de bits média (*average bit rate*, ABR) na saída do codificador. Assim como os sistemas com VBR, esses também almejam manter a qualidade do áudio codificado porém, contêm um controle da codificação para uma taxa de bits média na saída. O controle simplesmente melhora a qualidade de alguns trechos em detrimento de outros para manter a taxa de bits média desejada.

Existe também os sistemas que contam com uma taxa de bits constante (*constant bit rate*, CBR) na saída do codificador. Nesses a taxa de bits é mantido, a mesma durante todo o processo de codificação e, desse modo, a qualidade vai variar dependendo do tipo do áudio sendo codificado (como por exemplo, o gênero da música) e também a cada trecho do sinal de áudio (mais agitados ou silenciosos).

Quando se trata de escolher entre os três modos, a codificação VBR é a mais recomendada. Isso porque, dependendo do controle, essa pode ter um desempenho melhor visto que os bits são atribuídos somente para manter a qualidade do áudio. O ABR é menos recomendado que o VBR pois o limite imposto em alguns trechos prejudica a qualidade do áudio em geral. Contudo, tem maior qualidade comparada com o CBR e ainda conta com uma taxa previsível ao longo do tempo. Por fim, o modo CBR é necessário em algumas aplicações como, por exemplo, na transmissão em tempo real com largura de banda fixa. A modo de exemplo, DVD, CD e Blu-rays usam VBR; já sistemas de comunicação (celular, TV e etc) usam CBR.

Agora, para avaliar a qualidade do sinal de áudio comprimido existem as medições objetivas e subjetivas. As medições subjetivas envolvem testes com grupos de ouvintes em critérios como sensibilidade e gosto, e podem levar a uma medida muito precisa da qualidade do áudio comprimido, contudo não são práticas de se realizar pela dificuldade em montar o ensaio de testes. Por outro lado, as medições objetivas envolvem desde cálculos simples baseados na distorção causada na forma de onda do sinal à também modelos complexos simulando análises perceptuais, ambas fornecendo de maneira ágil seus resultados, contudo nem sempre garantem a qualidade do áudio para o usuário final.

A relação sinal-ruído (signal-to-noise ratio, SNR) é uma das medidas objetivas mais simples e conhecidas na área da engenharia. Ela mede o grau de distorção entre o sinal original e codificado através, principalmente, da média de diferenças nas formas de onda. Para isso, é calculada a razão da energia do sinal original com a energia da diferença e é geralmente dada em unidade de decibel logarítmico (abreviado como *dB*) conforme

$$SNR (dB) = 10 \log_{10} \frac{\sigma_x^2}{\sigma_d^2} \quad (2.3)$$

sendo a energia do sinal original  $\sigma_x^2$  e a energia da distorção  $\sigma_d^2$  calculadas respectivamente

através de

$$\sigma_x^2 = \frac{1}{N} \sum_{n=1}^N x^2[n] \quad (2.4)$$

e

$$\sigma_d^2 = \frac{1}{N} \sum_{n=1}^N |x[n] - y[n]|^2 \quad (2.5)$$

onde  $x[.]$  é a sequência de amostras do sinal original,  $y[.]$  a sequência do sinal codificado e,  $n$  o índice de tempo discreto definido por números inteiros de 1 até  $N$ , sendo  $N$  o número de amostras do sinal.

Semelhante a SNR, existe ainda a relação sinal-ruído de pico (*peak-signal-to-noise-radio*, PSNR) dada pela razão entre a potência de pico do sinal original e a energia de distorção conforme

$$PSNR \text{ (dB)} = 10 \log_{10} \frac{x_{max}^2}{\sigma_d^2} \quad (2.6)$$

onde  $x_{max}$  é o valor de pico do sinal original.

Em ambos casos, quanto maior o valor medido, menor é o efeito da distorção causada pelo sistema de compressão e possivelmente melhor a qualidade do áudio. Porém, essas medidas na forma de onda podem ser pouco expressivas em relação a qualidade do áudio, porque valores baixos não necessariamente indicam resultados ruins de qualidade, visto que podem existir muitas componentes do sinal que podem estar mascaradas (Seção 2.2).

Para melhorar as formas objetivas de avaliar a qualidade de áudio, surgiram com o tempo os modelos perceptuais de avaliação que simulam os resultados de vários ouvintes em um teste subjetivo. Supõe-se que esses modelos façam o intermédio do melhor dos dois mundos, ou seja, a agilidade de resultados das medições objetivas com a veracidade dos resultados das medições subjetivas, conforme ilustra-se na Fig. 2.6.

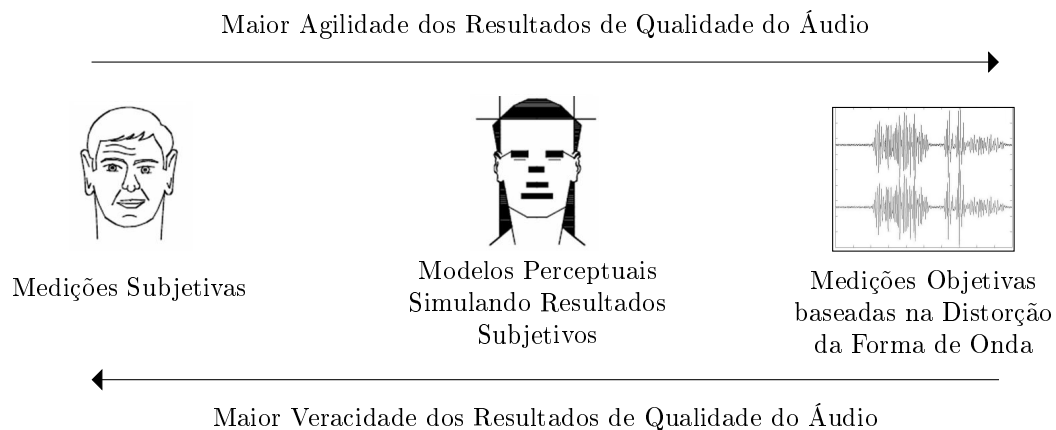


Figura 2.6 – Tipos de medição de qualidade de áudio codificado.

Os conceitos por trás das várias técnicas baseadas em modelo perceptual são todos bastante semelhantes, sendo a estrutura comum destes representada na Fig. 2.7. Primeiro, os algoritmos processam um modelo de audição para o áudio de referência (original) e o áudio testado (codificado). O resultado pode ser considerado como a representação interna dos sinais no sistema auditivo humano. Em seguida, é realizada uma comparação das duas representações que leva a uma estimativa da diferença audível entre elas. Essa informação é então processada através do referido modelo cognitivo de acordo como o cérebro de um humano faria em um teste de audição. No final, é tido uma predição da qualidade do áudio que pode ser comparado a uma pontuação de opinião média resultante de um teste subjetivo.

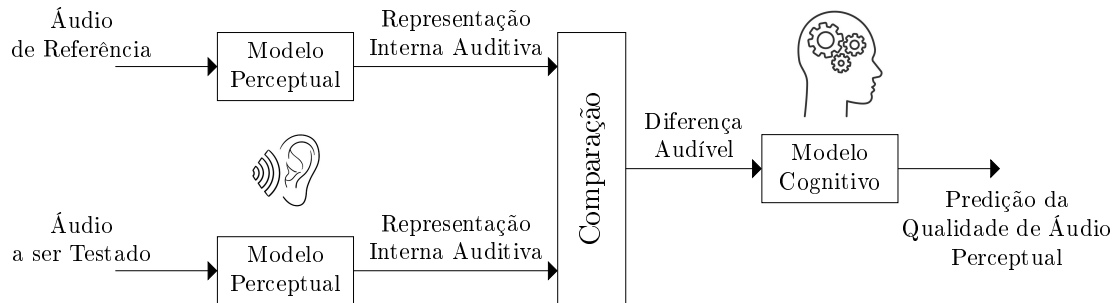


Figura 2.7 – A estrutura comum para a medição de qualidade de áudio perceptual.

Dentre as técnicas está o reconhecido PEAQ (Avaliação Perceptual da Qualidade do Áudio, do inglês, *Perceptual Evaluation of Audio Quality*) recomendado pela ITU-R (R - *Radiocommunication Sector*) em 2000/2001 através da BS.1387 [36] e a PEMO-Q [37] publicada em 2006 e inspirada no modelo de percepção auditiva PEMO de 1996 [38]. Ambas contém sistemas que simulam as propriedades perceptivas do ouvido humano e convergem várias variáveis de saída do modelo em uma única métrica.

A métrica é a mesma que os sujeitos fariam em um teste subjetivo de qualidade de áudio dado pela recomendação da ITU-R BS.1116 [39], chamada de grau de diferença objetivo (*objective difference grade*, ODG). Essa unidade corresponde a diferença entre a média dos escores dados ao áudio de referência e ao áudio de teste. Os escore individuais e a métrica ODG pode ser vista na Tabela 2.1 [36].

Tabela 2.1 – Métrica ODG do método objetivo de qualidade de áudio percebido.

Descrição das Imparidades	Escore Individual	ODG
Imperceptível	5.0	0
Perceptível, mas não incômodo	4.0	-1.0
Ligeiramente incômodo	3.0	-2.0
Incômodo	2.0	-3.0
Muito incômodo	1.0	-4.0

Através desse conjunto de medidas de desempenho é possível analisar objetivamente o compressor de áudio com perdas proposto nesse trabalho para os diferentes modos de operação VBR e CBR. Além disso, é possível comparar o desempenho do codificador proposto com o de outros compressores de áudio, utilizando como critérios as medidas apresentadas nessa seção e um conjunto representativo de amostras de áudio.

## 2.4 Processamento de Áudio

No processamento de áudio existem alguns aspectos comuns e independentes da finalidade do processamento ser para compressão ou não. Um desses aspectos é a representação digital do sinal de áudio, ou seja, a designada frequência de amostragem e resolução em bits das amostras de áudio.

Levando em consideração uma média de percepção auditiva com ouvintes jovens e sem deficiência auditiva, é visto que os sons percebidos pelo ouvido humano estão na faixa de frequência de 16 à 18 kHz [30], assim pelo teorema de Nyquist-Shannon [40, 41] é necessário pelo menos 36 kHz de frequência de amostragem para representar um áudio com todos os possíveis sons audíveis.

Na questão da resolução, o número de bits escolhido para representar as amostra de áudio determina o intervalo de níveis de som que podem ser reproduzidos (chamado de faixa dinâmica). Quanto maior essa, menor é o ruído introduzido na digitalização do áudio, sendo uma regra aproximada que a cada bit usado, são 6 dB a mais de faixa dinâmica na representação do áudio [30].

A representação digital do áudio está diretamente ligada com a finalidade dele que por sua vez corresponde a qualidade do som necessária na aplicação. Na Tabela 2.2 é visto algumas qualidades de som em função de sua representação digital [42].

Tabela 2.2 – Qualidade de som versus representação digital do sinal de áudio.

Qualidade do Som	Frequência de amostragem (kHz)	Resolução por amostra (bits)	Número de Canais	Taxa de bits (sem compressão) (kbits/s)
Telefonia	8	8	Mono	64
Rádio	11,025 ou 22,05	8 ou 16	Mono ou Estéreo	88,2 ou 352,8
CD	44,1	16	Mono ou Estéreo	705,6 ou 1411,2
DVD	de 48 até 192	até 24	até 6 canais	até 9600

Como visto, os valores usados na telefonia, de qualidade aceitável para uso comercial, são de 8 kHz para a frequência de amostragem e 8 bits/amostra de resolução. Neste caso, como a finalidade é para conversação, essa frequência de amostragem se torna adequada devido as componentes dos sons de voz irem, tipicamente, até 3,4 kHz. Por outro lado,

quando a finalidade é representar todos os tipos de som, como os instrumentais, músicas, vozes, ambientes e etc, uma frequência de amostragem como a utilizada no formato CD (*Compact Disk*) é adequada para representar toda faixa de frequência do ouvido humano.

Outro aspecto muito importante no processamento de áudio é a questão de saber se o sinal de áudio pode ser processado de uma só vez, ou se ele precisa ser dividido e processado em partes, assim como o tamanho dessas partes. Em um sistema real de processamento de áudio, essa questão geralmente gira em torno de quando uma saída é necessária pois, um sistema capaz de dividir e processar cada parte do áudio individualmente poderia produzir uma saída mais cedo do que se a saída fosse gerada de uma única vez.

Outra questão inclui o grau de poder de processamento necessário quando afrontado com gravações de áudio longas, podendo essas por exemplo, consumir toda a memória dentro do computador. Além disso, onde as características de áudio evoluem ao longo do tempo, geralmente é útil dividir o áudio em partes de tamanho igual ou regiões de característica similar [43]. Devido a essas questões o processo de segmentação de áudio é uma necessidade básica para muitos tipos de processamento de áudio, incluindo a compressão.

Determinado a necessidade pela segmentação e o tamanho das partes (ou blocos) existe ainda algumas considerações na realização desse processo [43]. A principal delas é sem dúvida o uso de sobreposição de blocos para evitar as discontinuidades provocadas pela segmentação e assim um efeito indesejado de estalos no som reconstruído. A título de exemplo, as discontinuidades formadas na segmentação são ilustradas na Fig. 2.8.

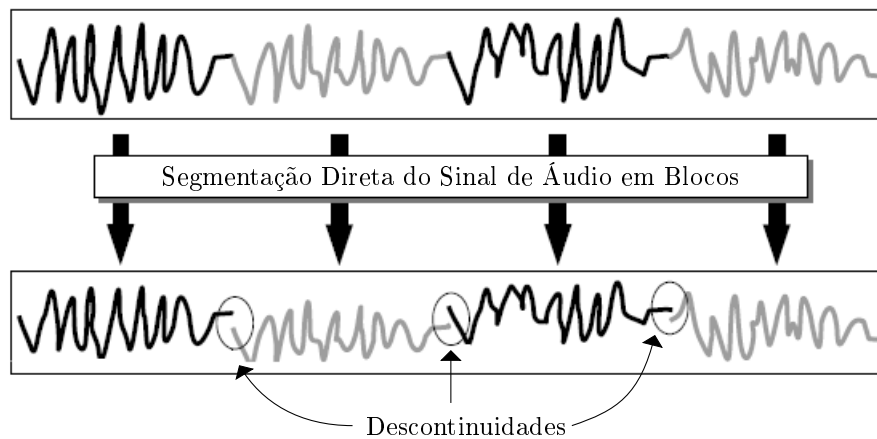


Figura 2.8 – Efeito de discontinuidades no processo de segmentação do áudio.

Desta forma, o processo de segmentação é feito com sobreposições e pode ser descrito matematicamente através da multiplicação do sinal de áudio original por uma função janela como

$$x_m[n] = x[m(N - S) + n] w[n] \quad (2.7)$$

onde  $x[\cdot]$  é o sinal de áudio original,  $w[\cdot]$  a função janela,  $m$  o índice do bloco e  $S$  o número de amostras sobrepostas.

Sobre a função janela, dependendo da finalidade do processamento de áudio pode-se haver uma escolha criteriosa dessa afim de aliviar os efeitos de borda (fenômenos de Gibbs [44]) na análise do áudio em frequência, sendo as janelas mais conhecidas a triangular, Hamming, Hann e Blackman [43]. Para o caso de não haver a necessidade de uma janela com essas propriedades, se utiliza a padrão chamada de retangular (ou topo plano).

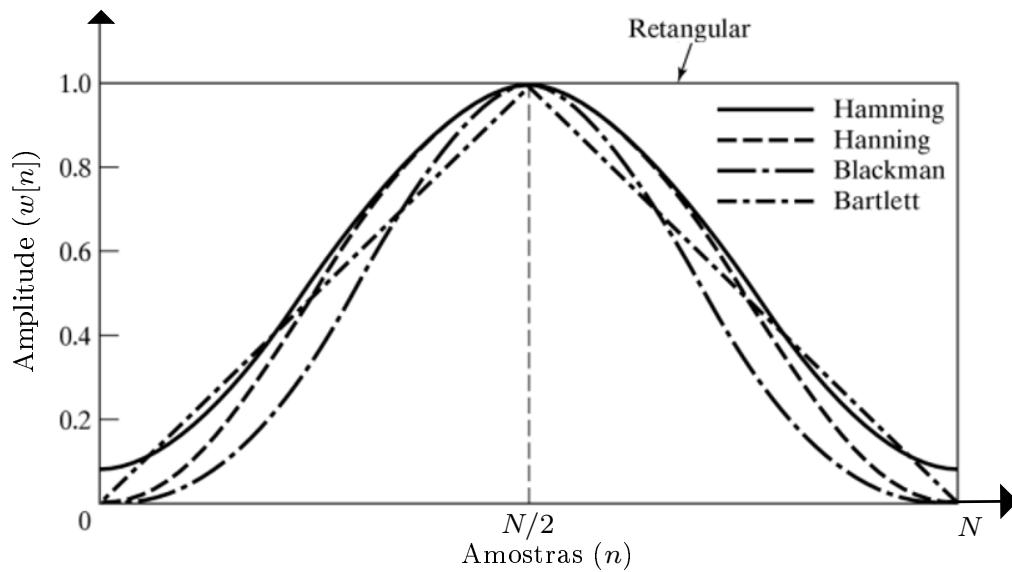


Figura 2.9 – Janelas frequentemente utilizadas no processamento de sinais.

## 2.5 Análise em Tempo-Frequência e Transformada de Wavelet

A conhecida transformada de Fourier é o método mais comum para análise de sinais no domínio da frequência. Esse tipo de análise permite por exemplo, visualizar todo conteúdo de frequência dos sons presentes em um sinal de áudio. Para isso, a transformada decompõe o sinal em espaços formados por exponenciais complexas ou, de mesmo modo, múltiplas componentes senoidais.

A definição da transformada de Fourier para a decomposição de sinais contínuos no tempo é chamada de Transformada Contínua de Fourier (*Continuous Fourier Transform*, CFT) e é calculada através de [45]

$$X(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad \text{para } t, \omega \in \mathbb{R}, \quad (2.8)$$



e sua transformada inversa dada por

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega \quad \text{para } t, \omega \in \mathbb{R}, \quad (2.9)$$

onde  $t$  e  $\omega$  são respectivamente, a variável temporal e a frequência angular. Observa-se que  $X(\omega)$  é uma função complexa que fornece a amplitude e fase do sinal contínuo  $f(t)$  no domínio da frequência através de sua amplitude para cada onda senoidal  $e^{j\omega t}$ .

Nesse ponto, um comentário conclusivo é que a transformada de Fourier é um método adequado para sinais estacionários pois, considera as componentes de frequência para todo período do sinal no tempo. Para sinais que incluem fenômenos transitórios (não-estacionários), como variações de frequência em períodos de tempo, é preferido uma decomposição com uma perspectiva de tempo-frequência.

Essa perspectiva é adequada na análise de sinais de áudio pois, como o próprio ouvido humano, ela é capaz de saber, até certo grau de certeza, quais as componentes do som que aconteceram e quando aconteceram. Um exemplo de decomposição com essa perspectiva é a Transformada de Fourier de Tempo Curto (*Short-Time Fourier Transform*, STFT).

A STFT é calculada através da transformada de Fourier de trechos do sinal (ou janelas) em diferentes instantes de tempo  $\tau$ , conforme sua definição [46]

$$S_w(\tau, \omega) = \int_{-\infty}^{\infty} f(t) \overline{g_{\tau, \omega}(t)} dt \quad \text{para } \tau \in \mathbb{R}, \quad (2.10)$$

onde  $g_{\tau, \omega}(t) = w(t - \tau) e^{j\omega t}$  e,  $w(\cdot)$  é uma função janela real, não-nula e simétrica como as apresentadas anteriormente na Fig. 2.9 com adição da comumente usada janela Gaussiana.

O tamanho da janela utilizada na STFT é relacionada à resolução do sinal no tempo-frequência, ou seja, ela determina se há boa resolução de frequência (os componentes de frequência próximos podem ser separados) ou boa resolução de tempo (o tempo em que as frequências mudam). Uma janela ampla oferece melhor resolução de frequência, mas uma resolução de tempo ruim. Uma janela mais estreita oferece boa resolução de tempo, mas baixa resolução de frequência.

Esta propriedade está relacionada ao princípio de incerteza de Heisenberg onde, não há possibilidade de ter a completa exatidão em ambos domínios, tempo e frequência, ao mesmo instante. Geralmente, esse princípio é ilustrado em um plano bidimensional tempo-frequência pelas chamadas caixas de Heisenberg quais tem seus lados esticados e contraídos em função das resoluções apresentadas, sendo o desenvolvimento dessas medidas encontradas na literatura [46]. Na Fig. 2.10 são apresentados algumas representações de domínios e transformadas pelas caixas de Heisenberg.

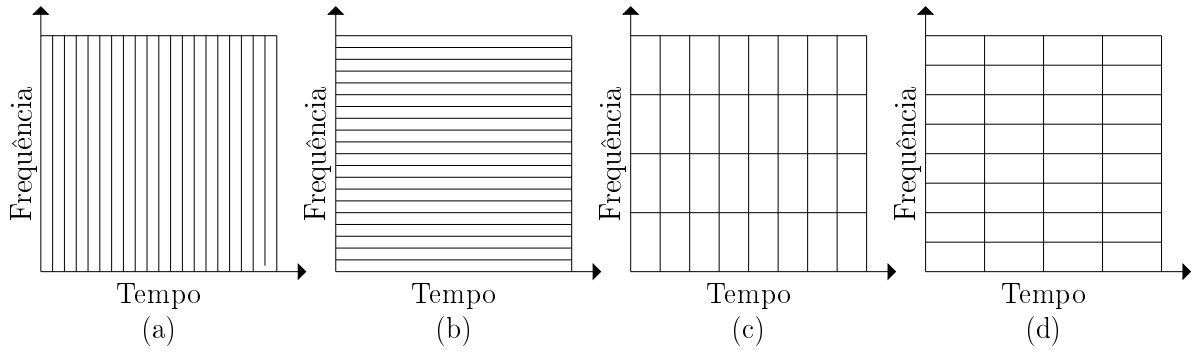


Figura 2.10 – Caixas de Heisenberg do (a) domínio do tempo, (b) transformada de Fourier, (c) STFT com uma janela estreita e (d) STFT com uma janela ampla.

Nota-se que, em um extremo, o domínio do tempo (Fig. 2.10a) fornece logicamente uma boa resolução de tempo, mas baixa resolução de frequência (na verdade, nenhuma) e, no outro extremo, o domínio de Fourier (Fig. 2.10b) fornece boa resolução de frequência, mas nenhuma resolução de tempo. Entre esses extremos, a STFT (Fig. 2.10c e Fig. 2.10d) fornece uma resolução mediana em ambos domínios tempo-frequência.

Escolhido um tamanho de janela, a STFT se limita a uma resolução fixa para todo domínio tempo-frequência do sinal. Esta é uma das razões para a criação da transformada de Wavelet e análise de multiresolução que podem fornecer uma boa resolução de tempo para eventos de alta frequência e boa resolução de frequência para eventos de baixa frequência, a combinação mais adequada para muitos sinais práticos.

A Transformada Contínua de Wavelet (*Continuous Wavelet Transform, CWT*) pode ser obtida substituindo a função  $g(\tau, \omega)$  anterior por  $\psi(a, b)$  da wavelet

$$W_\psi(a, b) = \int_{-\infty}^{\infty} f(t) \overline{\psi_{a,b}(t)} dt \quad \text{para } (a, b) \in \mathbb{R}_+ \times \mathbb{R}, \quad (2.11)$$

onde  $\psi_{a,b} = \frac{1}{\sqrt{a}} \psi(\frac{t-b}{a})$  da origem as chamadas wavelets filhas e,  $\psi(\cdot)$  é uma função chamada de wavelet mãe que tem valor médio nulo, centralização em  $t = 0$  e geralmente duração limitada. Nota-se que a variável  $a$  na expressão, altera o suporte no tempo e reescala  $\psi$  assim denominada de *parâmetro de escala ou dilatação*; e  $b$  muda a localização da função  $\psi$  no tempo assim chamado de *parâmetro de translação*. Além disso, é possível observar que pequenos valores de dilatação correspondem a altas frequências. Assim, variando  $(a, b)$ ,  $W_\psi$  pode ser calculado em todo o plano tempo-frequência.

Na Fig. 2.11a é então mostrado a representação padrão da transformada de wavelet no plano tempo-frequência. Conforme dito, esta representa um avanço na análise de sinais práticos em relação a STFT pois sua resolução muda conforme a necessidade, ou seja, contém uma resolução de tempo mais alta em frequências mais altas e uma resolução de frequência mais alta para frequências baixas. Além disso, nas Figs. 2.10b e c é mostrado

uma variação da transformada de wavelet padrão chamada de decomposição de pacotes wavelet que obtém resoluções personalizadas de acordo com a finalidade desejada.

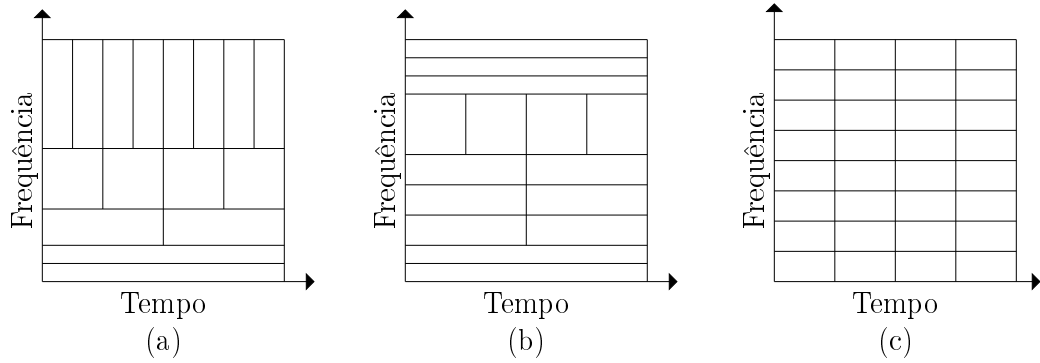


Figura 2.11 – Caixas de Heisenberg da (a) transformada de Wavelet padrão e, da decomposição de pacotes wavelets: (b) incompleta e (c) completa.

Nota-se através da Fig.2.10c que as caixas da transformada wavelet também podem ter tamanho fixo para todo plano tempo-frequência como a STFT. Estas podem até ter a mesma forma mas na prática são diferentes. Isto acontece porque as caixas representam uma situação idealizada, isto é, não há sobreposição entre caixas adjacentes e na prática isso não acontece. Em particular, a STFT fornece uma representação com alta resolução de frequência e a transformada wavelet fornece uma resolução de tempo-frequência flexível mas sofre de propriedades de localização deficientes. Em geral, a propriedade de localização de cada transformada é determinada pela escolha da função wavelet  $\psi(\cdot)$  da transformada wavelet e pela função janela  $w(\cdot)$  no caso da STFT.

## 2.6 Transformada Discreta de Wavelet e de Pacotes Wavelet

Na análise de sinais práticos, ou seja, sinais discretos de comprimento finito, são utilizadas as versões discretizadas das transformadas. Assim como a Transformada Discreta de Fourier (*Discrete Fourier Transform*, DFT) é a forma discreta para transformada de Fourier, existe a chamada Transformada Discreta de Wavelet (*Discrete Wavelet Transform*, DWT) para a definida transformada de Wavelet.

O conceito da DWT é que sob certas condições na função wavelet mãe ( $\psi(\cdot)$ ), todas as informações sobre o sinal transformado são preservadas quando a CWT é amostrada em certos subconjuntos discretos do plano de tempo-escala. Para isso, considera-se em (2.11) ao invés do par de variáveis  $(a, b)$  o par  $(a^{-j}, a^{-j}bk)$  onde são escolhidos geralmente  $a = 2$  e  $b = 1$  e, o suporte é dado por  $\{j, k \in \mathbb{Z}\}$ , conforme [46]

$$\psi_{2^{-j}, 2^{-j}k}(t) = \frac{1}{\sqrt{2^{-j}}} \psi\left(\frac{t - 2^{-j}k}{2^{-j}}\right) = 2^{j/2} \psi(2^j t - k), \quad (2.12)$$

ou seja, é escolhido apenas um conjunto de escalas e posições para fazer os cálculos, em que as escalas e as posições são escolhidas em potências de 2, formando uma grade diádica no plano tempo-escala (ou tempo-frequência).

Nessa nova forma da transformada, denota-se  $w_{j,k}$  como os coeficientes wavelets que representam os valores da DWT correspondentes à correlação

$$w_{j,k} = \langle f(t), \psi_{j,k} \rangle, \quad (2.13)$$

entre o sinal analisado  $f(t)$  (de energia finita) e a função wavelet  $\psi_{a,b}$  nos pontos específicos  $(a, b) = (2^{-j}, 2^{-j}k)$ . Dessa forma, para a reconstrução do sinal é de imediato considerado a equação (2.14) de síntese.

$$f(t) = \sum_{j,k} w_{j,k} \psi_{j,k} \quad \text{para } j, k \in \mathbb{Z}, \quad (2.14)$$

Uma condição suficiente para a perfeita reconstrução do sinal  $f(t)$  através de (2.14) é que as funções  $\psi_{j,k}$  formem bases ortonormais no espaço  $L^2(\mathbb{R})$ . A possibilidade de construção de uma função  $\psi(\cdot)$  tal que  $(\psi_{j,k})_{j,k}$  forme uma base ortonormal está diretamente ligada em algumas condições wavelet mãe [46]. Diversas bases foram construídas ao longo dos anos por pesquisadores, as quais geralmente foram dados os próprios nomes dos pesquisadores, uma destas é a dita base de suporte mínimo chamada de Daubechies [47].

Observa-se que (2.14) é composta por uma soma duplamente infinita sobre o índice de tempo  $k$  e o índice de escala  $j$ . No entanto, esta soma pode ser tornada finita com pouco ou nenhum erro. A soma finita para  $k$  é clara para o caso de funções wavelet finitamente suportadas e são consideradas aproximações para as funções infinitamente suportadas em razão da energia principal estar concentrada dentro de um certo intervalo. Agora, para entender por que a soma finita sobre  $j$  é válida, com alguma aproximação, apresenta-se o conceito de Análise de Multirresolução (*Multiresolution Analysis*, MRA) [46].

A MRA é definida como uma família de subespaços fechados  $V_j \subset L^2(\mathbb{R})$  com as seguintes propriedades:

$$V_j \subset V_{j+1}, \quad \forall j \in \mathbb{Z} \quad (2.15a)$$

$$f(t) \in V_j \Rightarrow f(2t) \in V_{j+1}, \quad \forall j \in \mathbb{Z} \quad (2.15b)$$

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R}) \quad (2.15c)$$

$$\bigcap_{j \in \mathbb{Z}} V_j = 0 \quad (2.15d)$$

$$\exists \varphi \in V_0 \text{ tal que } (\varphi(t - k))_{k \in \mathbb{Z}} \text{ é uma base ortogonal para } V_0 \quad (2.15e)$$

Na propriedade (2.15e) se introduz a existência de uma função auxiliar  $\varphi(\cdot)$  chamada de função de escala, de aproximação ou de wavelet pai. Nota-se que  $V_0$  nessa propriedade

equivale ao  $\text{span}(\varphi_k : k \in \mathbb{Z})$  onde  $\varphi_k(t) = \varphi(t-k)$ . Desse modo, a partir desta propriedade e a de dilatação (2.15b) alguém pode construir os chamados subespaços de aproximação

$$V_j = \text{span}(\varphi_{j,k} : k \in \mathbb{Z}), \text{ onde } \varphi_{j,k}(t) = 2^{j/2} \varphi(2^j t - k), \quad (2.16)$$

onde  $(\varphi_{j,k})_{k \in \mathbb{Z}}$  é uma base ortonormal para  $V_j$ .

Neste ponto, para tornar possível a soma finita na reconstrução, são definidos os chamados subespaços de detalhe  $W_j$  através da função wavelet em (2.12), conforme (2.17). Estes novos subespaços constituirão as diferenças ortogonais da sequência de subespaços  $V_j$ , ou seja,  $W_j$  é o complemento ortogonal de  $V_j$  dentro do subespaço  $V_{j+1}$ .

$$W_j = \text{span}(\psi_{j,k} : k \in \mathbb{Z}), \text{ onde } \psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k), \quad (2.17)$$

Com essas relações de ortogonalidade impostas ( $V_{j+1} = V_j \oplus W_j$ ) e as propriedades da MRA, pode-se derivar os subespaços de uma função  $f \in V_J$  com  $J$  representando a resolução atual da função (i.e, para sinais práticos de banda limitada, a própria frequência de amostragem) através de

$$\begin{aligned} f \in V_J &= V_{J-1} \oplus W_{J-1} \\ &= V_{J-2} \oplus W_{J-2} \oplus W_{J-1} \\ &= \dots \\ &= V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \dots \oplus W_{J-1}, \end{aligned} \quad (2.18)$$

onde  $j_0$  é a resolução mais baixa escolhida para ser incluída na análise.

Desde que  $\{\phi_{j,k}\}_{k \in \mathbb{Z}}$  e  $\{\psi_{j,k}\}_{k \in \mathbb{Z}}$  são bases ortogonais de  $V_j$  e  $W_j$ , todas as funções nesses subespaços podem ser representadas através de

$$\mu_j(t) = \sum_k a_{j,k} \varphi_{j,k}(t) \quad (2.19a)$$

$$\nu_j(t) = \sum_k d_{j,k} \psi_{j,k}(t) \quad (2.19b)$$

sendo  $a_{j,k}$  e  $d_{j,k}$  os chamados coeficientes de aproximação e detalhe, os quais são calculados através dos produtos internos  $a_{j,k} = \langle f, \phi_{j,k} \rangle$  e  $d_{j,k} = \langle f, \psi_{j,k} \rangle$ , respectivamente.

Usando (2.19) em (2.18) para qualquer sinal ou função  $f \in V_J$  pode se dar uma representação em funções básicas dos subespaços correspondentes como

$$\begin{aligned} f &= \nu_{J-1} + \nu_{J-2} + \dots + \nu_{j_0} + \mu_{j_0} \\ &= \sum_{j=j_0}^J \sum_k d_{j,k} \psi_{j,k} + \sum_k a_{j_0,k} \varphi_{j_0,k}, \quad \forall f \in V_J \end{aligned} \quad (2.20)$$

As relações de ortogonalidade e MRA também implicam na existência das sequências  $h_k$  e  $g_k$  que satisfazem as identidades

$$h_k = \langle \varphi_{0,0}, \varphi_{-1,k} \rangle \text{ de forma que } \varphi(t) = 2 \sum_{k \in \mathbb{Z}} h_k \varphi(2t - k), \text{ e} \quad (2.21a)$$

$$g_k = \langle \psi_{0,0}, \varphi_{-1,k} \rangle \text{ de forma que } \psi(t) = 2 \sum_{k \in \mathbb{Z}} g_k \varphi(2t - k). \quad (2.21b)$$

sendo  $\varphi(t)$  e  $\psi(t)$  chamadas de equação de escala e equação wavelet respectivamente e, a identidade (2.21a) é a equação de refinamento para wavelet pai  $\varphi$ .

Ambas expressões (2.21a) e (2.21b) formam a base para a Transformada Rápida de Wavelet (*Fast Wavelet Transform*, FWT) que assim como a Transformada Rápida de Fourier (*Fast Fourier Transform*, FFT) para a DFT, requer menos operações computacionais para sua realização, sendo que, para um sinal de entrada de comprimento  $N$ , a complexidade dada por  $O(N)$  é até menor do que a FFT com  $O(N \ln N)$ .

A FWT representa um teorema desenvolvido por Mallat [46] (por isso também é chamada de Algoritmo de Mallat) que prova a existência das expressões

$$a_{j+1}[p] = \sum_{k=-\infty}^{\infty} a_j[k] h[k - 2p] = a_j * \bar{h}[2p], \quad (2.22a)$$

$$d_{j+1}[p] = \sum_{k=-\infty}^{\infty} a_j[k] g[k - 2p] = a_j * \bar{g}[2p] \quad (2.22b)$$

para o processo de decomposição e,

$$\begin{aligned} a_j[p] &= \sum_{k=-\infty}^{\infty} a_{j+1}[k] h[p - 2k] + \sum_{k=-\infty}^{\infty} d_{j+1}[k] g[p - 2k] \\ &= \check{a}_{j+1} * h[p] + \check{d}_{j+1} * g[p] \end{aligned} \quad (2.23)$$

para o processo de reconstrução, onde  $*$  é a operação de convolução e denota-se  $\check{r}$  como

$$\check{x}[n] = \begin{cases} r[p], & \text{se } n = 2p \\ 0, & \text{se } n = 2p + 1 \end{cases} \quad (2.24)$$

As expressões (2.22a) e (2.22b) indicam que o processo de decomposição de um sinal  $x[n]$  para um nível de transformada ( $j_0 = 1$ ) pode ser feito através de filtragem e dizimação conforme Fig. 2.12a, enquanto que, a expressão (2.23) indica que a reconstrução pode ser feita através de interpolação e filtragem conforme Fig. 2.12b.

Observa-se a partir de (2.22) e (2.23) que os filtros de decomposição  $h_d[n]$  e  $g_d[n]$  são as formas dizimadas e conjugadas dos filtros  $h_r[n]$  e  $g_r[n]$ . Estes filtros funcionam respectivamente como filtros passa-baixas e passa-altas e as operações de interpolação e dizimação são justificadas pela possível mudança de frequência de amostragem.

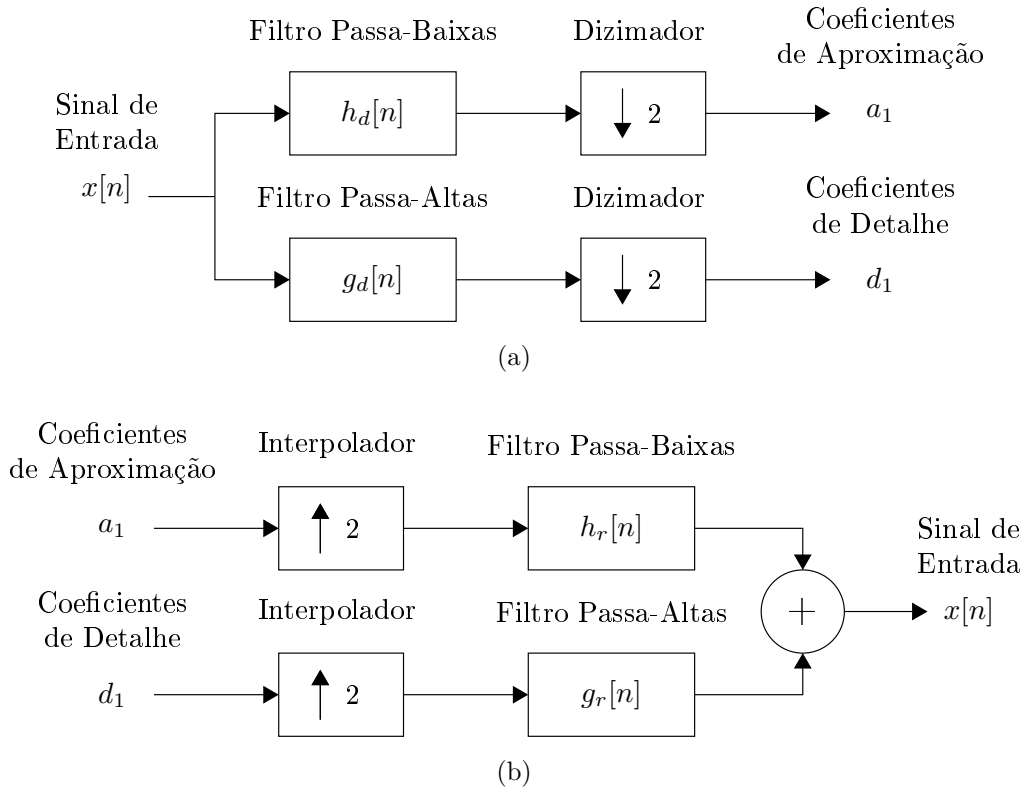


Figura 2.12 – Esquema de: (a) decomposição e (b) reconstrução de um sinal pela FWT.

Por exemplo, no processo de decomposição metade das amostras são descartadas na subamostragem devido que metade da frequência dos sinais são removidas pelos filtros, isso de acordo com o teorema de Nyquist-Shannon [40,41]. Por outro lado, a interpolação feita na reconstrução, aumenta o número de amostras do sinal para a nova frequência de amostragem maior enquanto que, os filtros apagam as imagens adicionais formadas por esse processo.

As respostas impulsivas destes filtros deverão satisfazer as condições de perfeita reconstrução e de ortogonalidade impostas na formulação da transformada. Nota-se em (2.21a) e (2.21b) que essas condições estão diretamente relacionadas com a família de wavelets utilizada (i.e, wavelet pai e mãe). Um exemplo de filtros que satisfazem as condições são aqueles da família de wavelets Daubechies, Haar, Morlet, Coiflets e Biortogonais [46].

Cada uma destas famílias de wavelet tem características próprias e dentro de cada uma ainda existem parâmetros que também variam suas propriedades. Um desses parâmetros é o chamado número de momentos nulos (*vanish moments*) das funções  $\psi(\cdot)$  e  $\phi(\cdot)$ . Este parâmetro está diretamente relacionado à seletividade dos filtros formados por essas funções, sendo que maiores valores implicam em maior custo computacional, já que o número de coeficientes dos filtros de decomposição e reconstrução também são incrementados.

Na Fig. 2.13 são ilustrados as respostas em amplitude dos filtros de decomposição

onde  $F_{mx}$  representa a metade da frequência de amostragem do sinal  $x[n]$ . Além disso, na ilustração é mostrada a seletividade dos filtros, nesse caso, o tracejado mostra uma seletividade menor a linha contínua e pode ter sido provocado por um número de momentos nulos menor.

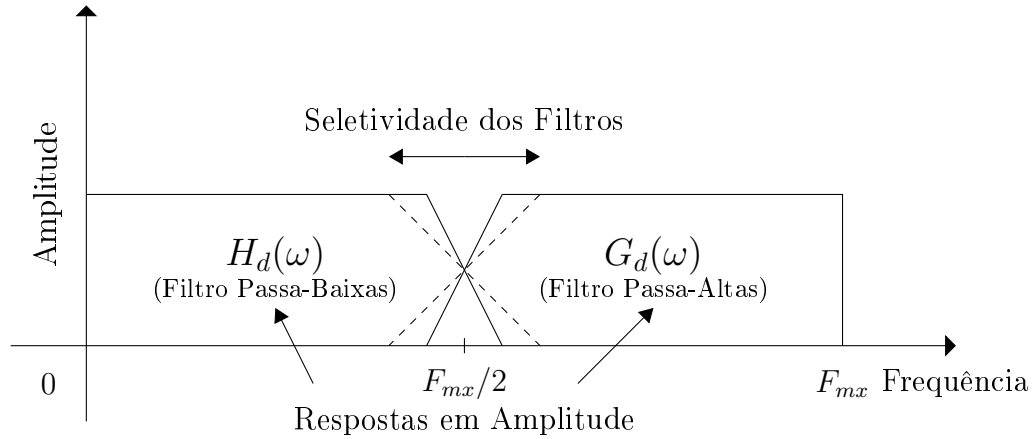


Figura 2.13 – Ilustração das respostas em amplitude e a seletividade dos filtros.

Para maiores níveis de decomposição, o processo de decomposição é realizado iterativamente decompondo-se a sequência de saída dos filtros passa-baixas até se chegar a um nível de decomposição ( $j_0$ ) desejado conforme Fig. 2.14. À vista disso, as respostas em amplitude resultante é vista na Fig. 2.15.

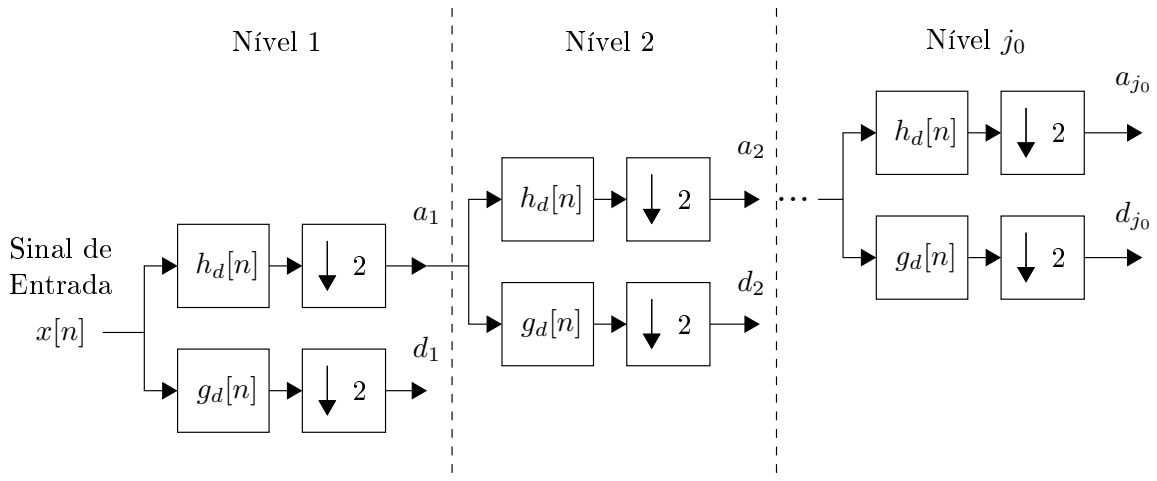


Figura 2.14 – Esquema de decomposição para multi-níveis de transformada.

Nota-se que o processo é realizado decompondo-se os coeficientes de aproximação ( $a_j$ ) até se chegar a um nível de decomposição desejado. Este processo de decomposição é muitas vezes representado através de uma estrutura de árvore conforme Fig. 2.16a na qual os nós à esquerda de cada nível representam os coeficientes de aproximação e os nós da direita os coeficientes de detalhe e cada par de ramos da árvore simboliza o processo de decomposição da Fig. 2.12.



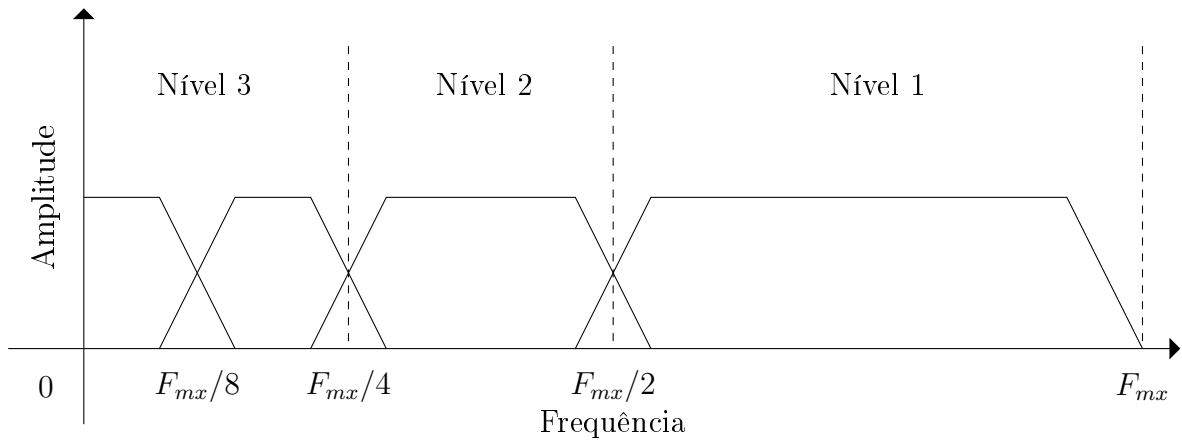


Figura 2.15 – Ilustração das respostas em amplitude para multi-níveis de transformada.

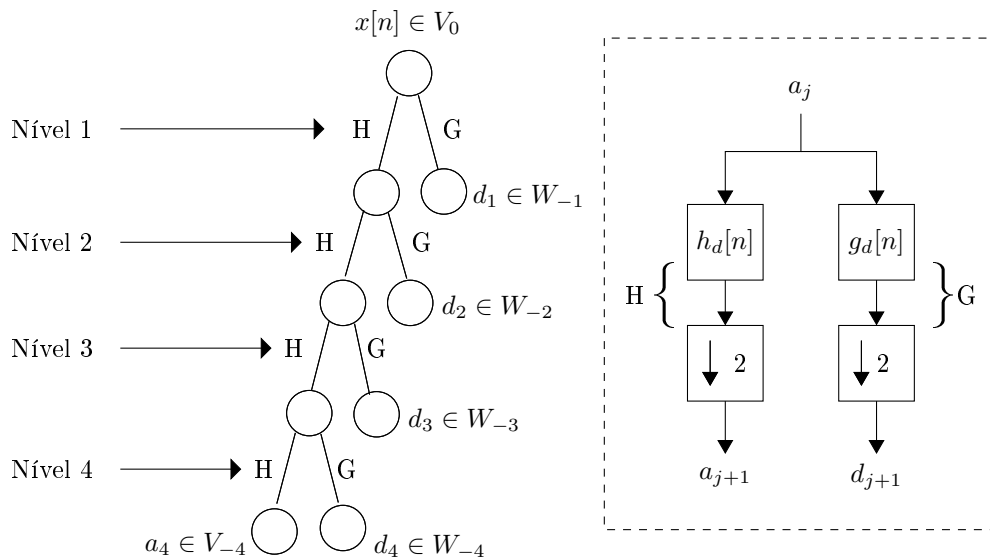


Figura 2.16 – Estrutura da árvore de decomposição DWT.

Por fim, através de uma extensão natural da DWT chamada de WPD se permite que a árvore de decomposição também se divida do lado direito em cada nó, ou seja, se permite também a decomposição dos coeficientes de detalhe ao invés de somente os de aproximação. Para isso, é possível demonstrar que a partir da decomposição do subespaço  $W_j$  pode-se gerar iterativamente diversos subespaços e formatos de decomposição de acordo com o tipo de aplicação na qual será utilizada a árvore [46].

Na Fig. 2.17 mostra-se dois formatos de WPD com 4 níveis de profundidade. Na Fig. 2.17a apresenta-se uma árvore personalizada para alguma aplicação e na Fig. 2.17b uma árvore com todos os subespaços de todos os níveis decompostos, denominada de árvore binária cheia (*full binary tree*). A representação destas decomposições no plano tempo-frequência são apresentadas nas Figura 2.10b e c da seção anterior.

A partir desta representação por árvore e o resultado das respostas em amplitude

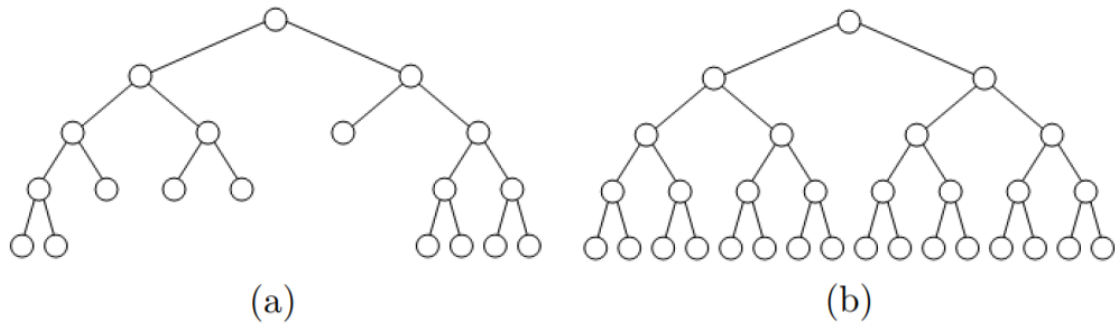


Figura 2.17 – Estrutura da árvore WPD do tipo: (a) incompleto (ou personalizada) e (b) árvore balanceada (ou árvore binária cheia).

dos filtros de cada ramo vistos na Fig. 2.15 é visto que a técnica WPD cria uma árvore onde cada folha representa uma sub-banda do sinal de entrada, assim também chamada de Árvore de Sub-bandas (*Subband Tree*). Esta técnica será utilizada neste trabalho para codificação de áudio.

## 2.7 Regressão Linear e Aproximação Polinomial

Em estatística, a regressão é o estudo da dependência, ou mais especificamente, a análise que permite explorar e inferir a relação de uma variável dependente (variável de resposta) com variáveis independentes específicas (variáveis explicatórias). Essa, assim como a maioria das análises estatísticas, tem como objetivo resumir os dados observados da maneira mais simples, útil e elegante possível [48].

Para elucidar a teoria por trás da regressão, constrói-se um experimento com uma variável explicatória chamada de  $X$  e uma variável de resposta  $Y$ . O objetivo da regressão é então entender como a distribuição de valores de  $Y$  muda quando  $X$  é variado em sua faixa de valores possíveis.

Nesse caso, um aspecto importante da distribuição de  $Y$  é saber o seu valor esperado dado uma amostra de  $X = x$ , qual denota-se por [48]

$$E(Y|X = x) = f(x) \quad (2.25)$$

onde  $E(\cdot)$  é o valor esperado do argumento. Outra característica da distribuição da variável de resposta é a função de variância, definida como

$$Var(Y|X = x) = \sigma^2(x) \quad (2.26)$$

onde  $Var(\cdot)$  é o cálculo da variância do argumento e  $\sigma^2$ , supõe-se ser uma constante, maior do que zero e desconhecida.

Abordando o uso da metodologia de regressão chamada de regressão linear, o modelo mais simples desta, tem seu valor esperado igual a

$$E(Y|X = x) = \beta_0 + \beta_1 x \quad (2.27)$$

onde o parâmetro  $\beta_0$  é o intercepto (valor da equação quando  $x$  é igual a zero) ou também chamado de termo constante e,  $\beta_1$  é a inclinação (taxa de variação da resposta para uma mudança de unidade na variável  $X$ ).

Dado que a variância do modelo é suposto ser maior do que zero, existe uma diferença entre os dados observados e os valores esperados da resposta. Essa pode ser definida para o  $i$ -ésimo valor como

$$e_i = y_i - E(Y|X = x_i) = y_i - (\beta_0 + \beta_1 x_i) \quad (2.28)$$

sendo  $y_i$  o  $i$ -ésimo dado observado e  $e_i$ , a  $i$ -ésima diferença chamada de erro estatístico. Nesse modelo, o erro estatístico também é uma variável aleatória e, supõe-se que todos os erros tem  $E(e_i|x_i)$  igual a zero, ou seja, o  $i$ -ésimo erro e o  $i$ -ésimo valor da variável independente são descorrelacionados.

Os parâmetros que caracterizam o modelo de regressão ( $\beta_0$  e  $\beta_1$  em 2.27) são desconhecidos e são estimados usando os dados do experimento. Podem ser tomados cuidados na notação da equação do modelo como

$$\hat{y}_i = \hat{E}(Y|X = x) = \hat{\beta}_0 + \hat{\beta}_1 x \quad (2.29)$$

e de mesma forma, os resíduos  $e_i$  para o  $i$ -ésimo caso são denotados como em (2.30).

$$\hat{e}_i = y_i - \hat{E}(Y|X = x_i) = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) \quad (2.30)$$

A estimativa dos parâmetros  $\beta_0$  e  $\beta_1$  pode ser feita seguindo o Método dos Mínimos Quadrados Ordinários (MMQ), no qual os parâmetros são escolhidos a fim de minimizar a função da soma dos quadrados do resíduo em (2.31).

$$\begin{aligned} g(\beta_0, \beta_1) &= \sum_{i=1}^n (y_i - E[Y|X = x_i])^2 \\ &= \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \end{aligned} \quad (2.31)$$

Tal função também é chamada de função custo e um método para encontrar o mínimo dela é pela diferenciação da função com respeito a  $\beta_0$  e  $\beta_1$  e igualando a zero, de

acordo com

$$\frac{\partial g(\beta_0, \beta_1)}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0 \quad (2.32a)$$

$$\frac{\partial g(\beta_0, \beta_1)}{\partial \beta_1} = -2 \sum_{i=1}^n x_i (y_i - \beta_0 - \beta_1 x_i) = 0 \quad (2.32b)$$

rearranjando os termos, tem-se que

$$\beta_0 n + \beta_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \quad (2.33a)$$

$$\beta_0 \sum_{i=1}^n x_i + \beta_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i. \quad (2.33b)$$

Esse sistema de equações 2.33 é chamado de equações normais. As equações normais dependem somente dos dados através da estatística suficiente dos termos  $\sum x_i$ ,  $\sum y_i$ ,  $\sum x_i^2$  e  $\sum x_i y_i$ . Contudo, é possível usar as fórmulas (2.34) que são equivalentes e numericamente mais estáveis. Nesse caso, as estatísticas suficientes são dadas pelos termos  $\bar{x}, \bar{y}$ , SXX e SXY, onde SXX significa a soma dos quadrados e SXY a soma dos produtos cruzados.

$$SXX = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n (x_i^2 - n\bar{x}^2) \quad (2.34a)$$

$$SXY = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n (x_i y_i - n\bar{x}\bar{y}) \quad (2.34b)$$

Resolvendo (2.33) através dos termos (2.34) tem-se que

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad \text{e} \quad \hat{\beta}_1 = \frac{SXY}{SXX}. \quad (2.35)$$

Desse modo, todos os cálculos de mínimos quadrados para regressão linear dependem apenas das médias, somas de quadrados e somas dos produtos cruzados dos dados do experimento.

A forma generalizada do modelo de regressão linear, é chamada de regressão linear múltipla. Na generalização supõe-se  $p$  variáveis explicativas  $X_1, X_2, \dots, X_{p-1}, X_p$  com a qual prever a resposta, assim obtém-se uma função média tal como

$$E(Y|X = \mathbf{x}) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p \quad (2.36)$$

onde  $\mathbf{x}$  significa que estamos condicionando a valores específicos todos os termos do lado direito da equação. Essa função pode ainda ser reescrita sem o intercepto como

$$E(Y|X = \mathbf{x}) = \beta_0 X_0 + \beta_1 X_1 + \dots + \beta_p X_p \quad (2.37)$$

e quando  $p = 1$  e  $X_0 = 1$  se tem o modelo de regressão linear simples dito anteriormente.

Pode-se também escrever o modelo de regressão linear múltiplo usando notação de matrizes e vetores para  $n$  dados observados conforme

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \quad (2.38)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{0,1} & x_{1,1} & x_{2,1} & \dots & x_{p,1} \\ x_{0,2} & x_{1,2} & x_{2,2} & \dots & x_{p,2} \\ x_{0,3} & x_{1,3} & x_{2,3} & \dots & x_{p,3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{0,n} & x_{1,n} & x_{2,n} & \dots & x_{p,n} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \quad (2.39)$$

onde  $\mathbf{Y}$  é um vetor ( $n \times 1$ ) e  $\mathbf{X}$  uma matriz ( $n \times (p+1)$ ). Também defini-se  $\boldsymbol{\beta}$  como sendo um vetor ( $p+1$ ) de parâmetros e  $\mathbf{e}$  o vetor ( $n \times 1$ ) de erros estatísticos,

A  $i$ -ésima linha de  $\mathbf{X}$  pode ser definida pelo símbolo  $\mathbf{x}_i$ , que é um vetor de tamanho  $(p+1)$  e tem função média avaliada com

$$E(Y|X = \mathbf{x}_i) = \mathbf{x}_i' \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} \quad (2.40)$$

e a  $i$ -ésima resposta com  $y_i = \mathbf{x}_i' \boldsymbol{\beta} + e_i$ .

Para o modelo de regressão linear múltipla a função custo da estimativa de  $\hat{\boldsymbol{\beta}}$  pelo MMQ é dada por

$$g(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})' - (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \quad (2.41)$$

e com desenvolvimento análogo ao desenvolvimento para o modelo de regressão linear simples, se tem os parâmetros estimados por

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{Y} \quad (2.42)$$

desde que o inverso  $(\mathbf{X}'\mathbf{X})^{-1}$  exista. O estimador  $\hat{\boldsymbol{\beta}}$  depende apenas das estatísticas suficientes  $\mathbf{X}'\mathbf{X}$  e  $\mathbf{X}'\mathbf{Y}$ , que são matrizes de soma de quadrados e produtos cruzados.

A suposição de que a função de variância em (2.26) é constante para todos os valores de  $x$  pode ainda ser relaxada. No caso mais simples, ao invés de assumir que os erros são constantes, assume-se que

$$Var(Y|X = x_i) = Var(e_i) = \sigma^2/w_i \quad (2.43)$$

onde  $w_1, \dots, w_n$  são números positivos conhecidos. A função de variância ainda é caracterizada por apenas um número positivo desconhecido  $\sigma^2$ , mas as variações podem ser diferentes para cada caso.

Em termos matriciais, seja  $\mathbf{W}$  uma matriz diagonal  $n \times n$  com o  $w_i$  na diagonal, tem-se que a função variação pode ser representada como em (2.44).

$$\text{Var}(\mathbf{e}) = \sigma^2 \mathbf{W}^{-1} \quad (2.44)$$

Além disso, com uma pequena alteração na função custo anterior (2.41), o estimador  $\hat{\boldsymbol{\beta}}$  agora é escolhido para minimizar a função (2.45).

$$g(\boldsymbol{\beta}) = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})' \mathbf{W} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = \sum w_i (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2 \quad (2.45)$$

Nesse caso, a estimação dos parâmetros é feita pela chamada de Método dos Mínimos Quadrados Ponderados (MMQP) e tem solução por

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}' \mathbf{W} \mathbf{X})^{-1} \mathbf{X}' \mathbf{W} \mathbf{Y} \quad (2.46)$$

qual reconhece que alguns dos erros são mais variáveis do que outros.

Por fim, os problemas de regressão têm diversos tipos de variáveis explicativas em potencial. Algumas delas são:

- **Normal:** onde as variáveis explicativas são compostas diretamente pelas medidas do experimento, uma medida diferente para cada variável.
- **Transformação:** o mesmo que as normais contudo as medições originais são transformadas nas variáveis explicativas (e.g. transformação para escala logarítmica).
- **Polinomial:** nesse caso as variáveis explicativas são baseadas em uma mesma medida com graus diferentes, as funções médias resultante são então polinômios.
- **Combinação:** as variáveis explicativas são o resultado de combinações das medidas do experimento, tal combinação muitas vezes é útil.

Nesse trabalho as variáveis explicativas são consideradas na forma de termos polinomiais, assim se acomodam na regressão linear múltipla como

$$E(Y|X = x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_{(p-1)} x^{(p-1)} + \beta_p x^p. \quad (2.47)$$

Desse modo, a regressão linear é uma curva polinomial estimada a partir dos dados no espaço bidimensional. Enquanto que, os resíduos do modelo são equivalentes as distâncias verticais entre os valores observados (ou reais) e aqueles da curva estimada.

Esse tipo de regressão pode ser utilizada para compressão através da construção de modelos para as estruturas nos dados. Para exemplificar, considere-se um conjunto de

dados como os da Fig. 2.18a. Utilizando a regressão é possível construir um modelo para esse conjunto de dados através, por exemplo, de uma única equação, como mostrado na Fig. 2.18b. Dessa forma, obtém-se compressão apenas por transmitir (ou armazenar) os parâmetros desta equação ao invés dos dados, podendo também transmitir a sequência de erro para os casos de compressão sem perdas.

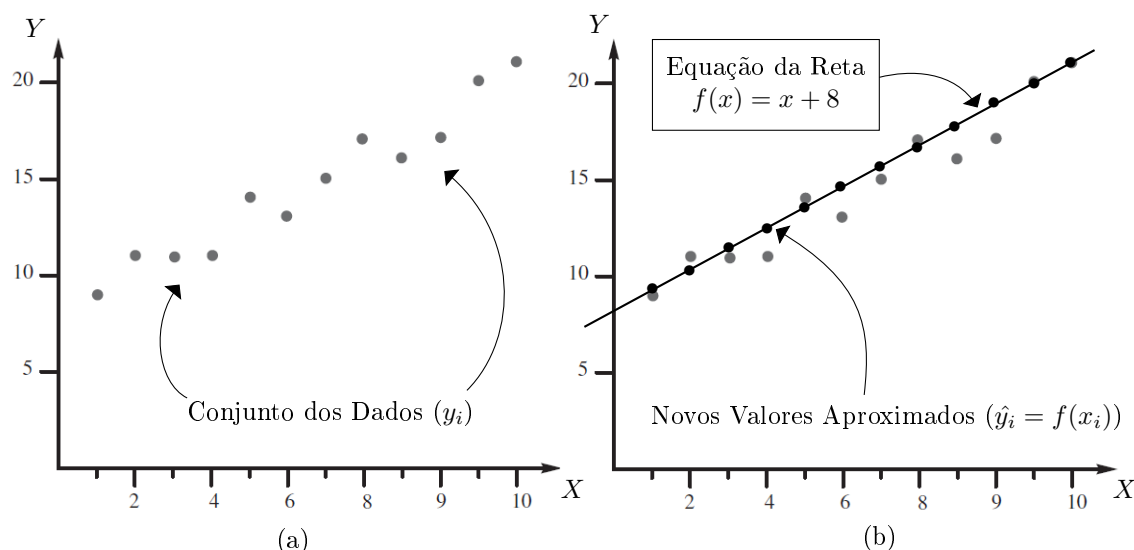


Figura 2.18 – Exemplo de regressão: (a) dados originais e (b) dados aproximados com regressão.

## Capítulo 3

# Codificador de Áudio Proposto

No presente capítulo descreve-se o codificador de áudio proposto neste trabalho. O processo de compressão é baseado na eliminação dos coeficientes menos expressivos das sub-bandas wavelet, assim como, na codificação dos coeficientes preservados usando técnicas como a aproximação polinomial e codificação de entropia. Os conceitos fundamentais dessas técnicas foram apresentados na revisão bibliográfica do Capítulo 2.

Todas as operações do codificador podem ser resumidas no fluxograma da Fig. 3.1.

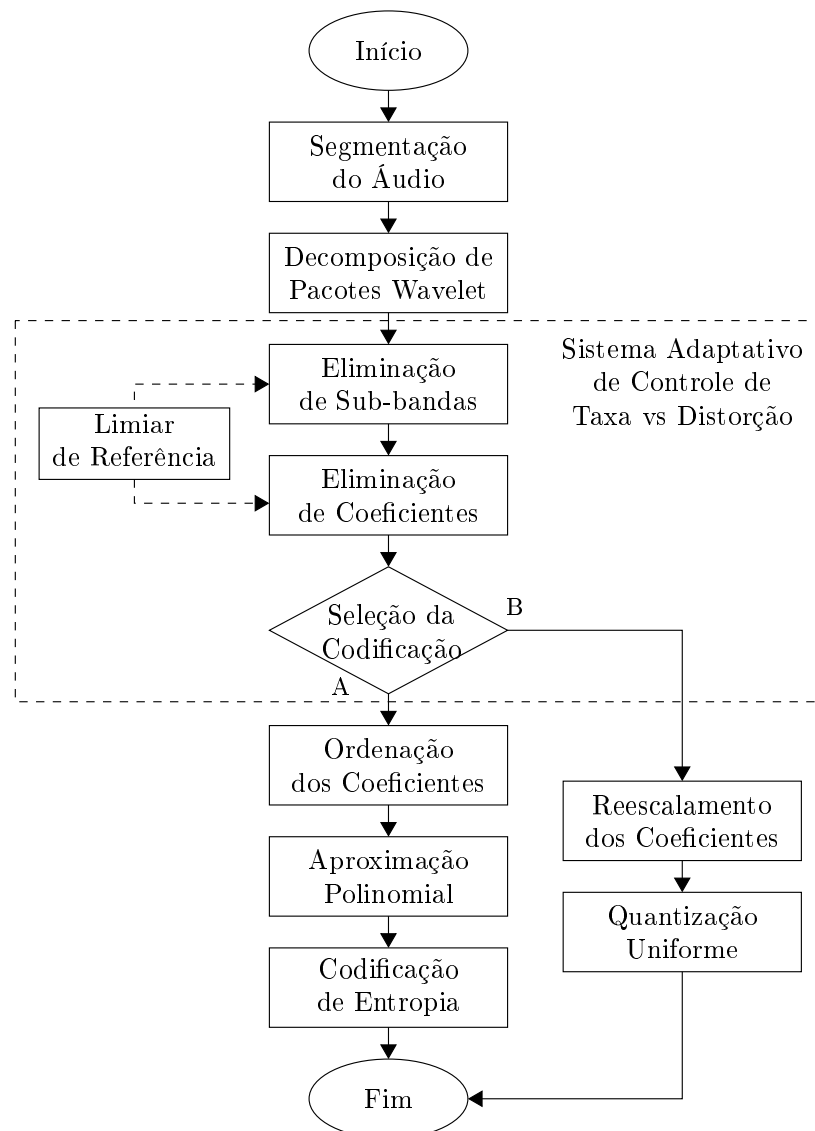


Figura 3.1 – Fluxograma do codificador proposto.



De maneira descritiva, o algoritmo inicia-se com a segmentação do sinal de áudio em blocos. Em seguida, os blocos de áudio são codificados pela WPD sendo assim representados em função das sub-bandas wavelet. Para cada bloco codificado, ocorre então o primeiro e segundo estágio de compressão através da eliminação de sub-bandas e coeficientes com valores abaixo de um limiar de referência. Neste processo de eliminação há necessidade de salvar as posições das sub-bandas e coeficientes. Os coeficientes preservados no processo de eliminação podem então ser representados através de duas formas: na primeira forma, os coeficientes são ordenados do menor e substituídos por valores de uma curva polinomial estimada; são salvos então somente os coeficientes polinomiais. Nesse caso, as posições dos coeficientes são codificadas entropicamente na forma ordenada. Na segunda forma, é simplesmente realizada a quantização direta dos valores dos coeficientes e suas posições guardadas. Em paralelo ao processo de eliminação há um sistema adaptativo de controle de taxa versus distorção para permitir uma aproximada taxa de bits constante na saída do codificador.

O capítulo está organizado da seguinte forma: a Seção 3.1 trata sobre a segmentação do sinal de áudio; a Seção 3.2 especifica a WPD e sua implementação; a Seção 3.3 explica detalhadamente o processo de compressão através da eliminação das sub-bandas e coeficientes menos expressivos; a Seção 3.4 apresenta o critério utilizado na seleção do método de codificação para os coeficientes preservados; a Seção 3.4.1 descreve o método de codificação baseado na ordenação dos coeficientes e aproximação polinomial; a Seção 3.4.2 descreve o código de entropia; a Seção 3.4.3 detalha o método de quantização direta dos coeficientes preservados e suas posições; a Seção 3.5 caracteriza o sistema adaptativo de controle de taxa versus distorção; e finalmente a Seção 3.6 descreve o *bitstream* de saída do codificador e o processo de decodificação do sinal de áudio.

## 3.1 Segmentação do Áudio

O sinal de entrada do codificador é inicialmente segmentado em blocos sobrepostos de áudio de mesmo número de amostras. Essa segmentação introduz alguns benefícios no codificador como menor atraso na saída do sinal e menor custo computacional. Além disso, o processo permite a regularização do sinal de áudio para um número de amostras adequado a decomposição em sub-bandas que será realizada no próximo estágio. Nesse sentido, o número de amostras por bloco  $N$  está em função da configuração desejada para a WPD através de

$$N = 2^{(u+v)} \quad (3.1)$$

onde  $u$  e  $v$  representam, respectivamente, o nível de profundidade utilizado na WPD e o logaritmo na base dois do número de coeficientes nas sub-bandas da WPD. Esses

parâmetros são também chamados de hiper parâmetros neste trabalho e são esclarecidos na Seção 3.2 do estágio WPD enquanto que, um estudo sobre o desempenho do codificador em função destes é dado no Capítulo 4 de resultados.

Para facilitar a decodificação das amostras que participam da sobreposição de blocos, é utilizado no processo de segmentação dito por (2.7) um formato de janela  $w[n]$  retangular (ou também chamado de topo-plano) definido como em (3.2). Esse tipo de janela permite um menor número de tarefas computacionais para segmentar e recuperar o sinal de áudio, visto que não são feitas atenuações nas bordas dos blocos.

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N \\ 0, & \text{outros casos} \end{cases} \quad (3.2)$$

Tratando-se de um sistema de compressão, é ainda considerado o mínimo valor de amostras sobrepostas nos blocos de áudio adjacentes afim de não aumentar o número de amostras a serem armazenadas. Este por sua vez, foi definido através de testes empíricos com vários níveis de  $u$  e  $v$  para evitar o efeito indesejado de cliques no som reconstruído. O valor de amostras sobrepostas estabelecido é dado por

$$S = 2 \lfloor \sqrt{N} \rfloor \quad (3.3)$$

onde  $\lfloor . \rfloor$  é a função piso que retorna o menor número inteiro mais próximo do argumento.

Todo processo de segmentação é ilustrado na Fig. 3.2. Porém, há somente algumas diferenças no primeiro e ultimo bloco do sinal. No primeiro bloco, devido esse não ter um vizinho a esquerda para sobrepor amostras, são adicionadas amostras nulas em seu início. No ultimo bloco, caso esse não tenha amostras suficientes ao número de amostras do bloco, o mesmo é preenchido com amostras nulas.

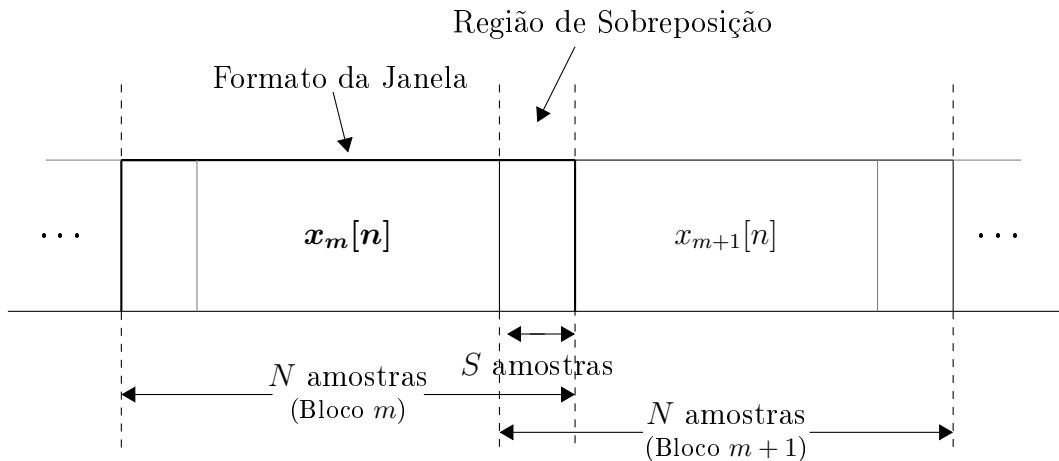


Figura 3.2 – Ilustração do processo de segmentação de áudio do codificador.

Note-se que nesta ilustração  $x_m[n]$  é o bloco de áudio  $m$  obtido que será, posteriormente submetido à WPD do próximo estágio. Além disso, considera-se que no processo de reconstrução do sinal serão somente usados  $N - S$  amostras do bloco. Isso se deve ao fato de que ambas  $S/2$  amostras pertencentes as margens sobrepostas serão recuperadas nos blocos de áudio adjacentes.

## 3.2 Decomposição de Pacotes Wavelet

Após o processo de segmentação, cada bloco de áudio é submetido a WPD. Essa decomposição, conforme descreve-se na Seção 2.5, é uma variação da conhecida DWT onde há diversas possibilidades de escolha para a estrutura da árvore de decomposição. Neste trabalho, utiliza-se a estrutura chamada de árvore binária completa (*full binary tree* [46]) onde há completa decomposição das componentes de aproximação e detalhe em todos os níveis da transformada e que implica em sub-bandas com mesma largura de frequência e número de coeficientes.

A implementação da WPD é feita utilizando o algoritmo rápido de Mallat (Seção 2.5) e, desse modo, as componentes de aproximação e detalhe são respectivamente obtidas através de filtros passa-baixas e passa-altas, conforme Fig 2.12a e 2.12b. Os filtros escolhidos para o processo de decomposição ( $h_d[n]$  e  $g_d[n]$ ), assim como os de reconstrução ( $h_r[n]$  e  $g_r[n]$ ) são da família de wavelets Daubechies com 40 momentos nulos que tem suas respostas impulsivas mostradas na Fig. 3.3.

O número escolhido de momentos nulos é devido alguns resultados em testes empíricos no codificador. Esses testes indicaram ganhos consideráveis de compressão até neste nível de grau de seletividade e pouco ganho nos níveis posteriores, aos quais teria um custo evidente de mais carga computacional nos cálculos de filtragem.

Em razão da implementação rápida da WPD ser realizada por filtros o número total de coeficientes na saída da decomposição em sua forma padrão é maior do que o número total de amostras do bloco de entrada. Isso acontece devido as operações de convolução necessárias em cada nível da transformada.

Calcula-se, de acordo com a teoria de processamento digital de sinais [45], que a convolução de um bloco de  $N$  amostras com um filtro de  $L$  coeficientes resulta em uma sequência de  $N + L - 1$  amostras. Se esse processo é seguido de uma etapa de dizimação por um fator de 2, então o número de amostras resultantes é igual a  $\lfloor (N + L - 1)/2 \rfloor$ . No caso da decomposição pela forma de árvore binária completa, para cada ramo da árvore escolhida ocorre o processo de convolução e sub-amostragem duas vezes. Este processo estende-se a cada nível da transformada por uma potência de dois. Dessa forma,

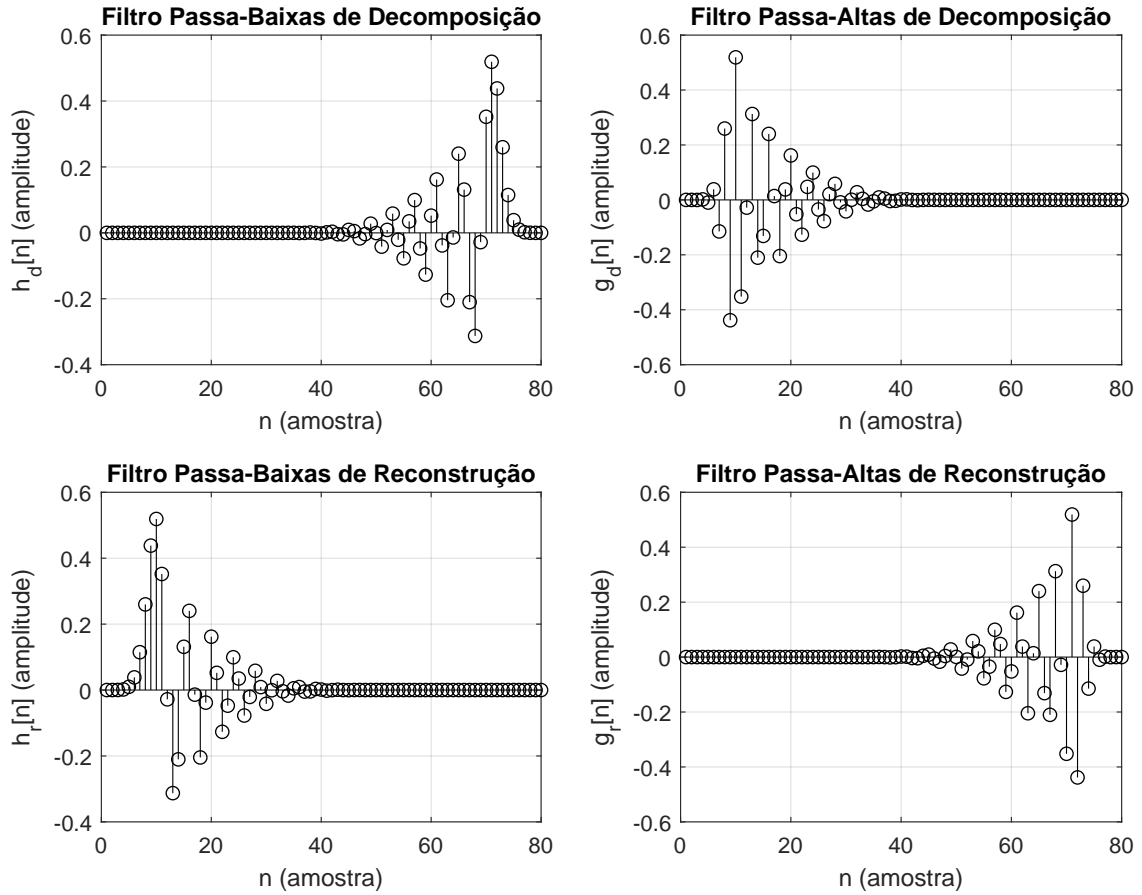


Figura 3.3 – Coeficientes dos filtros de decomposição e reconstrução do codificador.

escolhendo-se uma decomposição com nível de profundidade  $u$  e objetivando-se sub-bandas com número de coeficientes igual a  $2^v$ , seria necessário um número de amostras no bloco de entrada de

$$N_1 = 2^{(u+v)} - (2^u - 1)(L - 1) \quad (3.4)$$

ou

$$N_2 = 2^{(u+v)} - (2^u)(L - 1) \quad (3.5)$$

para que o número total de coeficientes na saída da decomposição respeite o número de coeficientes por sub-bandas desejado. Nota-se, que a dualidade de soluções é devido a função piso do processo de sub-amostragem.

A título de exemplo, considere-se um bloco de áudio que ingressa em uma decomposição de  $u = 2$  níveis de profundidade, filtros de tamanho  $L = 16$  e tamanhos desejados de sub-bandas de 16 coeficientes ( $v = 4$ ). De acordo com (3.4) e (3.5), o tamanho do bloco teria que ser de 19 ou 4 amostras para que possa-se obter na saída 16 coeficientes por sub-banda e o número total de coeficientes da decomposição seria igual a 64 coeficientes.

Observa-se assim que o número de coeficientes de saída é muito maior do que o número de amostras do bloco de entrada. Portanto, à medida que aumenta o nível de

profundidade da transformada e o número de coeficientes dos filtros, tem-se um aumento considerável do número total de coeficientes resultantes do processo de decomposição.

Isso constitui um problema no processo de compressão do codificador, devido ao fato de que o número total de coeficientes das sub-bandas pode ser muito maior do que o do bloco de entrada, o qual comprometerá, em todos os aspectos, o desempenho do codificador tendo esse que armazenar e processar mais valores.

Para solucionar esse problema, a implementação da WPD no codificador é feita pelo método da transformação discreta periódica [46] e consequentemente utilizando convolução circular. Nesse método, cada bloco de  $N$  amostras que ingressa nos ramos de decomposição, é primeiramente estendido periodicamente com  $N - 1$  amostras. O bloco de amostras resultante da filtragem e da subamostragem será portanto periódico com período igual a  $N/2$ . Considerando-se somente um período dessa sequência na saída de cada ramo, obtém-se um número total de amostras igual aquele do bloco de entrada.

Através deste método de implementação, o número de amostras do bloco de entrada na forma de árvore binária completa para um nível de profundidade estipulado  $u$  e sub-bandas desejadas com  $2^v$  coeficientes pode ser dado por (3.1) qual foi mencionado anteriormente no processo de segmentação do codificador. Sendo assim, todo processo de decomposição neste estágio pode ser ilustrado através da Fig. 3.4.

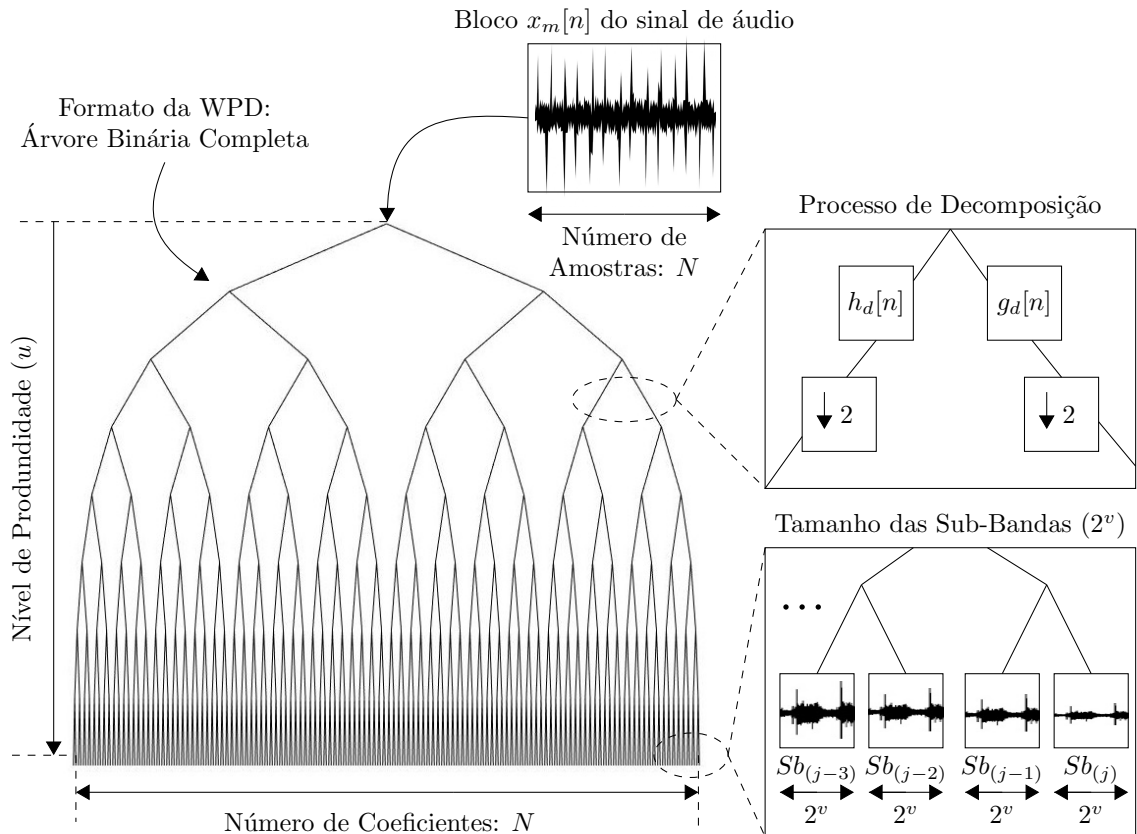


Figura 3.4 – Processo de decomposição por pacotes wavelet do codificador.

Observa-se que cada bloco de entrada  $x_m[n]$  é agora representado através do conjunto de  $2^u$  sub-bandas que são denotadas através de  $Sb_j$  onde  $j = \{1, \dots, 2^u\}$ . Adicionalmente, considera-se que os coeficientes dentro de cada sub-bandas são denotados como  $d_j[p]$  onde o subíndice  $j$  é pertinente a sub-banda à qual esse coeficiente pertence e, o índice  $p$  é a posição do coeficiente na sub-banda (isto é,  $p = \{1, \dots, 2^v\}$ ).

Do ponto de vista de análise em tempo-frequência (Seção 2.5), cada coeficiente representa um intervalo de tempo na faixa de frequência da sub-banda. Essas sub-bandas, nesta configuração de WPD, tem largura de frequência equivalente à  $Fs/2^{u+1}$  sendo  $Fs$  a frequência de amostragem utilizada no sinal de entrada. Um exemplo da divisão do espectro para uma decomposição com nível de profundidade  $u = 8$  e frequência de amostragem  $Fs = 44,1$  kHz é dada na Fig. 3.5.

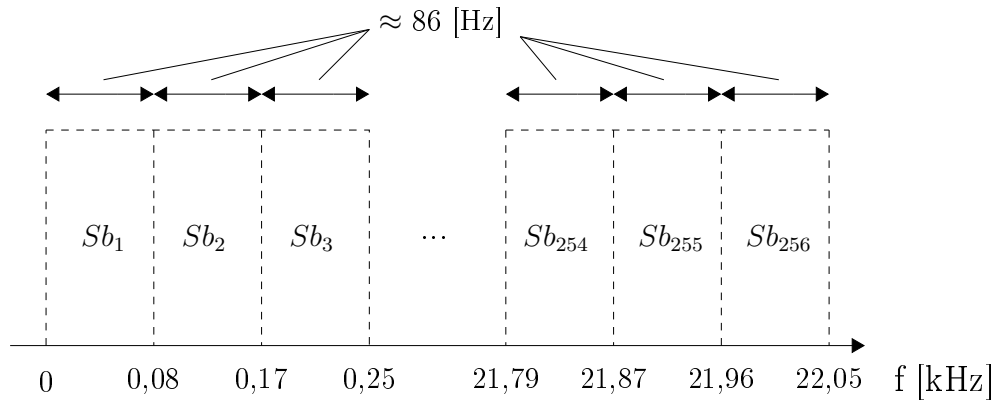


Figura 3.5 – Exemplo das faixas de frequências das sub-bandas geradas na WPD.

### 3.3 Eliminação de Sub-bandas e Coeficientes Menos Expressivos

Para cada bloco de áudio decomposto em sub-bandas no estágio anterior, acontece o primeiro e segundo estágio de compressão através da eliminação (ou também chamada anulação) das sub-bandas com todos os valores de coeficientes menores que um determinado limiar de referência e, através da eliminação dos coeficientes presentes nas sub-bandas preservadas que também estão abaixo do limiar de referência, respectivamente.

O limiar de referência é uma aproximação do limiar auditivo da ISO 226:2003 decomposta nas sub-bandas wavelets. Como descrito na revisão bibliográfica da Seção 2.2, esse limiar refere-se ao mínimo SPL necessário para um ouvinte humano ouvir qualquer som e, portanto, este processo de compressão, supondo ser realizado idealmente, não afeta a qualidade do áudio pois há eliminação de dados irrelevantes à finalidade do áudio.

A aproximação sobre o limiar padronizado é mostrada na Fig. 3.6 onde observa-se diferenças no início e fim das curvas. No início, os valores padronizados são substituídos

pelos valores de uma reta começando em 16 Hz com 20 dB e terminando em 1 kHz com 0 dB de SPL. No fim, há uma extensão na curva após 12,5 kHz através de duas retas, uma de 12,5 até 14,2 kHz com SPL indo de 14,63 a 9 dB e outra até 22,05 kHz com 20 dB.

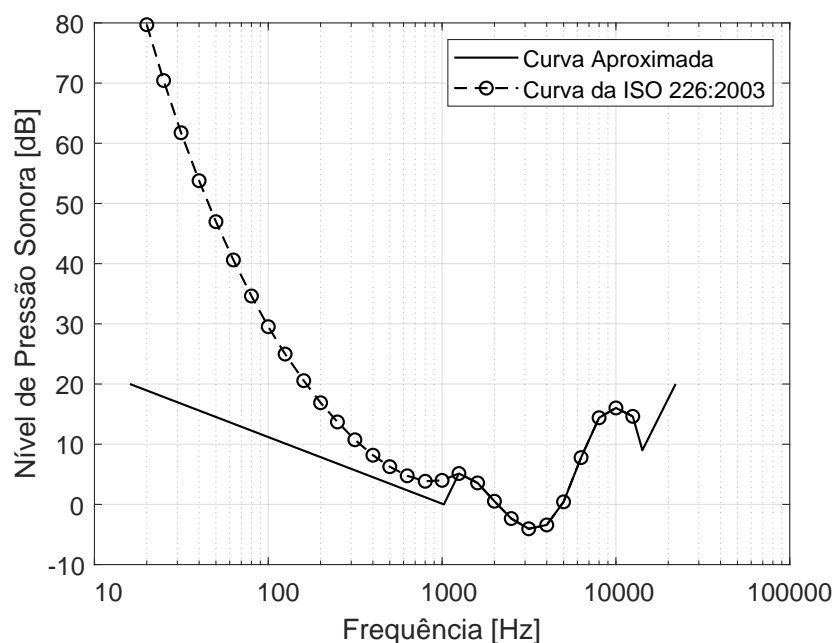


Figura 3.6 – Comparação do limiar auditivo definido para o codificador proposto e o padronizado pela ISO 226:2003.

Nota-se que, a linha tracejada na curva padronizada é uma interpolação das marcações com círculos. Essa é feita por retas e faz parte dos valores assumidos do modelo do limiar auditivo utilizado no codificador proposto. De fato, somente 29 frequências são cobertas no padrão ISO 226:2003 (sendo uma limitação desta) as quais são representadas através das marcações por círculos [35].

A reta aproximada nas baixas frequências foi escolhida visando um declínio mais lento do nível de SPL nessa faixa de frequência visto que há uma alta variabilidade de SPL nessa faixa qual produzem erros consideráveis na decomposição do limiar nas sub-bandas. As retas pertinentes a extensão no final da curva tem o intuito de preencher os valores faltantes da curva padronizada em relação a um sinal amostrado a 44,1 kHz.

O valores de declínio e crescimento destas retas são frutos de testes empíricos os quais mostram melhores resultados no *trade-off*<sup>1</sup> da qualidade de áudio versus compressão.

A decomposição do limiar nas sub-bandas wavelets é feita simplesmente através da discretização da curva em função da largura em frequência destas sub-bandas. Isto é, para cada sub-banda representando uma faixa de frequências na curva, é assumido um único valor de SPL sendo, este valor dado pela frequência central da sub-banda correspondente.

<sup>1</sup> Termo em inglês, destinado, neste caso, ao ato de escolher uma coisa em detrimento de outra.

Para exemplificar, a Fig. 3.7 mostra como fica o limiar auditivo discretizado em sub-bandas resultantes de uma WPD com nível de profundidade  $u = 8$ .

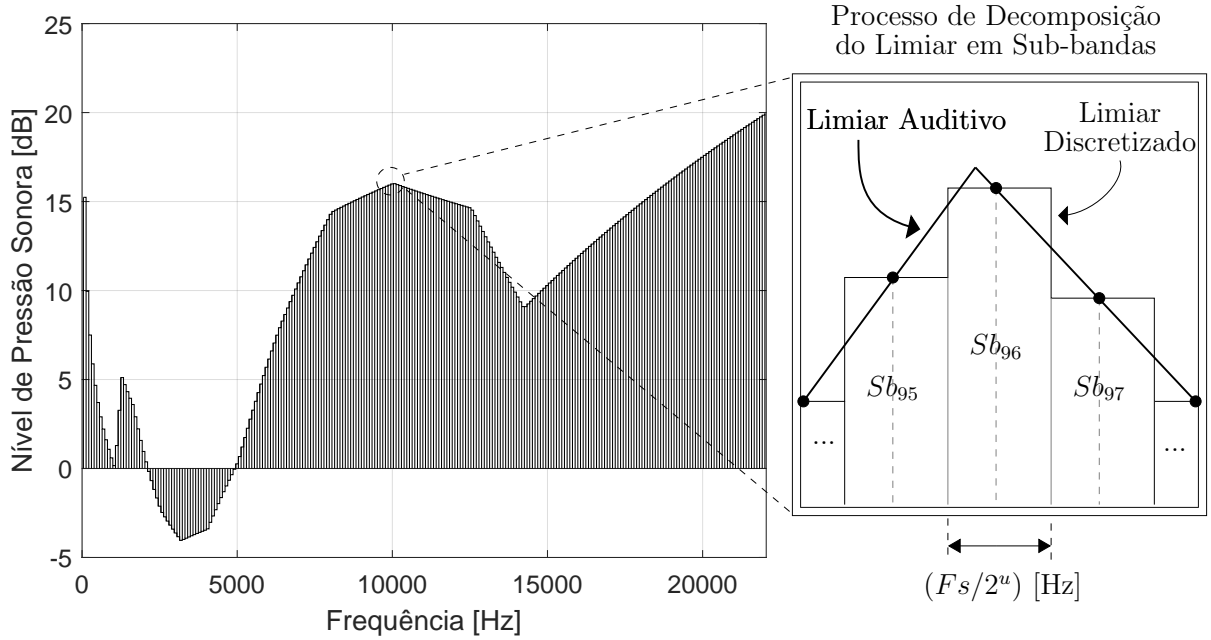


Figura 3.7 – A decomposição da curva do limiar auditivo nas sub-bandas wavelets.

Outro processo a ser realizado sobre a curva é a multiplicação por um ganho  $\theta$ . A principal razão para esse ganho é a conversão do limiar auditivo que está em SPL para nível de áudio digital. Dificilmente define-se este em um único valor, desse modo, testes empíricos mostraram bons resultados quando  $\theta$  varia entre 0.01 a 0.15, dependendo do som. Contudo, este ganho também é controlado pelo controle de taxa versus distorção (*rate versus distortion control*, RDC) adotado no codificador e discutido na seção 3.5; sendo assim  $\theta$  varia entre 0 a 1.

Por fim, após a escolha do fator de ganho  $\theta$  pelo sistema RDC, é encontrado o máximo valor absoluto dos coeficientes de cada sub-banda  $Sb_j$  de um bloco de áudio  $x_m[n]$  e comparado com o valor do modelo do limiar auditivo naquela faixa de frequência e, caso seja menor, a sub-banda é eliminada. Essas sub-bandas eliminadas são consideradas com coeficientes nulos posteriormente na decodificação. Um exemplo desse processo é dado na Fig. 3.8 para duas sub-bandas arbitrárias de 256 coeficientes (isto é,  $v = 8$ ).

Para que o decodificador conheça quais sub-bandas foram preservadas e eliminadas, é enviado no cabeçalho dos blocos codificados dois sinalizadores: o primeiro com a quantidade de sub-bandas preservadas e o segundo com as posições das sub-bandas preservadas ou eliminadas na árvore de decomposição. A decisão entre quais posições enviar é dada pela que tiver menor número e, este processo é de conhecimento do decodificador em razão do envio da quantidade de sub-bandas preservadas no primeiro sinalizador. Estes



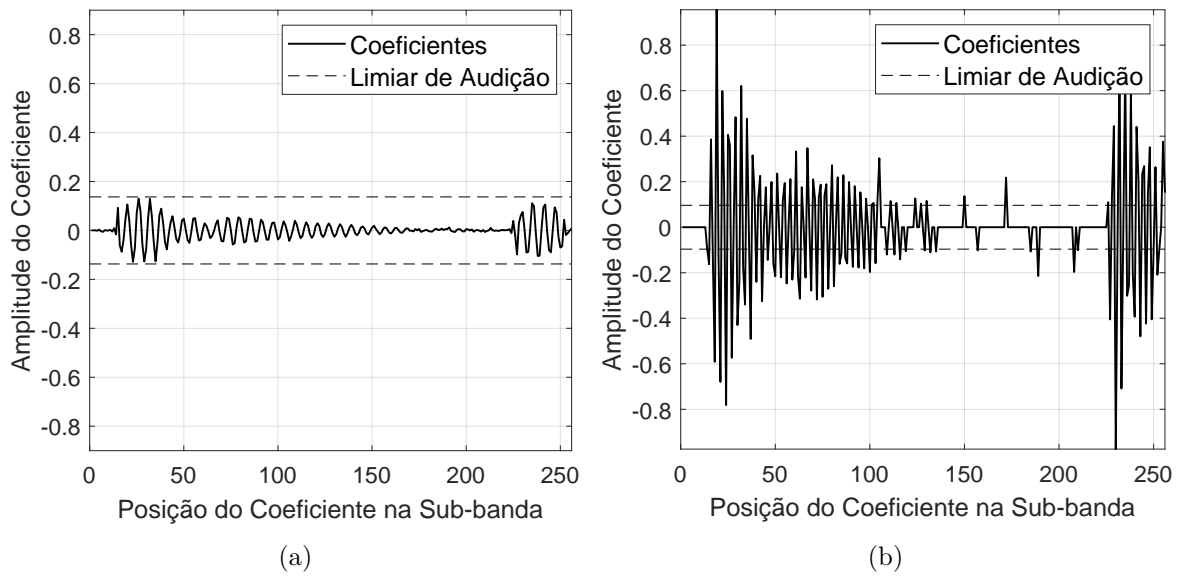


Figura 3.8 – Exemplo de uma sub-banda (a) eliminada e uma (b) preservada.

são respectivamente representados com  $(u + 1)$  e  $(u)$  bits.

Após à eliminação das sub-bandas, cada sub-banda restante têm seus coeficientes com valores abaixo do limiar de referência também eliminados. Para isso, cada coeficiente na sub-banda é analisado individualmente em relação ao modelo do limiar estabelecido positivamente e negativamente. Para a mesma sub-banda preservada no exemplo anterior, o processo de eliminação dos coeficientes pode ser visto na Fig. 3.9.

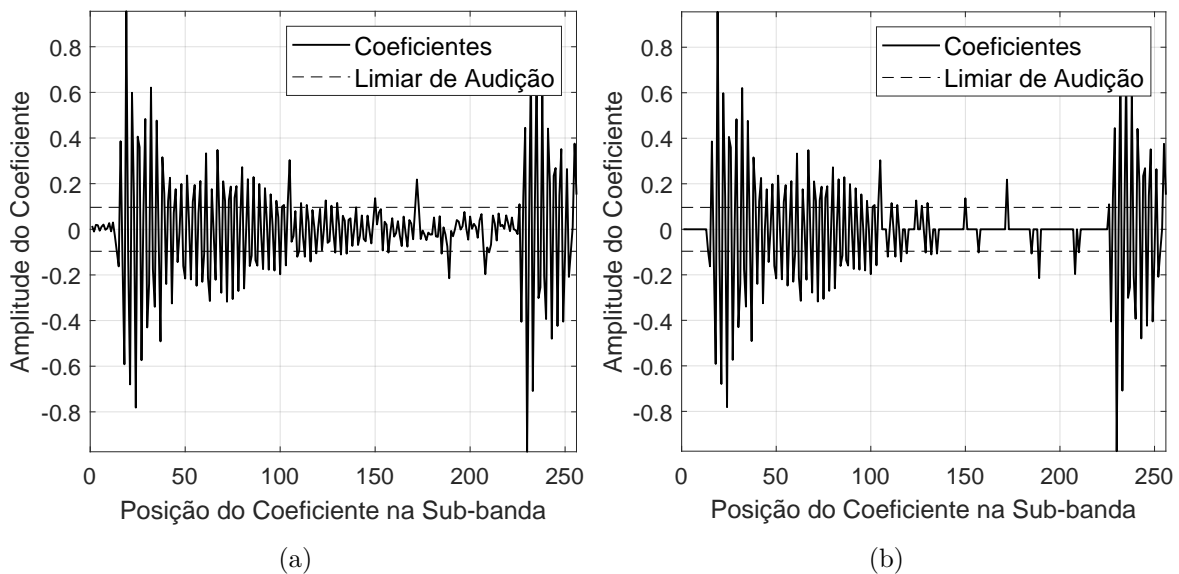


Figura 3.9 – Exemplo do processo de eliminação dos coeficientes em uma sub-banda preservada: (a) antes e (b) depois.

Este processo também implica na criação de sinalizadores para informar o decodificador sobre quais coeficientes foram eliminados e quais foram preservados. Porém, o jeito

que está informação é passada varia de acordo com o método de codificação escolhido para os valores dos coeficientes preservadas que é explicado nas seções seguintes.

### 3.4 Métodos de Codificação para os Coeficientes Mais Expressivos

Para codificar os coeficientes preservados em cada sub-banda, o codificador decide por dois diferentes métodos, são eles: o método de ordenação e aproximação polinomial, chamado de método A e, o método de reescala e quantiza, chamado de método B. Ambos métodos são detalhados nas próximas seções e a decisão tomada para cada sub-banda é salva em uma lista no cabeçalho do bloco codificado através de 1 bit.

Para decidir entre os dois métodos, o codificador simplesmente compara qual seria o consumo de bits que seria produzido em cada método e escolhe o menor, sendo o principal fator de decisão, a quantidade de coeficientes preservados na sub-banda sendo codificada. O cálculo dos consumos e a decisão são definidas pela seguinte função

$$f(R, \lambda, \delta, v) = \begin{cases} \text{Método A,} & 16(R - \lambda) > (32\delta + v + 2) \\ \text{Método B,} & 16(R - \lambda) \leq (32\delta + v + 2) \end{cases} \quad (3.6)$$

onde  $R$  é a quantidade de coeficientes preservados e,  $\delta$  é o número de parâmetros do polinômio adotado no método A e,  $\lambda$  é um parâmetro chamado de fator de peso. Dado os valores destes parâmetros cada lado da desigualdade representa o consumo de um método.

O fator de peso  $\lambda$  está presente na decisão para permitir um nível de controle de taxa versus distorção através da sua variação entre 1 a  $2^v$  que pondera a decisão (3.6) entre os métodos de codificação A e B. Este parâmetro assim como o valor de  $\delta$  são variados pelo sistema RDC que será detalhado na Seção 3.5.

#### 3.4.1 Ordenação e Aproximação Polinomial (Método A)

No método A, os coeficientes preservados são ordenados e substituídos pelos valores de uma curva polinomial estimada através de aproximação polinomial. Este método é bastante efetivo em relação a compressão pois a curva polinomial estimada pode ser representada no decodificador através dos parâmetros polinomiais os quais são codificados com 32 bits em cada sub-banda.

O processo de ordenação é feito em forma ascendente de valor e é adequada para tornar mais suave a distribuição dos valores dos coeficientes nas sub-bandas e assim, permitir uma representação efetiva destes valores através de uma curva polinomial. De maneira

intuitiva, na Fig. 3.10 é mostrado um exemplo do resultado deste processo para a mesma sub-banda escolhida no exemplo do processo de eliminação de coeficientes (Fig.3.9).

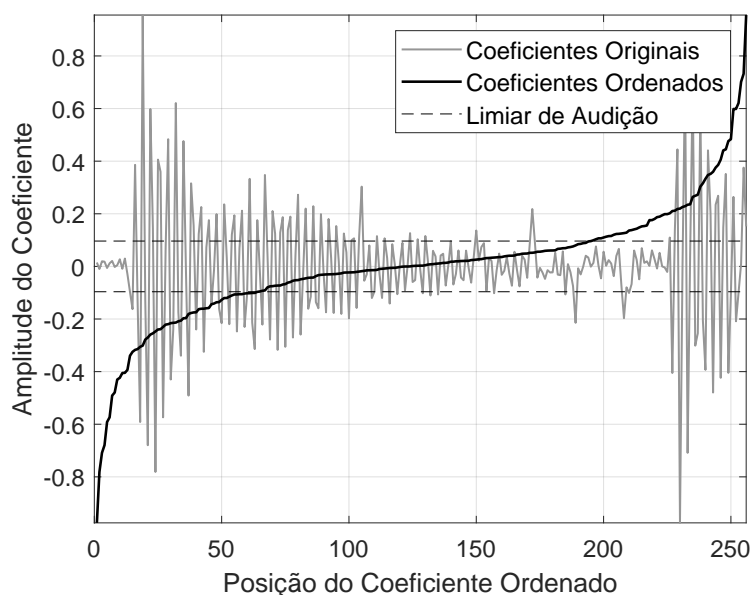


Figura 3.10 – Exemplo de ordenação dos coeficientes feita de forma crescente de valor.

Observa-se que são considerados todos os 2<sup>o</sup> coeficientes da sub-banda para o processo de ordenação e, não só os preservados ao processo de eliminação do estágio anterior. Todos estes também serão considerados posteriormente no cálculo da aproximação polinomial porém, os coeficientes eliminados são mantidos com valores nulos no decodificador (conforme Fig. 3.11) devido a ausência de suas posições após a eliminação.

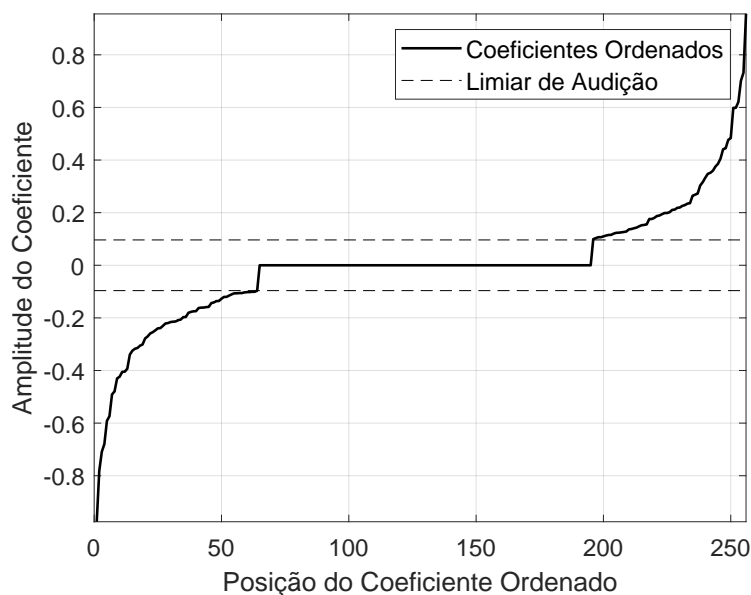


Figura 3.11 – Exemplo de ordenação dos coeficientes feita de forma crescente de valor.

Neste método, as posições dos coeficientes preservados são armazenados na ordem dos coeficientes ordenados, isto é, as posições originais dos coeficientes na sub-banda são

armazenados em um vetor na ordem que foram ordenadas neste processo. Além disso, as posições são posteriormente codificados entropicamente como discutido na Seção 3.4.2 e devido ao processo de eliminação de coeficientes há uma redução significativa na quantidade de posições a serem codificadas conforme apresenta-se na Fig. 3.12.

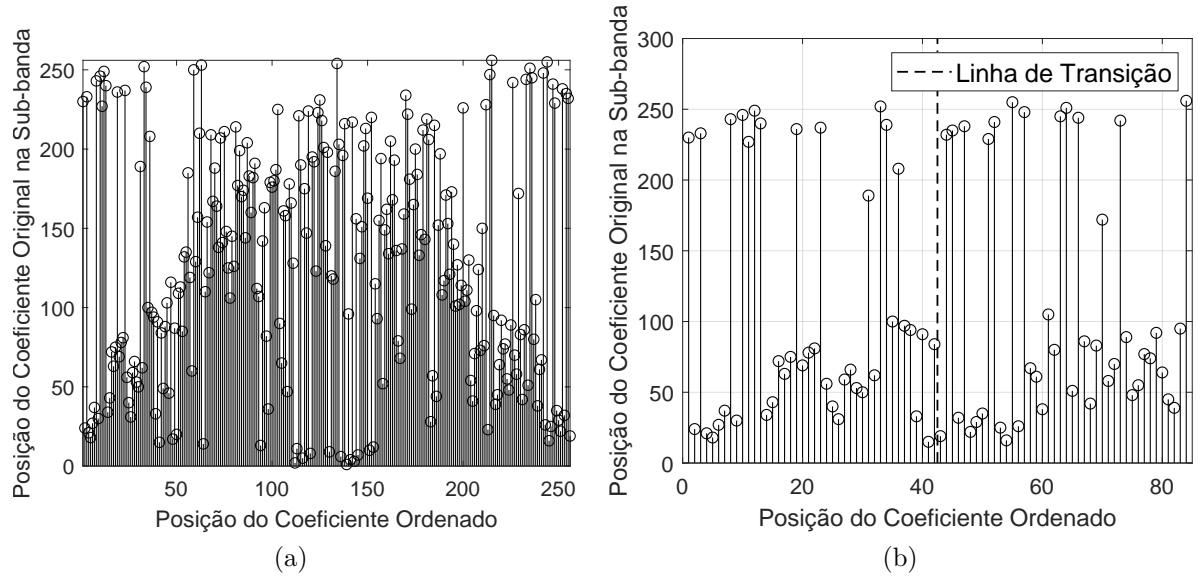


Figura 3.12 – Posições a serem armazenadas: (a) antes e (b) depois da eliminação.

A partir das Figuras 3.11 e 3.12b é justificado a criação de dois sinalizadores para indicar a quantidade de coeficientes preservados nas sub-bandas. Isso, porque é necessário um para os coeficientes abaixo do limiar de audição e outro para os acima, para que assim o decodificador conheça os limites de cada região. Ambos sinalizadores são representados por  $(v + 1)$  bits no conteúdo de cada sub-banda codificada.

Após o processo de ordenação, é realizado a aproximação polinomial dos coeficientes ordenados utilizando o critério dos mínimos quadrados descrito na revisão da Seção 2.7. Adicionalmente, é considerado na implementação da aproximação uma normalização sobre as variáveis observáveis (i.e., dos valores de posições ordenados) para permitir um melhor desempenho numérico. Esta normalização é feita através da subtração pela média e divisão pelo desvio padrão da sequência de posições ordenada  $(1, \dots, 2^v)$ .

Os polinômios utilizados na aproximação são selecionados pelo sistema RDC (Seção 3.5) através da lista apresentada na Tabela 3.1. Este conjunto de polinômios foi definido através de testes onde várias amostras de sub-bandas exibiam simetria impar e coeficientes nulos na origem, boa parte ocasionado pelo processo de ordenação e normalização. Deste modo, foram considerados alguns polinômios sem os termos de graus pares e todos sem o termo constante.

Nota-se que a ultima coluna é designada ao números de parâmetros do polinômio, ou seja, o número de parâmetros  $\beta$  a serem estimados. Além disso, estes foram ordenados do

Tabela 3.1 – A Lista de Polinômios para a Aproximação Polinomial

Polinômio da Aproximação Polinomial	Termos Pares	Ordem Polinomial	Número de Parâmetros ( $\delta$ )
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^2 + \dots + \beta_{13}(p)^{14} + \beta_{14}(p)^{15}$	Com	15	15
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^2 + \dots + \beta_{11}(p)^{12} + \beta_{12}(p)^{13}$	Com	13	13
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^2 + \dots + \beta_9(p)^{10} + \beta_{10}(p)^{11}$	Com	11	11
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^2 + \dots + \beta_7(p)^8 + \beta_8(p)^9$	Com	9	9
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^3 + \dots + \beta_6(p)^{13} + \beta_7(p)^{15}$	Sem	15	8
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^3 + \dots + \beta_5(p)^{11} + \beta_6(p)^{13}$	Sem	13	7
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^3 + \dots + \beta_4(p)^9 + \beta_5(p)^{11}$	Sem	11	6
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^3 + \dots + \beta_3(p)^7 + \beta_4(p)^9$	Sem	9	5
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^3 + \beta_2(p)^5 + \beta_3(p)^7$	Sem	7	4
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^3 + \beta_2(p)^5$	Sem	5	3
$\hat{d}_j[p] = \beta_0(p) + \beta_1(p)^3$	Sem	3	2

maior ao menor propositalmente, a fim de indicar menores parâmetros para armazenar ao custo de piores estimações dos valores reais dos coeficientes, sendo o parâmetro  $\delta$  considerado pelo sistema RDC para o controle de taxa versus distorção.

A estimação destes parâmetros é feita através das relações matriciais (2.42) e (2.46) sendo que, a técnica com ponderação de erro é somente utilizado nos polinômios com os termos pares devido a ineficiência nos demais polinômios. Para estes casos, são dadas ponderações de 10:1 no ponto inicial e final da sub-banda devido a quantidade de energia presente nestes pontos críticos, conforme ilustra-se na Fig. 3.13 para o exemplo anterior.

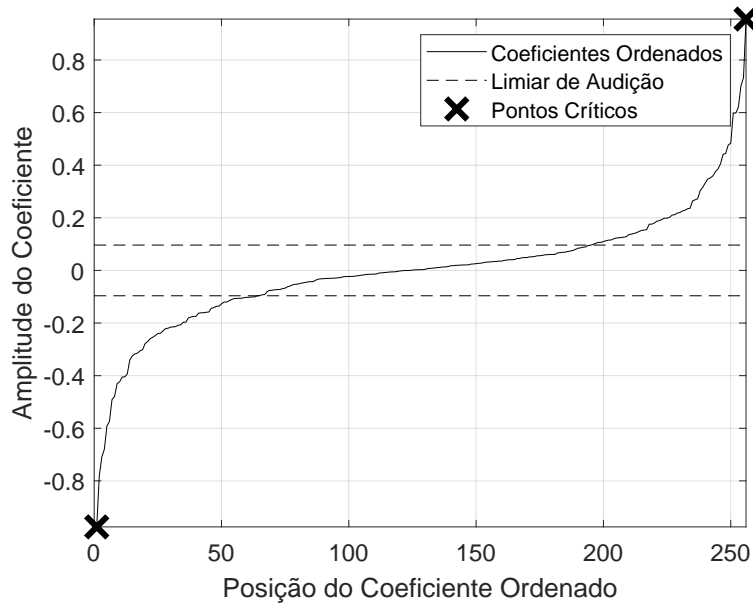


Figura 3.13 – Pontos críticos considerados na aproximação polinomial pelo MMQP.

Por fim, para a mesma sub-banda com os coeficientes ordenados da Fig. 3.10, um exemplo da curva estimada pela aproximação é dado na Fig. 3.14 para o caso de um

polinômio com número de parâmetros  $\delta = 8$  (isto é, um polinômio de ordem 15 mas sem os termos pares e o termo constante).

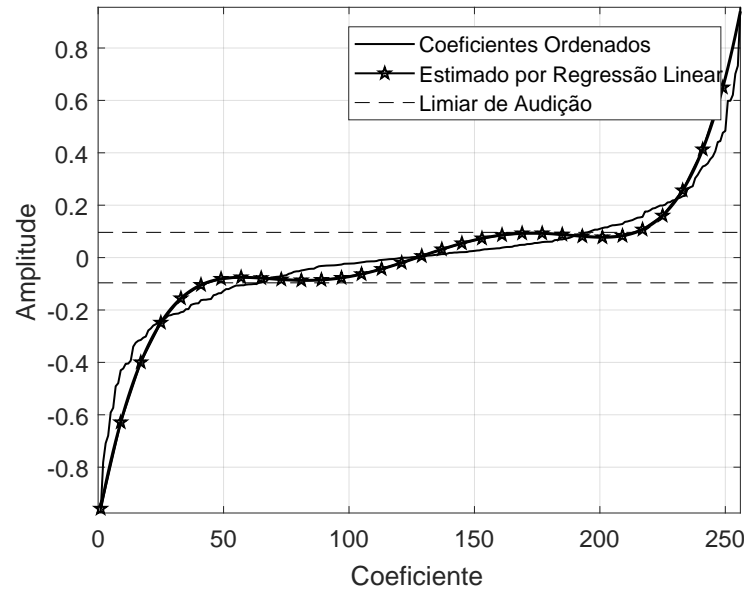


Figura 3.14 – Um exemplo da curva estimada por aproximação polinomial.

### 3.4.2 Código de Entropia (Método A)

Além dos estágios de ordenação e aproximação polinomial, o método A contém um estágio de codificação de entropia. Tal estágio realiza a compactação das posições dos coeficientes preservados nas sub-bandas (visto em forma de exemplo na Fig. 3.12b). Para isso, é utilizado um código de entropia desenvolvido neste trabalho baseado no código de entropia utilizado no codificador adotado pelo JPEG (*Joint Photographic Experts Group*) [49].

Antes de chegar na técnica de codificação baseada no JPEG foram testadas outras técnicas para a compactação das posições. Em geral, as técnicas eram de codificação de entropia, ou seja, que exploravam as redundâncias estatísticas do sinal. Porém, também foram exploradas técnicas auxiliares (e antecessoras) ao estágio de codificação de entropia como as de predição linear que são baseadas na correlação das posições.

Entre as técnicas testadas estavam algumas conhecidas da literatura como o código de Huffman, o de Golomb e o Aritmético [17] e outras não tão conhecidas como a de Oliveira [50]. Essas foram implementadas por métodos estáticos baseados nos histogramas das posições, assim como métodos de dois passos e adaptativos. Contudo, para o conjunto de amostras testado, nenhuma destas técnicas teve maior compactação que a baseada no JPEG.

No código de entropia do JPEG, os coeficientes da fonte de informação são agrupados pela corrida<sup>2</sup> e categoria<sup>3</sup>. Para cada grupo deste era indexado um código de Huffman gerado a partir da probabilidade que cada grupo ocorrer. A compactação ocorria então porque na média as palavras códigos finais de cada coeficiente (resultante de sua corrida, categoria e indexação) produziam uma taxa de bits menor.

No código de entropia desenvolvido neste trabalho, somente as categorias de coeficientes e o código de Huffman são utilizados, ou seja, sem as corridas. Este código consiste em quatro categorias as quais são sinalizadas utilizando diferentes árvores de Huffman escolhidas através da frequência de coeficientes preservados em cada categoria, atribuindo assim palavras códigos menores as categorias com mais coeficientes.

As quatro categorias são nomeadas de A, B, C e D e seus valores cobrem toda faixa dinâmica de posições dos coeficientes na sub-banda, conforme Fig. 3.15. Estas categorias no JPEG variam de tamanho, com tamanhos menores nas faixas de valores mais frequentes, porém, neste código, não é assumido a priori que alguma faixa de valores é mais frequente que a outra, portanto, são de igual tamanho sendo este de  $2^v/4$ ,

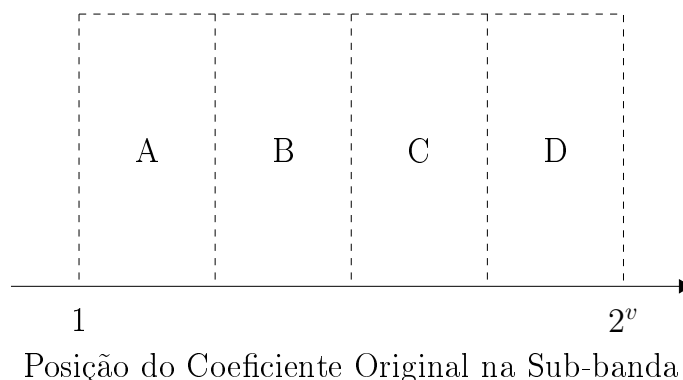


Figura 3.15 – Ilustração das categorias do código de entropia.

Para exemplificar, considere uma sub-banda de 256 coeficientes (ou seja, hiper parâmetro  $v = 8$ ); neste caso, o tamanho de cada categoria, a sua faixa de valores e o respectivo número de bits para representar os valores da faixa (dado pelo logaritmo na base 2 do tamanho da categoria) são vistos na Tabela 3.2.

Nota-se que, para indicar um valor de posição neste novo formato, é preciso sinalizar em qual categoria está a posição e, em seguida, o valor da posição de 1 a 64. Para esta sinalização o JPEG utiliza uma estrutura de árvore Huffman fixa para todo processo de codificação. Neste código não é fixada somente uma árvore, mas na verdade utilizam-se todas as combinações de árvores possíveis para o número de categorias.

<sup>2</sup> O termo *corrida* denota uma sequência de valores repetidos e é pertinente à técnica RLE.

<sup>3</sup> O termo *categoria* denota a estruturação dos coeficientes por faixas de valores.

Tabela 3.2 – Um exemplo das categorias formadas pelo código de entropia.

Categoria	Tamanho da Categoria	Faixa de Valores	Tamanho em Bits
<b>A</b>	64	de 1 até 64	6
<b>B</b>	64	de 65 até 128	6
<b>C</b>	64	de 129 até 192	6
<b>D</b>	64	de 193 até 256	6

Para 4 categorias, ou seja, 4 símbolos, as duas estruturas de árvores Huffman possíveis podem ser vistas na Figura 3.16.

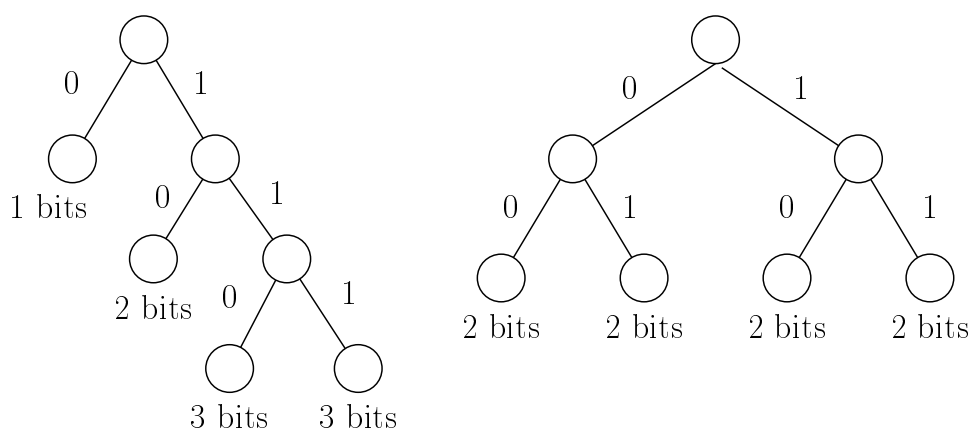


Figura 3.16 – As duas estruturas de árvores de Huffman possíveis para 4 símbolos.

A árvore à direita, com folhas de mesmo número de bits, produz a chamada (neste trabalho) de combinação *Uncoded*, visto que estás, somadas às categorias de tamanho igual, produzem palavras códigos com mesmo número de bits e portanto, sem compactação. As sinalizações por meio desta árvore são dadas as sub-bandas com quantidade de coeficientes distribuídas uniformemente entre as categorias.

Na árvore à esquerda, cada folha representa um tamanho de palavra código diferente e, desse modo, existe um número de combinações possíveis para sinalizar as categorias através dessa árvore. Para descrever essas combinações alguém pode nomear as folhas em função de tamanho de seu código como mostrado Tabela 3.3, sendo o asterisco dado a uma das palavras código de mesmo tamanho para que não haja ambiguidade.

Tabela 3.3 – Nomes dados as folhas da árvore com tamanho variado de código.

Código de Huffman	Tamanho em Bits	Nome da Folha
0	1	<b>1</b>
10	2	<b>2</b>
110	3	<b>3*</b>
111	3	<b>3</b>

O número de combinações possíveis para associar cada folha às categorias A, B, C e D é dado então pela permutação das categorias, ou seja,  $4! = 24$  combinações. Entretanto,



observa-se que no sentido de tamanho de bits, há pares de folhas iguais e não faria diferença quais categorias seriam associadas a elas. Desse modo, o número de combinações cai para 12 as quais são mostradas na Tabela 3.4.

Tabela 3.4 – Tabela com as 12 combinações das categorias nas folhas.

Categoria	Combinação de Árvore											
	1 <sup>a</sup>	2 <sup>a</sup>	3 <sup>a</sup>	4 <sup>a</sup>	5 <sup>a</sup>	6 <sup>a</sup>	7 <sup>a</sup>	8 <sup>a</sup>	9 <sup>a</sup>	10 <sup>a</sup>	11 <sup>a</sup>	12 <sup>a</sup>
<b>A</b>	1	1	1	2	2	2	3*	3*	3*	3*	3*	3*
<b>B</b>	2	3*	3*	1	3*	3*	1	1	2	2	3	3
<b>C</b>	3*	2	3	3*	1	3	2	3	1	3	1	2
<b>D</b>	3	3	2	3	3	1	3	2	3	1	2	1

Nota-se que, para distinguir folhas com mesmo tamanho de bits na combinação, uma simples regra de ordem lexicográfica é utilizada, isto é, sempre a categoria com a letra mais antecessora na ordem alfabética é sinalizada com a folha marcada pelo asterisco. Para exemplificar a indexação das categorias através das folhas, na Fig. 3.17 é ilustrado a primeira e décima combinação de sinalização das categorias.

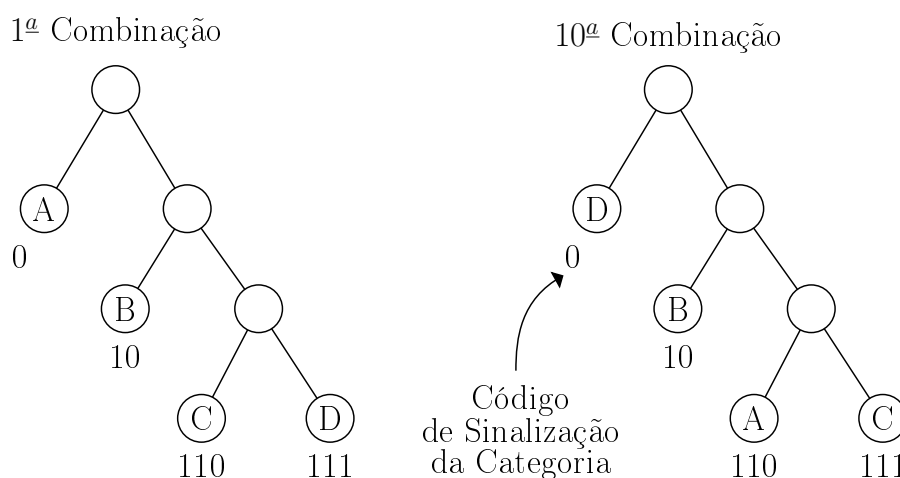


Figura 3.17 – Exemplo da 1<sup>a</sup> e 10<sup>a</sup> combinação de folhas para sinalização das categorias.

A título de exemplo, pode-se verificar na Tabela 3.5 como seria o número de bits total das palavras códigos a serem utilizados nas posições dos coeficientes preservados para todas as 12 combinações possíveis em um caso com sub-bandas de 256 coeficientes. Observa-se que, o tamanho em bits total das palavras códigos varia entre 7, 8 ou 9. A maior efetividade do código diretamente relacionada a ter mais posições de coeficientes com palavras códigos de 7 ou 8 bits que, nesse caso, estão abaixo ou são iguais a representação sem codificar das posições que seria de  $\log_2(256) = 8$  bits.

Para saber qual combinação usar em cada sub-banda é realizado uma contagem dos coeficientes em cada categoria (visto que estão incompletas pelo processo de eliminação

Tabela 3.5 – O tamanho total das palavras códigos usando cada combinação.

Cat	1 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total	2 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total
<b>A</b>	0	1	6	<b>7</b>	0	1	6	<b>7</b>
<b>B</b>	10	2	6	<b>8</b>	110	3	6	<b>9</b>
<b>C</b>	110	3	6	<b>9</b>	10	2	6	<b>8</b>
<b>D</b>	111	3	6	<b>9</b>	111	3	6	<b>9</b>

Cat	3 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total	4 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total
<b>A</b>	0	1	6	<b>7</b>	10	2	6	<b>8</b>
<b>B</b>	110	3	6	<b>9</b>	0	1	6	<b>7</b>
<b>C</b>	111	3	6	<b>9</b>	110	3	6	<b>9</b>
<b>D</b>	10	2	6	<b>8</b>	111	3	6	<b>9</b>

Cat	5 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total	6 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total
<b>A</b>	10	2	6	<b>8</b>	10	2	6	<b>8</b>
<b>B</b>	110	3	6	<b>9</b>	110	3	6	<b>9</b>
<b>C</b>	0	1	6	<b>7</b>	111	3	6	<b>9</b>
<b>D</b>	111	3	6	<b>9</b>	0	1	6	<b>7</b>

Cat	7 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total	8 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total
<b>A</b>	110	3	6	<b>9</b>	110	3	6	<b>9</b>
<b>B</b>	0	1	6	<b>7</b>	0	1	6	<b>7</b>
<b>C</b>	10	2	6	<b>8</b>	111	3	6	<b>9</b>
<b>D</b>	111	3	6	<b>9</b>	10	2	6	<b>8</b>

Cat	9 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total	10 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total
<b>A</b>	110	3	6	<b>9</b>	110	3	6	<b>9</b>
<b>B</b>	10	2	6	<b>8</b>	10	2	6	<b>8</b>
<b>C</b>	0	1	6	<b>7</b>	111	3	6	<b>9</b>
<b>D</b>	111	3	6	<b>9</b>	0	1	6	<b>7</b>

Cat	11 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total	12 <sup>a</sup> Comb.	Código (bits)	Categoria (bits)	Total
<b>A</b>	110	3	6	<b>9</b>	110	3	6	<b>9</b>
<b>B</b>	111	3	6	<b>9</b>	111	3	6	<b>9</b>
<b>C</b>	0	1	6	<b>7</b>	10	2	6	<b>8</b>
<b>D</b>	10	2	6	<b>8</b>	0	1	6	<b>7</b>

de coeficientes) e escolhido a melhor combinação das 12 possíveis. Um exemplo dessa contagem para a sub-banda anterior pode ser vista na Fig. 3.18.

A escolha é baseada na ordem das categorias pela quantidade de coeficientes nelas, isto é, quanto maior o número de coeficientes na categoria, menor a palavra código de

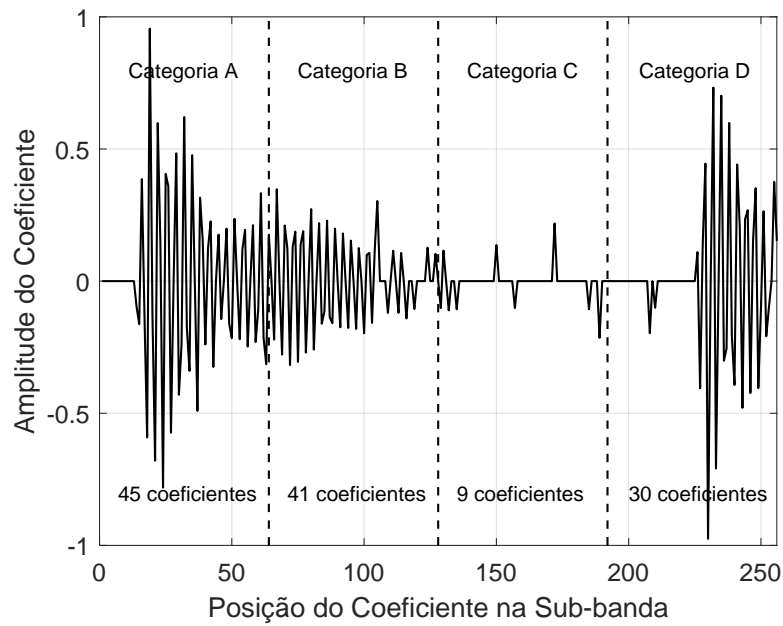


Figura 3.18 – Exemplo de contagem dos coeficientes preservados por categoria.

sinalização desta. Após a atribuição correta da combinação, é calculado o consumo de bits dessa codificação e comparado com o consumo da combinação *Uncoded* e a que tiver o menor consumo de bits, é escolhida para codificar as posições dos coeficientes.

Para o exemplo da Fig. 3.18, a ordem de categorias pela quantidade de coeficientes é: A, B, D, C. Nesse caso, a combinação correta seria a 1ª combinação e, portanto, o consumo de bits dessa combinação pode ser calculado através de  $(45)7 + (41)8 + (30)9 + (9)9 = 994$  bits e da combinação *Uncoded*  $(45)8 + (41)8 + (30)8 + (9)8 = 1000$  bits. Desse modo, a escolha para a codificação das posições seria através da 1ª e o ganho associado seria de 6 bits.

Neste processo, para ser possível a recuperação das posições no decodificador, este precisa saber qual combinação foi utilizada. Sendo assim, é enviado por sub-banda uma sinalização para representar as 13 combinações (isto é, as 12 combinações da árvore com diferentes tamanhos de palavras código mais a combinação *Uncoded*) com tamanho de 4 bits e, seguindo a estrutura mostrada na Tabela 3.6.

Tabela 3.6 – Códigos para o decodificar identificar a combinação de folhas utilizada.

	Código do Sinalizador		Código do Sinalizador
<i>Uncoded</i>	0000	Comb. #7	0111
Comb. #1	0001	Comb. #8	1000
Comb. #2	0010	Comb. #9	1001
Comb. #3	0011	Comb. #10	1010
Comb. #4	0100	Comb. #11	1011
Comb. #5	0101	Comb. #12	1100
Comb. #6	0110		

Todo esse processo permitiu ganhos na compressão do áudio com cerca de 2% mas que não aparecem nos resultados devido a não consideração desse estágio no sistema RDC.

### 3.4.3 Quantização Direta dos Coeficientes (Método B)

No método B, a codificação dos coeficientes preservados é simplesmente feita por um processo de quantização uniforme (do tipo *mid-tread* e simétrico [29]) onde são considerados 16 bits por coeficiente. Para diminuir o erro de quantização também é considerado neste processo o escalamento da faixa dinâmica de cada sub-banda para valores entre -1 e 1 através da utilização de dois fatores de escala: um máximo e um mínimo. Nesse caso, a quantização escalar de um coeficiente  $d_j[p]$  pode ser expressa através da Fig. 3.19.

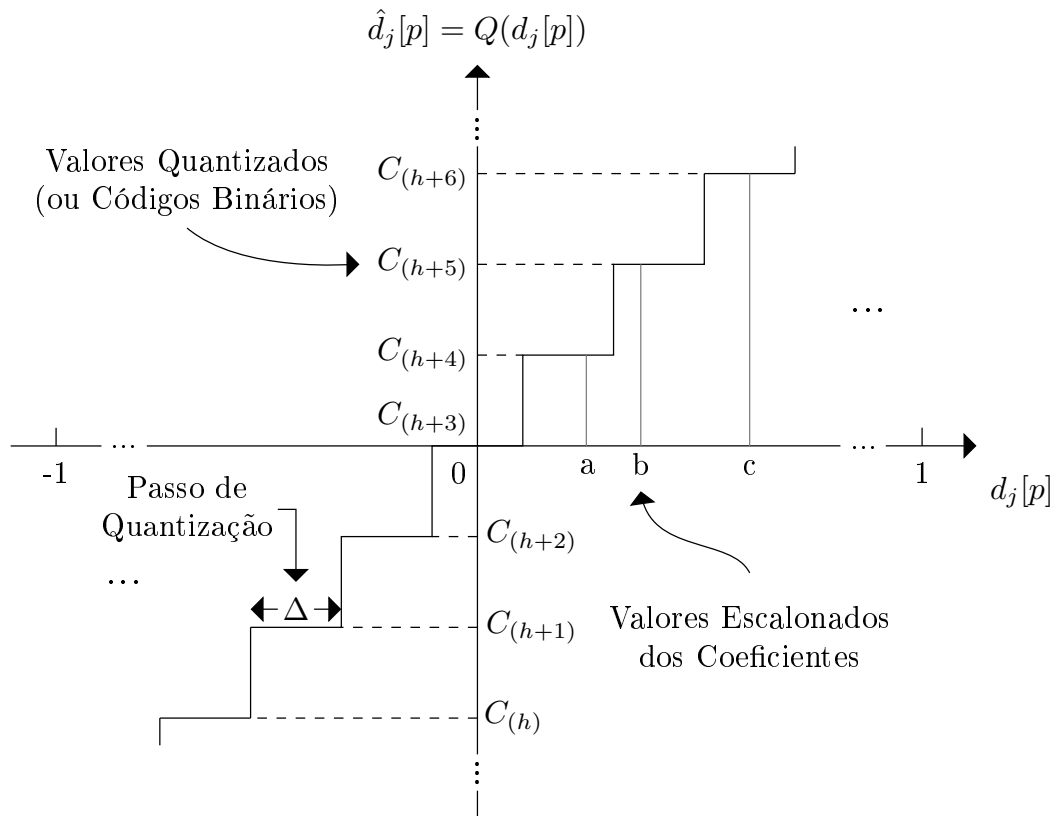


Figura 3.19 – Quantização utilizada na codificação dos coeficientes pelo método B.

A utilização dos fatores de escala se justifica devido ao fato de que os coeficientes nas sub-bandas apresentam uma faixa dinâmica bastante grande devido à resolução utilizada. Por mais que a representação destes por 16 bits forneça 65.536 níveis de quantização, a normalização contribuiu no desempenho do codificador aumentando a SNR do sinal codificado sem um considerável custo na compressão.

O fator de escala mínimo é considerado como o próprio valor do limiar de audição para a sub-banda sendo codificada, e este não precisa ser armazenado pois é de conheci-

mento do decodificador. Por outro lado, o fator de escala máximo é encontrado como sendo o máximo valor absoluto dos coeficientes daquela sub-banda sendo este posteriormente quantizado uniformemente com 7 bits no conteúdo da sub-banda.

No processo de quantização do fator de escala máximo é estipulado um valor máximo de 30 para o máximo valor possível que pode atingir um coeficiente dentro de qualquer sub-banda. Este valor foi encontrado após o monitoramento das sub-bandas resultantes de um conjunto de 100 amostras de áudio. O mesmo conjunto de amostras que será apresentado no Capítulo 4 de resultados.

Neste tipo de codificação, os valores das posições dos coeficientes preservados são diretamente codificadas com  $v$  bits em sua forma original, isto é, sem ser em qualquer forma ordenada como a no método A. Além disso, a quantidade de coeficientes preservados é representada unicamente através de um sinalizador com  $(v + 1)$  bits e não duas vezes como no outro método.

### 3.5 Sistema Adaptativo de Controle de Taxa Versus Distorção

Conforme mencionado nas seções anteriores, o codificador proposto tem um sistema adaptativo de controle de taxa versus distorção (RDC) para produzir uma CBR aproximada em sua saída. Esta, por sua vez, não é totalmente constante devido ao estágio de codificação de entropia (que fornece ganhos de compressão variados) e o fato que alguns bits ainda sobram para serem usados devido a própria estrutura do sistema.

Para alcançar a CBR, o sistema adaptativo RDC limita o consumo de bits (CB) por bloco de áudio através de um *buffer*<sup>4</sup> calculado em função da taxa de bits desejada. Dado os valores desse *buffer* e dos CB's produzidos pelos estágios de codificação, o sistema ajusta alguns parâmetros de codificação através de um laço de repetição até que o sinal codificado esteja na taxa de bits especificada ou a um valor um pouco abaixo.

Os parâmetros ajustados no sistema são o fator de ganho  $\theta$  do limiar auditivo, o fator de peso  $\lambda$  da seleção do método de codificação e o número de parâmetros polinomiais  $\delta$  considerados na aproximação polinomial. Os valores finais encontrados destes parâmetros, com exceção do fator de peso  $\lambda$ , são armazenados no cabeçalho dos blocos de áudio através de 10 e 4 bits respectivamente, em razão de suas faixas de valores.

O RDC corresponde ao retângulo tracejado no fluxograma geral do codificador da Fig. 3.1 e é um processo auxiliar ao codificador. Ele é representado em forma de fluxograma na Fig. 3.20.

<sup>4</sup> Nesse caso, o termo *buffer* é utilizado para indicar um espaço de armazenamento limitado.

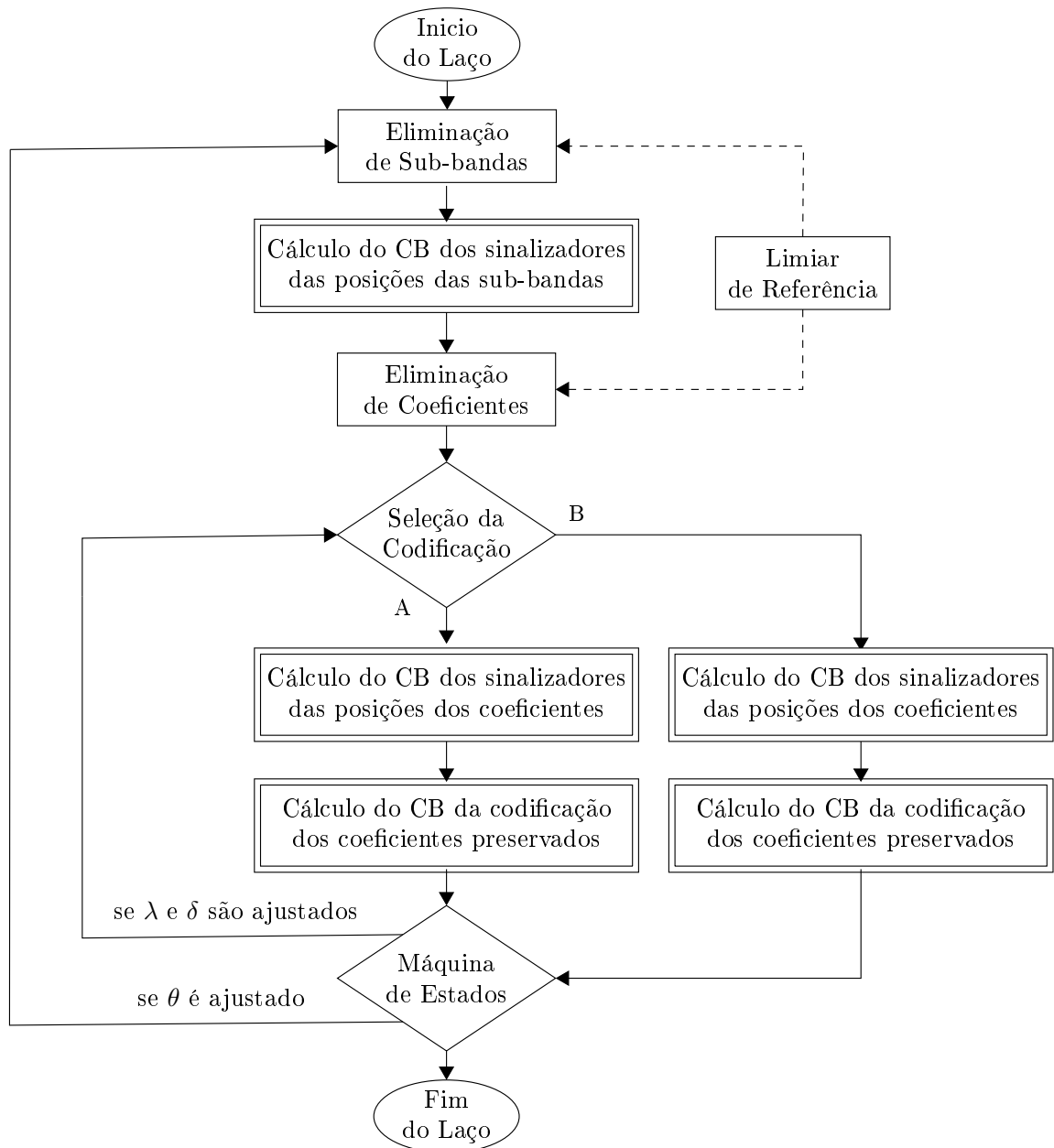


Figura 3.20 – Fluxograma do RDC incorporado ao codificador proposto.

Observa-se que, a cada laço executa-se o processo de eliminação de sub-bandas, eliminação de coeficientes e a seleção da codificação, calculando após cada um desses o CB resultante. Desde que a maioria dos estágios do codificador tem códigos de comprimento fixo (com exceção do código de entropia), os CB's são rapidamente calculados através do número de sub-bandas e coeficientes preservados e o método de codificação escolhido em cada sub-banda. Deste modo, não se torna necessário executar os processos mais custosos computacionalmente como, a aproximação polinomial, nesse laço.

É visto também que o valor final do CB é enviado a uma máquina de estados que tem como saída uma realimentação dos sinais ajustados ou o fim do laço. A função dessa máquina é armazenar os estados atuais, isto é, os valores dos parâmetros e, decidir qual

parâmetro será ajustado, como será ajustado, assim como, direcionar a realimentação em função dos estágios pertinentes ao parâmetro ajustado.

O diagrama da máquina de estados pode ser visto na Fig. 3.21 e foi construído para priorizar algumas mudanças de parâmetros sobre outros e a contribuir com a procura rápida pelos parâmetros finais através de resoluções variáveis do fator de ganho  $\theta$ . Tanto a estrutura como o estado inicial definido como  $\theta = 0.1$ ,  $\delta = 7$  e  $\lambda = 1$  foram estabelecidos de acordo com o desempenho apresentado em testes empíricos.

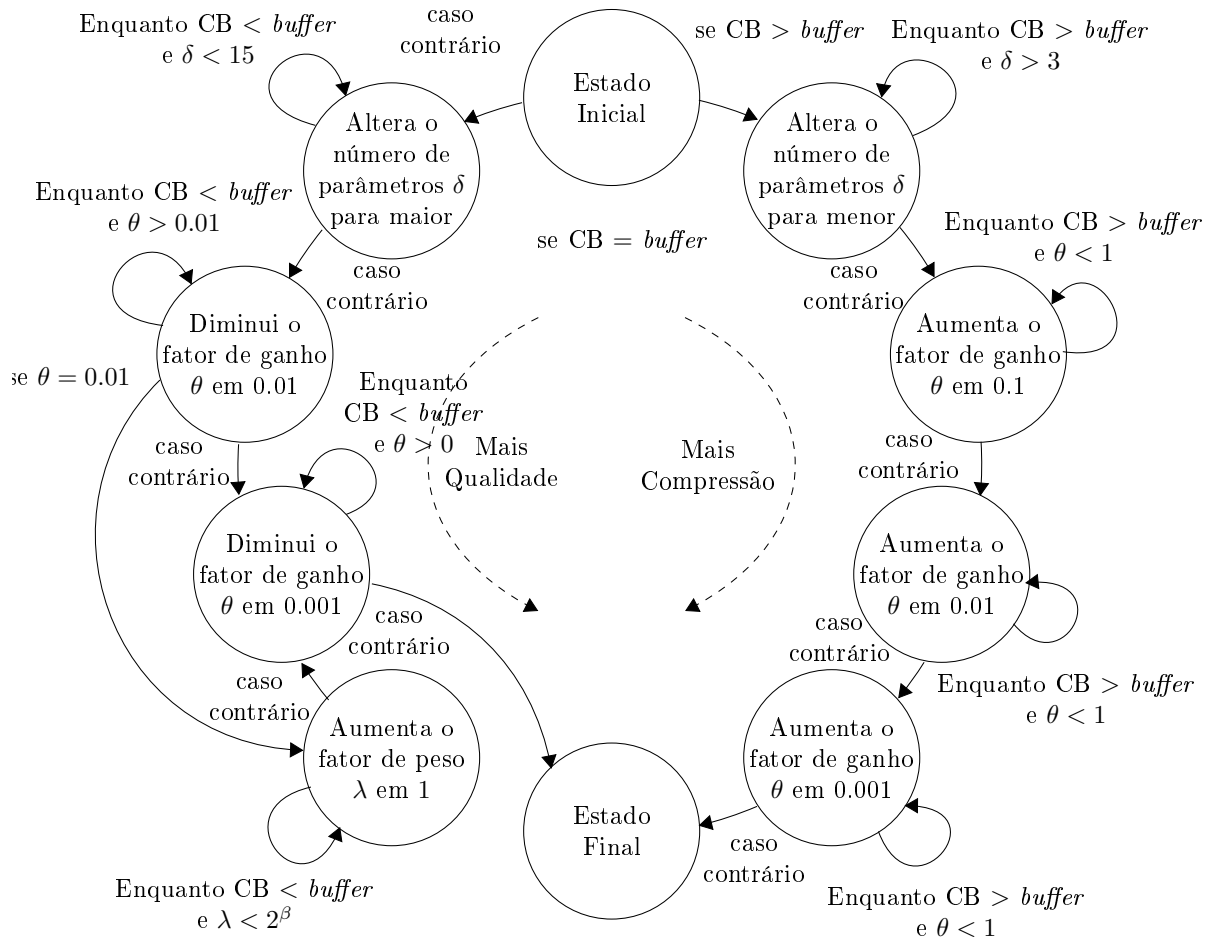


Figura 3.21 – Diagrama de estados do sistema RDC.

Nota-se que, o diagrama da máquina de estados forma um círculo incompleto onde o primeiro resultado de CB indica se a máquina de estados muda para estados que almejam mais qualidade do sinal (ou seja, diminuem a distorção introduzida pela codificação) ou, muda para estados que almejam mais compressão (ou diminuem a taxa de bits na saída).

Para melhorar a qualidade do sinal, a máquina transita para estados que aumentam o número de coeficientes  $\delta$  da aproximação polinomial, melhorando assim a estimação e diminuindo a distorção do sinal. Similarmente, transita para estados que diminuem o fator de ganho  $\theta$  sobre o limiar auditivo para diminuir a distorção na forma de onda causada pela eliminação de sub-bandas e coeficientes. Por fim, o fator de peso  $\lambda$  poder

ser aumentado em um estado para que a escolha de codificação pelo método B, dado pela quantização direta, seja mais predominante devido este garantir um menor erro do que a aproximação polinomial do método A ao custo de maior taxa.

De maneira antagônica, para melhorar a compressão do sinal, o número de coeficientes  $\delta$  da aproximação polinomial é diminuído para que assim haja maior compressão dos dados através de salvar menos parâmetros polinomiais, assim como o fator de ganho  $\theta$  do limiar de referência é aumentado para a eliminação de mais sub-bandas e coeficientes menos expressivos.

Para cada estado na Fig. 3.21, com exceção do estado inicial e final (que terminam ou iniciam a máquina), a máquina de estados repete aquele estado até o que resultado do CB seja maior (ou menor, para os estados que visam maior compressão) do que o *buffer* estipulado ou então que a faixa de valores daquele parâmetro tenha terminado. Quando a transição de estados acontece devido ao estouro do *buffer*, a máquina de estados volta o valor do parâmetro para seu valor anterior para evitar que uma taxa de bits maior do que o limite estipulado seja produzida.

Após o término do laço de controle de taxa versus distorção, o algoritmo retorna até o estágio de eliminação de sub-bandas para assim definitivamente codificar os blocos de áudio utilizando os parâmetros finais encontrados.

O codificador proposto pode trabalhar sem o sistema adaptativo RDC se desejado. Para isso, são utilizados os parâmetros iniciais da máquina de estados para a codificação de todos os blocos de áudio. Neste caso, a taxa de bits na saída é do tipo VBR e muda conforme o tipo de som a ser codificado.

## 3.6 Formato do *bitstream* e Processo de Decodificação

Para clarificar o formato do *bitstream*<sup>5</sup> da saída do codificador proposto, o mesmo é ilustrado na Fig. 3.23 através dos trechos de bits de cada estágio.

Nota-se que, o cabeçalho do bloco codificado é composto pelos dados dos parâmetros finais do sistema RDC, os sinalizadores do processo de eliminação de sub-bandas e os sinalizadores dos métodos de codificação escolhidos para cada sub-banda. Observa-se também que, os trechos de dados pertinentes ao conteúdo de cada sub-banda está em função do método de codificação escolhido e segue o padrão de primeiro enviar os sinalizadores do processo de eliminação dos coeficientes e depois as palavras códigos pertinentes as codificações dos valores dos coeficientes.

<sup>5</sup> Traduzido para o português como fluxo de bits, é uma série temporal de bits.



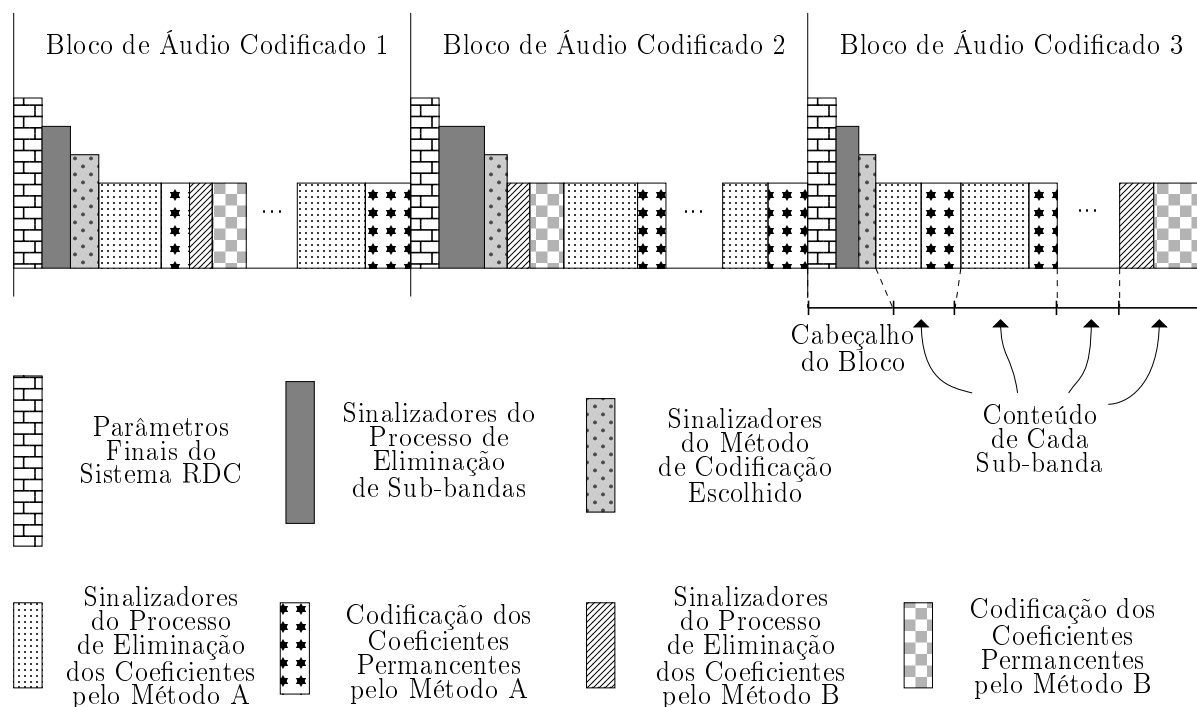


Figura 3.22 – O formato do *bitstream* para cada bloco de áudio codificado.

Cada bloco de áudio codificado tem mesmo tamanho em bits e, com exceção dos parâmetros finais do sistema RDC, o consumo de bits de cada trecho de dados varia em função do áudio. A título de curiosidade, através de testes com algumas amostras de áudio, nota-se que o maior consumo de bits do codificador está presente nos sinalizadores das posições dos coeficientes que são gerados pelo processo de eliminação de coeficientes. Por fim, um detalhamento do *bitstream* para cada trecho de dados é mostrado na Fig 3.23.

Como usual nos codificadores fonte, o processo de decodificação do áudio é mais simples do que o de codificação devido, neste caso, à ausência do sistema adaptativo RDC e também da do estágio de seleção do método de codificação. De maneira resumida, o decodificador lê todos os parâmetros do cabeçalho do bloco codificado e reconstrói, sub-banda a sub-banda, o sinal de áudio original (com algumas aproximações e perdas). Na Fig. 3.24 é mostrado o fluxograma do decodificador para recuperação do sinal codificado.

Nota-se que a partir da leitura do cabeçalho do bloco codificado, o decodificador ajusta alguns estágios e também indica com qual método as sub-bandas estão codificadas.

Caso estejam codificadas pelo método A, a reconstrução da sub-banda é iniciada pela leitura da quantidade de coeficientes acima e abaixo do limiar de audição. Em seguida, há decodificação entrópica das posições dos coeficientes na forma ordenada. Para cada posição de coeficiente recuperada é atribuído um valor de coeficiente através da curva polinomial formada pelos parâmetros lidos. Valores nulos são dados aos coeficientes sem posições salvas. Por fim, ocorre a reordenação dos coeficientes em suas posições originais.

<b>Cabeçalho do Bloco Codificado</b>	
<b>Parâmetros Finais do Sistema RDC:</b>	
<ul style="list-style-type: none"> <li>• O fator de ganho <math>\theta</math> (10 bits).</li> <li>• O número de parâmetros polinomiais <math>\delta</math> (4 bits).</li> </ul>	
<b>Sinalizadores do Processo de Eliminação de Sub-bandas:</b>	
<ul style="list-style-type: none"> <li>• A quantidade de sub-bandas permanentes ( <math>(u + 1)</math> bits).</li> <li>• As posições das sub-bandas permanentes ou eliminadas na árvore WPD (<math>u</math> bits).</li> </ul>	
<b>Sinalizadores do Método de Codificação Escolhido:</b>	
<ul style="list-style-type: none"> <li>• Lista indicando o método escolhido em cada sub-banda (1 bits por sub-banda).</li> </ul>	
<b>Conteúdo de Cada Sub-banda</b>	
Se escolhido o <u>Método A</u> :	
<b>Sinalizadores do Processo de Eliminação dos Coeficientes:</b>	
<ul style="list-style-type: none"> <li>• A quantidade de coeficientes permanentes abaixo e acima do limiar de referência (duas vezes em <math>(v + 1)</math> bits).</li> <li>• As palavras do código de entropia das posições dos coeficientes na ordem em que estão ordenados (indeterminado a quantidade de bits por palavra código).</li> </ul>	
<b>Codificação dos Coeficientes Permanentes:</b>	
<ul style="list-style-type: none"> <li>• Os parâmetros polinomiais da curva aproximada dos coeficientes (32 bits por parâmetro).</li> </ul>	
Se escolhido o <u>Método B</u> :	
<b>Sinalizadores do Processo de Eliminação dos Coeficientes:</b>	
<ul style="list-style-type: none"> <li>• A quantidade de coeficientes permanentes (<math>v</math> bits).</li> <li>• As posições dos coeficientes quantizadas na ordem original (<math>v</math> bits por coeficiente).</li> </ul>	
<b>Codificação dos Coeficientes Permanentes:</b>	
<ul style="list-style-type: none"> <li>• O fator de escala máximo (7 bits).</li> <li>• As palavras código dos coeficientes quantizados (16 bits por coeficiente).</li> </ul>	

Figura 3.23 – Detalhamento dos dados e suas representações em cada trecho do *bitstream*.

Similarmente ao método A, a reconstrução da sub-banda pelo método B é iniciada pela leitura da quantidade de coeficientes. Em seguida, as posições são decodificadas (e já estão na sua forma original). Para cada posição é também atribuído o valor desquantizado dos coeficientes, utilizando o dequantizador ajustado pelos fatores de escala máximo e mínimo. Nos demais coeficientes sem posições salvas são assumidos valores nulos.

Finalmente, as sub-bandas são inseridas em seus devidos lugares na árvore e, valores

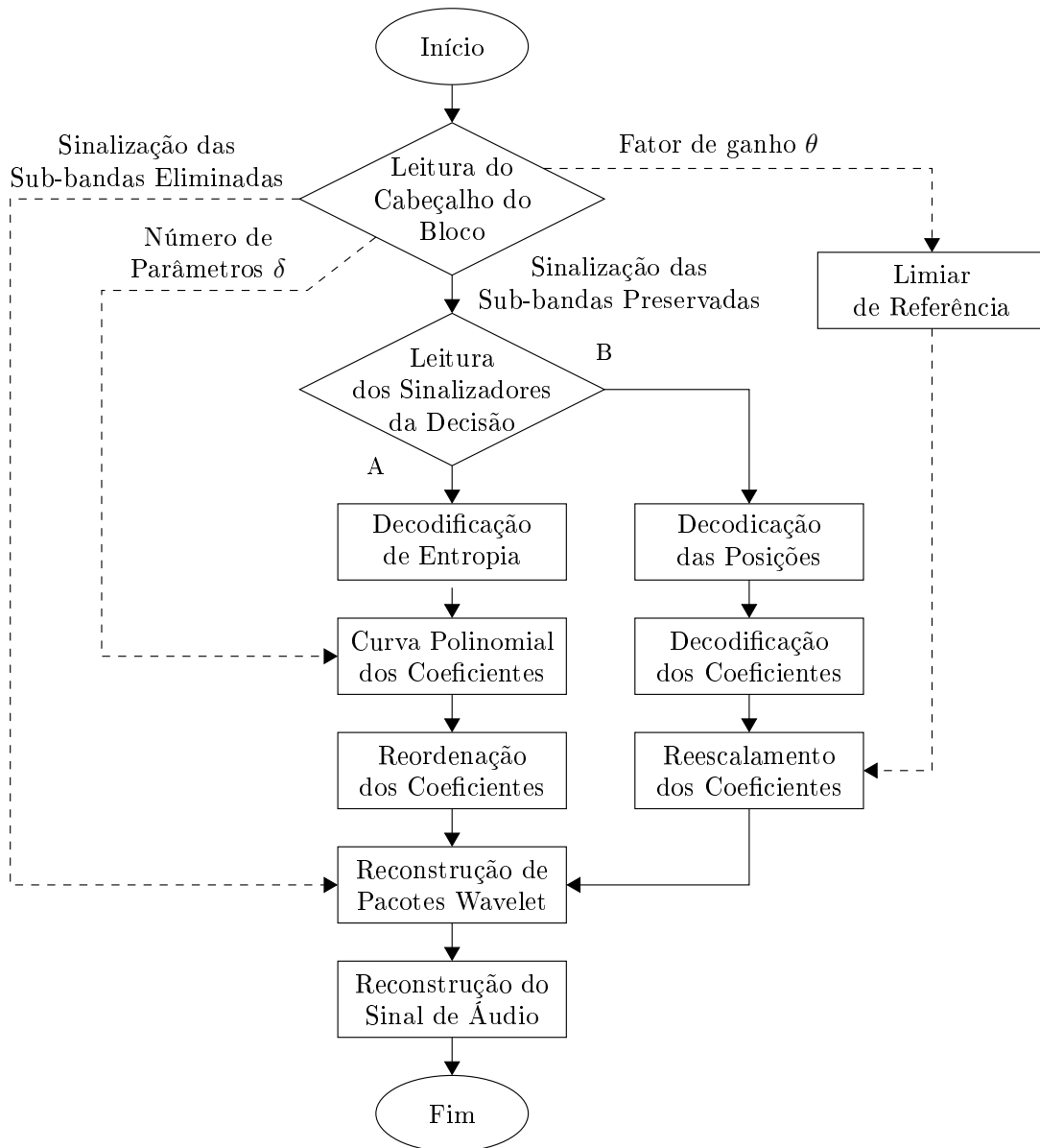


Figura 3.24 – Fluxograma do processo de decodificação.

nulos são dados as sub-bandas indicadas como eliminadas. Após o preenchimento da árvore, ocorre a sintetização dos pacotes wavelet (ou seja, a inversa da WPD) para formar o bloco de áudio reconstruído. Os blocos tem então suas margens excedentes cortadas e estes são todos concatenados para forma o sinal de áudio decodificado.

---

## Capítulo 4

---

---

# Resultados

---

Neste capítulo descreve-se os resultados deste trabalho de dissertação. Esses resultados são divididos em duas partes: os resultados pertinentes ao estudo dos hiper parâmetros na Seção 4.1 e os resultados dos testes de desempenho do codificador proposto em relação a outros codificadores na Seção 4.2.

Os resultados foram obtidos simulando o sistema no *software* MATLAB [51]. As amostras de áudio utilizadas são fragmentos de 10 segundos no formato de CD (*Compact Disk*) convencionais, ou seja, formato de arquivo WAV, com frequência de amostragem de 44,1 kHz, resolução de 16 bits por amostra e sistema de canal mono ou estéreo.

O conjunto de amostras de áudio utilizado contém trechos de sons provenientes de músicas de vários gêneros como pop, rock, rap e sons sintetizados como as músicas eletrônicas entre outros. Além disso, as amostras contém sons originados de instrumentos solos como piano, gaita, violino, saxofone e gaita de fole.

Além dos testes apresentados neste capítulo, outras avaliações isoladas estão presentes na descrição do codificador proposto no Capítulo 3; são elas: o número de momentos nulos dos filtros da WPD; as aproximações do limiar auditivo; o fator de peso e função de decisão dos métodos de codificação; os polinômios escolhidos para a aproximação polinomial e a ponderação dada na estimação pelo MMQP; as outras técnicas de codificação de entropia; os fatores de escala na quantização direta dos coeficientes; a estrutura da máquina de estados e de seus valores no estado inicial. Esses testes não são mostrados em detalhes mas seus resultados constituem a metodologia do codificador proposto.

### 4.1 Estudo dos Hiper Parâmetros

Através dos chamados hiper parâmetros  $u$  e  $v$ , relacionados respectivamente ao nível de profundidade da decomposição do bloco de áudio e ao tamanho das sub-bandas *wavelets* geradas, o projeto do codificador proposto pode ser facilmente adaptado para outras configurações. Nesta seção, é discutido o desempenho do codificador em relação a essas configurações. Para isso, são executadas a compressão de 100 amostras de áudio para taxas de bits comuns de 32 a 192 kbps e, tomado a média da relação sinal-ruído (SNR) das amostras codificadas com cada configuração.

Devido as muitas possibilidades de arranjo destes hiper parâmetros, a discussão usa uma configuração padrão com  $u = 8$  e  $v = 8$  e, em seguida, altera apenas um dos parâmetros por vez para estudá-lo. Além disso, como mostrado em (3.1), estes parâmetros definem o tamanho do bloco de áudio. Deste modo, quanto maiores os parâmetros, maior o tamanho do bloco de áudio, sendo o limite superior equivalente ao tamanho do sinal de áudio. Porém, tamanhos pequenos de sub-bandas implicam em maior custo computacional devido ao grande número de operações de aproximação polinomial. Portanto, a faixa de valores escolhida para essa discussão é de  $u$  e  $v$  de 5 até 10, o que implica em sub-bandas com largura de 689 até 22 Hz e com número de coeficientes de 32 até 1024.

O desempenho do codificador proposto em relação a mudança do hiper parâmetro  $u$ , com  $v$  fixo em 8, é apresentado na Fig. 4.1.

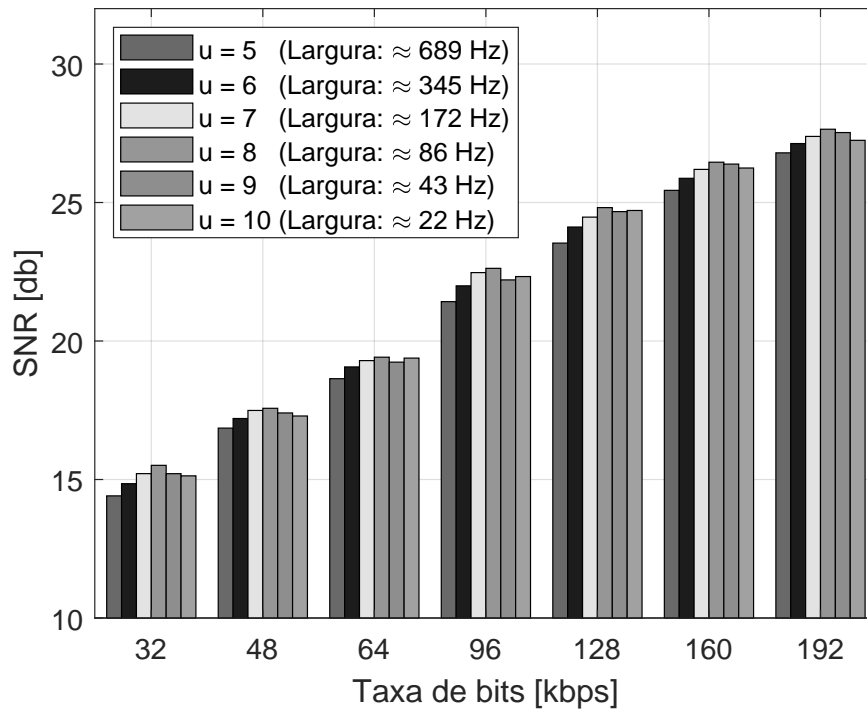


Figura 4.1 – Desempenho do codificador para diferentes níveis de profundidade  $u$ .

Nota-se que o nível de profundidade  $u = 8$  apresenta uma SNR média ligeiramente maior que os outros níveis. Os níveis mais baixos do que o  $u = 8$  apresentam também uma serem mais baixos do que os níveis mais altos.

A primeira suposição para queda de desempenho tanto acima quanto abaixo de  $u = 8$  é a respeito da implementação da codificador. A implementação é focada na leitura e escritas dos arquivos no formato original, codificado e decodificado. Dessa forma, às funções de chamadas para essas operações (feitas pela linguagem de programação utilizada) são gerenciadas pelo sistema de arquivos do computador. Esse por sua vez, tem um padrão de escrita baseada em bytes e quando a soma dos bits não é múltipla de um byte,

o sistema de arquivos completa a sequência de bits com bits nulos.

Esse comportamento do sistema de arquivos supõe-se ser independente da linguagem de programação utilizada. Dessa forma, essa situação foi levada em conta na codificação e decodificação, através de funções de auxílio para converter a sequência em múltiplos de bytes e através de precauções nos cálculos do consumo de bits considerando esse fenômeno. Contudo, supõe-se que a quantidade de vezes que esse fenômeno ocorre é maior quando o nível de profundidade é diferente de oito visto que as posições das sub-bandas podem facilmente ser representadas uma à uma por bytes.

Existe uma suposição sobre a menor eficiência dos níveis mais baixos de profundidade. Níveis mais baixos impactam diretamente na aproximação do limiar de audição pelas sub-bandas. Isso porque sub-bandas com maior largura de frequência discretizam mais grosseiramente as curvas do limiar e, desse modo, podem gerar ou não mais distorção na forma de onda do sinal codificado. Na Fig 4.2 é ilustrado a resolução final do limiar para cada nível de profundidade.

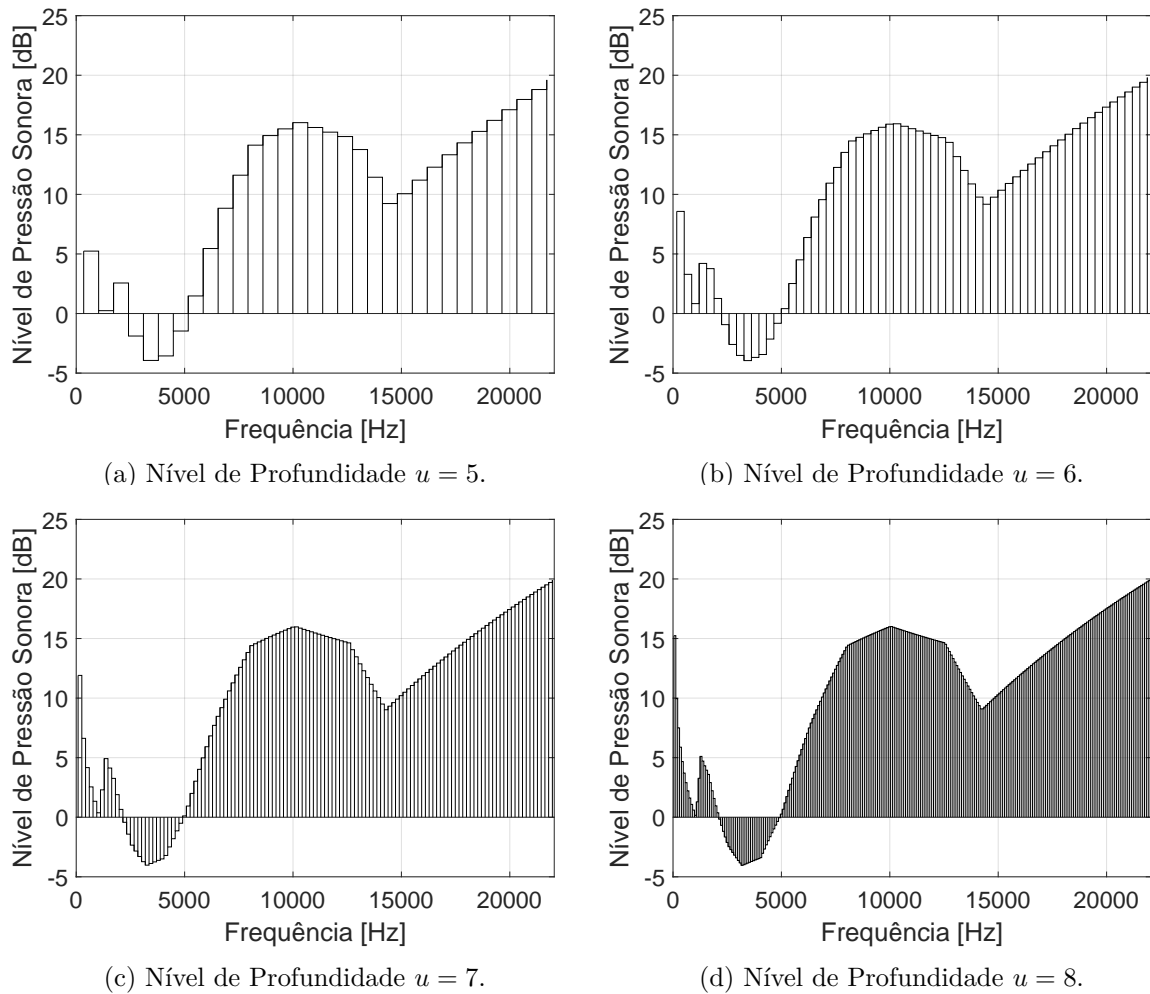


Figura 4.2 – Resolução do limiar de audição nos diferentes níveis de profundidade.

Outra suposição é que que níveis mais altos de profundidade geram maior quantidade de sub-bandas e, conseqüentemente, um custo maior nos sinalizadores de quais sub-bandas foram eliminadas ou preservadas. Esse custo maior se transformada em qualidade menor através do controle de taxa versus distorção feito pelo sistema adaptativo do codificador proposto.

A Fig. 4.3 mostra o desempenho do codificador para diferentes valores do hiper parâmetro  $v$  que representa à potência de dois do tamanho das sub-bandas.

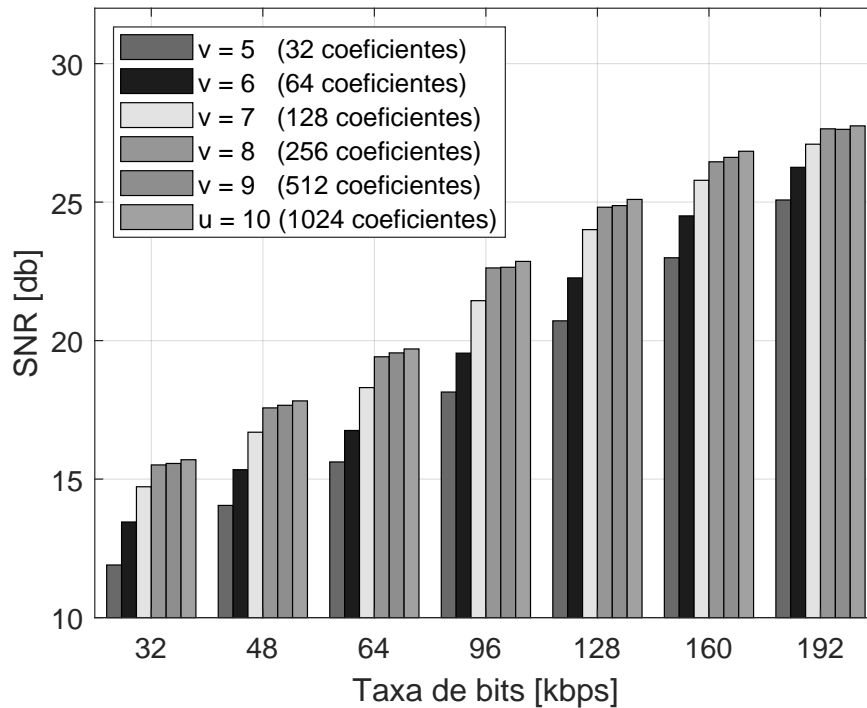


Figura 4.3 – Desempenho do codificador para diferentes tamanhos de sub-bandas  $v$ .

Observa-se que, quanto maior o tamanho da sub-banda maior o desempenho do codificador proposto em relação a SNR média.

Supõe-se que, o aumento ocorre devido aos melhores ganhos de compressão no estágio de aproximação polinomial ao usar sub-bandas maiores. Contudo, de maneira similar a primeira suposição da discussão anterior, é provável que o desempenho também esteja sendo afetado pela regra do sistema de arquivos de sempre completar a sequência de bits a ser escrita no arquivo com zeros, caso essa não seja múltiplo de bytes. Isso poderia justificar a diminuição do aumento de desempenho depois das sub-bandas com 256 coeficientes visto que a posição dos coeficientes nesse tamanho podem ser representados por bytes.

Pode-se afirmar que o melhor nível de profundidade (ou melhor hiper parâmetro  $u$ ), na medida de SNR média das 100 amostras usadas, é de 8. Enquanto que, o melhor tamanho de sub-banda, neste mesmo critério, seria quanto maior possível, dado que sempre é verificado um aumento de desempenho quanto maior o tamanho da sub-banda. Entre-

tanto, sub-bandas com maior número de coeficientes implicam em blocos de áudio com maior número de amostras e consequentemente aumenta-se o atraso no processamento. Dessa forma, tornam-se apropriadas as sub-bandas com 256 coeficientes já que os ganhos apresentados após esse tamanho são somente ligeiramente maiores.

## 4.2 Teste de Desempenho

Nesta seção apresenta-se os resultados de desempenho do codificador proposto em relação a outros codificadores de áudio. Para isso, o codificador é configurado com os hiper parâmetros  $u = 8$  e  $v = 8$ , ditos como melhores na Seção anterior. São escolhidos para essa comparação os codificadores MP3, AAC e Vorbis [4, 9, 11] os quais foram testados usando ferramenta de conversão grátis Freemake Audio Converter [52].

Semelhante ao estudo de hiper parâmetros, o primeiro teste de comparação é feito usando 100 amostras de áudio que são codificadas e decodificadas e posteriormente comparadas com o sinal original em relação a SNR. Esse teste tem um pequeno adicional: são calculados, além da média, os desvios padrões das SNR's obtidas. Na Fig. 4.4 é possível ver os resultados dessa comparação para as taxas comuns de 32 a 196 kbps.

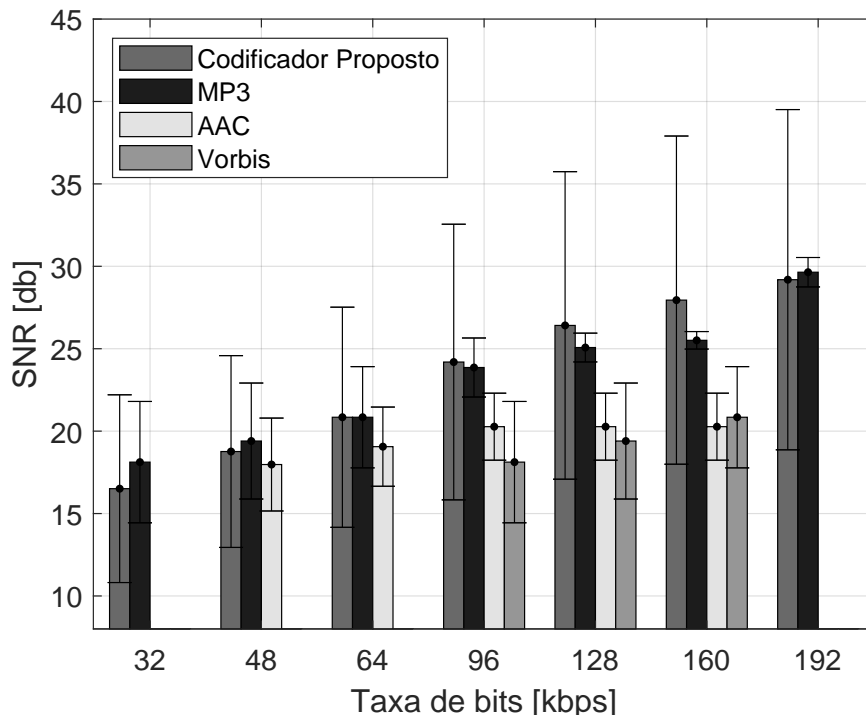


Figura 4.4 – Comparação de desempenho de 100 sinais de áudio de 32 a 196 kbps.

Nota-se que o codificador proposto atinge uma SNR média maior (de 1 à 3 dB maior em relação ao segundo) que os outros codificadores para as taxas de 96, 128 e 160 kbps.



Observa-se também que os resultados do codificador proposto mostram um desvio padrão maior em comparação com os outros codificadores para todas as taxas.

O valor alto de desvio padrão indica que o desempenho do codificador proposto em relação a SNR está variando conforme o tipo de áudio a ser codificado. Em vista disso, no segundo teste, analisa-se o desempenho do codificador para diferentes tipos de áudio. Nesse caso, o teste é feito com 10 amostras de áudio por tipo, sendo os tipos agrupados por gênero ou instrumento musical. O resultado desse teste para as taxas de 32, 64, 128 e 192 kbps é exibido na Fig. 4.5.

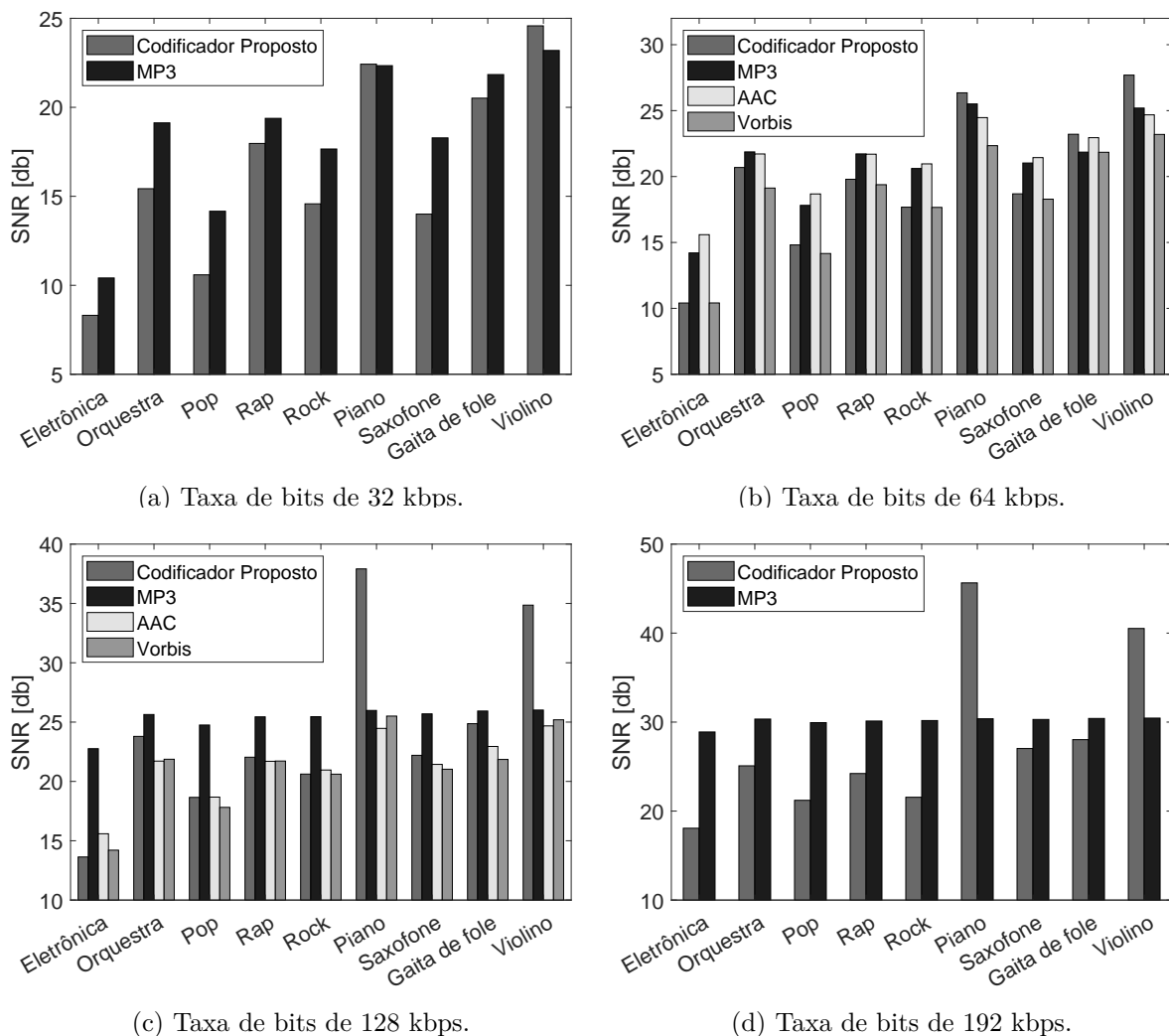


Figura 4.5 – Comparação de SNR usando amostras de áudio por gênero.

Verifica-se que, usualmente o codificador proposto mostra maior SNR média nos áudios produzidos por pianos e violinos e que, este se sobrepõe excessivamente aos demais codificadores nas taxas de 128 e 192 kbps (pelo menos 10 dB a mais). Nos demais tipos de áudio, o codificador se encontra aproximadamente similar aos outros codificadores com exceção de algumas diferenças altas no gênero de música eletrônica, pop e rock, nas taxas de 128 e 192 kbps em relação ao MP3.

Para avaliar o desempenho do codificador proposto em termos de qualidade de áudio perceptiva, realizou-se o teste do codificador com dois modelos perceptuais de avaliação objetiva de qualidade de áudio, o PEAQ e PEMO-Q citados na Seção (2.3). O teste é realizado da mesma forma do primeiro teste, ou seja, sem o agrupados das amostras de áudio por tipo.

A implementação do PEAQ utilizada nos testes foi proposta por Kabal da Universidade McGill [53] qual contém uma versão em código MATLAB. Nesse caso, o modelo PEAQ restringe os sinais de entrada a serem amostrados a 48 kHz e que os sinais de referência e teste estejam alinhados no tempo. Dessa forma, uma conversão de amostragem e um processo de correção de atraso são utilizados em todas as amostras de áudio decodificadas dos codificadores. O resultado do teste é dado pela ODG média das 100 amostras de áudio, conforme Fig. 4.6.

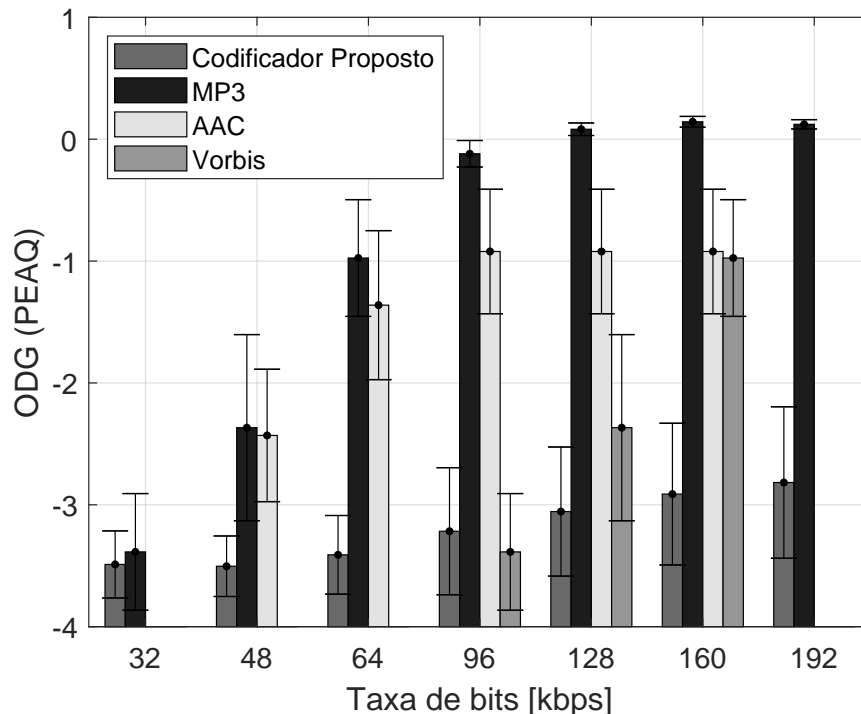


Figura 4.6 – Comparação de desempenho de 100 amostras de áudio de 32 a 196 kbps.

Constata-se que, nesse tipo de medição o codificador proposto apresenta um desempenho inferior em relação aos outros codificadores na maioria das taxas de bits testadas. Variando sua descrição de qualidade nas amostras de áudio como Ligeiramente Incômoda, Incômoda e Muito Incômoda.

Para consolidar os resultados do codificador proposto neste tipo de avaliação, é realizado o teste com o PEMO-Q. Neste caso, a implementação utilizada para o teste é uma levemente modificada do PEMO-Q proposto, por Huber e Kollmeier [37]. É encontrada em uma versão em MATLAB para fins acadêmicos no site da companhia alemã HörTech gGmbH [54]. Os resultados deste teste podem ser visto na Fig. 4.7.

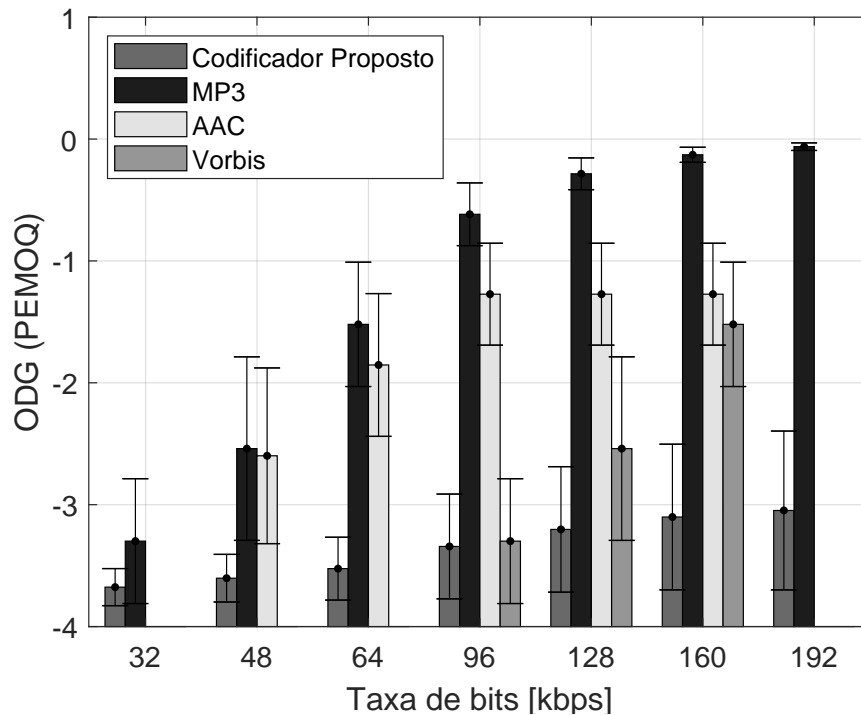


Figura 4.7 – Comparação de desempenho de 100 sinais de áudio de 32 a 196 kbps.

Similarmente ao resultado anterior, constata-se que o codificar proposto apresenta desempenho inferior em relação aos outros codificadores na maioria das taxas de bits testadas.

Por fim, considerando uma análise perceptual e pessoal do autor deste trabalho, alguns artefatos são ouvidos a baixas taxas, como 32 e 48 kbps, geralmente em amostras de áudios como os do tipo eletrônico. Cogita-se que, o principal estágio responsável pode ser a aproximação polinomial que pode produzir um ruído considerável nas baixas taxas devido aos erros de aproximação. Além disso, o estágio de eliminação de sub-bandas e coeficientes menos expressivos que idealmente deve eliminar sons inaudíveis ou muito baixos pode produzir sons de artefatos devido à aproximação do limiar auditivo por sub-bandas ou pela própria definição da curva do limiar ser criada por testes subjetivos [35].

---

## Capítulo 5

---

---

# Considerações Finais

---

### 5.1 Conclusões Finais

Este trabalho apresenta um codificador de áudio baseado na codificação dos coeficientes mais expressivos das sub-bandas wavelet do sinal. Nesse codificador, foram adaptados esquemas comuns de codificação de áudio para o domínio das sub-bandas como o modelo do limiar de audição. Foram também implementadas técnicas além do processo eliminação de sub-bandas/coeficientes menos expressivos, a aproximação polinomial e codificação de entropia para ganhos considerados de compressão.

O codificador proposto ainda conta com um sistema de controle de taxa versus distorção próprio para que o mesmo trabalhe com uma CBR aproximada na saída.

Nos estudos do codificador para diferentes configurações são indicados os melhores parâmetros a serem utilizados em relação a SNR. Os autores supõem que os resultados encontrados nesse estudo foram influenciados pela tendência do sistema de arquivos em completar as sequências de dados em bytes e que o melhor desempenho com sub-bandas de maior número de coeficientes é devido ao maior ganho de compressão do procedimento de aproximação polinomial nesses casos.

O codificador manifesta resultados competitivos com os padrões de codificação de áudio MP3, AAC e Vorbis, em relação à SNR, alcançando valores aproximados aos outros codificadores na análise de áudios em geral e, uma diferença positiva considerável para algumas amostras de áudio com sons harmônicos como os de piano e violino. Por outro lado, apresentou valores baixos em relação aos outros codificadores para as medidas objetivas de qualidade de áudio perceptual.

Conclui-se que as principais contribuições do trabalho são: o esquema detalhado de compressão de áudio baseado na ordenação e aproximação polinomial dos coeficientes mais expressivos. O código de entropia desenvolvido para as posições dos coeficientes. O sistema de controle de taxa versus distorção (RDC). Os demais componentes, são periféricos e são apresentados para fornecer uma base de teste para o esquema de compressão proposto.

## 5.2 Propostas de Trabalhos Futuros

O codificador proposto foi analisado por métodos objetivos de avaliação como a SNR e os modelos perceptuais PEAQ e PEMO-Q. Esses métodos permitiram a obtenção de resultados mais ágeis para este trabalho, porém em trabalhos futuros é desejado realizar testes de desempenho do codificador com algum método subjetivo de avaliação como por exemplo a Recomendação BS.1116 da ITU-R [39].

No trabalho é utilizado uma estrutura de árvore WPD do tipo cheia (*full binary tree*). No entanto, podem ser testadas e avaliadas outras estruturas de árvore como, por exemplo, aquelas baseadas no esquema de bandas críticas do ouvido humano do trabalho de Sinha et al. [16]. Através das sub-bandas de tamanho variado é possível acompanhar a resolução que o ouvido pode detectar ou perceber os sons.

Na literatura sobre compressão de áudio podem ser encontrados também outras técnicas perceptuais para serem modeladas nas sub-bandas wavelet como o efeito de mascaramento de sons e as bandas críticas do ouvido humano (Seção 2.2). Um exemplo do uso do efeito de mascaramento de sons para compressão de áudio utilizando sub-bandas wavelet é visto no trabalho de Kemper et al. [13].

Existe também a possibilidade de substituir a técnica de aproximação polinomial para a codificação dos coeficientes permanentes. Um exemplo seria a técnica de quantização vetorial que pode ter uma eficiência melhor do que a aproximação polinomial em relação a distorção do sinal codificado e sua compressão. Essa é bastante vista na literatura sobre compressão de dados [17, 29].

Da mesma forma, pode-se buscar outros códigos de entropia para as posições dos coeficientes permanentes ou até estudar melhorias no código atual. Isso é porque grande parte do consumo de bits do codificador é pertinente as posições que é necessária armazenar devido o processo de eliminação de coeficientes sendo este um dos principais estágios do codificador proposto.

Cogita-se também outros controles para permitir uma CBR na saída do codificador ao invés do sistema adaptativo desenvolvido neste trabalho, talvez métodos diretos de alocação de bits por exemplo. Por outro lado, também há possibilidade de trabalhar com o codificador em outros modos de compressão, como o ABR ou VBR. Desenvolvendo nestes casos sistemas inteligentes baseados no controle da qualidade do áudio codificado.

Por fim, considera-se a aplicação das técnicas de codificação utilizadas para a compressão de outros tipos de sinais além do áudio, como o de imagem, vídeo e os mais recentes, sinais de vídeos imersivos utilizados nos óculos de realidade virtual.

---

## Referências

---

- [1] DOMO: For the Good The Company, “Data Never Sleeps 6.0,” Disponível em: <http://www.domo.com/learn/data-never-sleeps-6>, Acesso em: Jan. 23, 2019. Citado na página 15.
- [2] M. Hilbert and P. López, “The World’s Technological Capacity to Store, Communicate, and Compute Information,” *Science*, vol. 332, no. 6025, pp. 60–65, 2011. Citado na página 15.
- [3] Xiph.Org Foundation, “Flac - Free Lossless Audio Codec,” Disponível em: <http://www.xiph.org/flac/>, Acesso em: Jan. 23, 2019. Citado na página 15.
- [4] ISO/IEC 13818-3:1998, “Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 3: Audio,” Standard, 1998. Citado 4 vezes nas páginas 15, 16, 17 e 80.
- [5] ISO/IEC 11172-3:1993, “Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1,5 Mbit/s – Part 3: Audio,” Standard, 1993. Citado 2 vezes nas páginas 16 e 17.
- [6] M. Xie, P. Chu, A. Taleb, and M. Briand, “ITU-T G.719: A New Low-Complexity Full-Band (20 kHz) Audio Coding Standard for High-Quality Conversational Applications,” in *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct 2009, pp. 265–268. Citado 2 vezes nas páginas 16 e 17.
- [7] U. S. A. T. S. Committee, “Digital Audio Compression (AC-3) Standard,” Standard, Doc A/52, 10 Nov. 94 - 12 Apr 95. Citado 2 vezes nas páginas 16 e 17.
- [8] Dolby Laboratories, “Dolby AC-4: Audio Delivery for Next-Generation Entertainment Services,” Technical Report, 2015. Citado 2 vezes nas páginas 16 e 17.
- [9] Xiph.Org Foundation, “Vorbis Audio Compression,” Disponível em: <http://www.xiph.org/vorbis/>, Acesso em: Jan. 23, 2019. Citado 3 vezes nas páginas 16, 17 e 80.
- [10] —, “Opus Interactive Audio Codec,” Disponível em: <http://www.opus-codec.org/>, Acesso em: Jan. 23, 2019. Citado 2 vezes nas páginas 16 e 17.

- [11] ISO/IEC 13818-7:2004, “Information Technology — Generic Coding of Moving Pictures and Associated Audio Information —Part 7: Advanced Audio Coding (AAC),” Standard, 1993. Citado 2 vezes nas páginas 16 e 80.
- [12] R. Sharma, S. V. Singh, A. K. Verma, and R. S. Tomar, “A Novel Hybrid DWPT and MDCT Based Coding Technique for Sounds of Musical Instruments,” in *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb 2017, pp. 498–502. Citado na página 17.
- [13] G. Kemper and Y. Iano, “An Audio Compression Method Based on Wavelets Sub-band Coding,” *IEEE Latin America Transactions*, vol. 9, no. 5, pp. 610–621, Sept 2011. Citado 2 vezes nas páginas 17 e 85.
- [14] K. Abid and K. Ouni, “Audio Compression Using a Munich and Cambridge Morlet Wavelet,” in *2009 First International Conference on Advances in Multimedia*, July 2009, pp. 1–5. Citado na página 17.
- [15] W. K. Dobson, J. J. Yang, K. J. Smart, and F. K. Guo, “High quality low complexity scalable wavelet audio coding,” in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Apr 1997, pp. 327–330 vol.1. Citado na página 17.
- [16] D. Sinha and A. H. Tewfik, “Low Bit Rate Transparent Audio Compression Using Adapted Wavelets,” *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3463–3479, 1993. Citado 2 vezes nas páginas 17 e 85.
- [17] K. Sayood, *Introduction to Data Compression*, 4th ed. Newnes, 2012. Citado 4 vezes nas páginas 17, 20, 62 e 85.
- [18] D. Salomon, *Data Compression: The Complete Reference*, 4th ed. Springer Science & Business Media, 2007. Citado 4 vezes nas páginas 17, 20, 22 e 26.
- [19] M. V. J. Kovacevic, “Wavelets and Subband Coding,” 1995. Citado na página 17.
- [20] A. L. Luvizotto, “Modelos de Representação de Sinais Musicais via Transformada Wavelets,” Tese de Mestrado, Faculdade de Engenharia Elétrica e de Computação - UNICAMP, 2007. Citado na página 18.
- [21] R. Fabbri and R. S. Mendes, “Compressão de Áudio via Wavelets, Aproximações Polinomiais e Permutações,” *II EADCA*, 2009. Citado na página 18.
- [22] C. E. Shannon, “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948. Citado na página 20.

- [23] T. M. Cover e J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2012. Citado na página 20.
- [24] D. Salomon, *Variable-length Codes for Data Compression*, 1st ed. Springer Science & Business Media, 2007. Citado na página 20.
- [25] I. M. Pu, *Fundamental data compression*. Butterworth-Heinemann, 2005. Citado na página 20.
- [26] P. D. Johnson Jr, G. A. Harris, and D. Hankerson, *Introduction to information theory and data compression*. Chapman and Hall/CRC, 2003. Citado na página 20.
- [27] P. C. Y. Kamisetty Ramam Rao, *The Transform and Data Compression Handbook*, 1st ed. CRC Press, 2000. Citado 2 vezes nas páginas 20 e 21.
- [28] S. Golomb, “Run-length encodings (corresp.),” *IEEE transactions on information theory*, vol. 12, no. 3, pp. 399–401, 1966. Citado na página 22.
- [29] R. M. G. Allen Gersho, *Vector Quantization and Signal Compression*, 1st ed. Springer Science & Business Media, 1991. Citado 3 vezes nas páginas 23, 68 e 85.
- [30] B. C. J. Moore, *An Introduction to the Psychology of Hearing*, 6th ed. BRILL, 2012. Citado 3 vezes nas páginas 24, 25 e 30.
- [31] R. E. G. Marina Bosi, *Introduction to Digital Audio Coding and Standards*, 1st ed. Springer Science & Business Media, 2012. Citado 2 vezes nas páginas 24 e 26.
- [32] H. Fletcher and W. A. Munson, “Loudness, Its Definition, Measurement and Calculation,” *The Journal of the Acoustical Society of America*, vol. 5, no. 2, pp. 82–108, 1933. Citado na página 25.
- [33] D. W. Robinson and R. S. Dadson, “A Re-Determination of The Equal-Loudness Relations for Pure Tones,” *British Journal of Applied Physics*, vol. 7, no. 5, p. 166, 1956. Citado na página 25.
- [34] S. Suzuki, Y. Suzuki, S. Kono, T. Sone, M. Kumagai, H. Miura, and H. Kado, “Equal-loudness level contours for pure tone under free field listening conditions (I),” *Journal of the Acoustical Society of Japan (E)*, vol. 10, no. 6, pp. 329–338, 1989. Citado na página 25.
- [35] I. 226:2003, “Acoustics – Normal Equal-Loudness-Level Contours,” Standard, 2003. Citado 3 vezes nas páginas 25, 55 e 83.
- [36] ITU-R, “Method For Objective Measurements of Perceived Audio Quality,” ITU-R, Recommendation ITU-R BS.1387-1, 2001. Citado na página 29.



- [37] R. Huber and B. Kollmeier, “PEMO-Q - A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception,” *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 6, pp. 1902–1911, 2006. Citado 2 vezes nas páginas 29 e 82.
- [38] T. Dau, D. Püschel, and A. Kohlrausch, “A Quantitative Model of the “Effective” Signal Processing in the Auditory System. I. Model structure,” *The Journal of the Acoustical Society of America*, vol. 99, no. 6, pp. 3615–3622, 1996. Citado na página 29.
- [39] ITU-R, “Methods For The Subjective Assessment of Small Impairments in Audio Systems,” ITU-R, Recommendation ITU-R BS.1116-3, 02/2015. Citado 2 vezes nas páginas 29 e 85.
- [40] H. Nyquist, “Certain Topics in Telegraph Transmission Theory,” *Proceedings of the IEEE*, vol. 47, no. 2, p. 617–644, 1928. (Reprint as classic paper in: Proc. IEEE, Vol. 90, No. 2, Feb 2002). Citado 2 vezes nas páginas 30 e 39.
- [41] C. E. Shannon, “Communication in The Presence of Noise,” *Proceedings of the IEEE*, vol. 37, no. 1, pp. 10–21, 1949. (Reprint as classic paper in: Proc. IEEE, Vol. 86, No. 2, 1998). Citado 2 vezes nas páginas 30 e 39.
- [42] Z.-N. Li, M. S. Drew, and J. Liu, *Fundamentals of Multimedia*. Texts in Computer Science, Springer, 2014. Citado na página 30.
- [43] I. McLoughlin, *Applied Speech and Audio Processing: with Matlab Examples*. Cambridge University Press, 2009. Citado 2 vezes nas páginas 31 e 32.
- [44] J. W. Gibbs, “Fourier’s series,” *Nature*, vol. 59, no. 1539, p. 606, 1899. Citado na página 32.
- [45] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999. Citado 2 vezes nas páginas 32 e 51.
- [46] S. Mallat, *A Wavelet Tour of Signal Processing*. Elsevier, 1999. Citado 8 vezes nas páginas 33, 35, 36, 38, 39, 41, 51 e 53.
- [47] I. Daubechies, *Ten Lectures on Wavelets*. Siam, 1992, vol. 61. Citado na página 36.
- [48] S. Weisberg, *Applied Linear Regression*, ser. Wiley Series in Probability and Statistics. Wiley, 2013. Citado na página 42.

- 
- [49] G. K. Wallace, “The JPEG Still Picture Compression Standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, Feb 1992. Citado na página 62.
- [50] F. Oliveira and M. Costa, “Embedded DCT Image Encoding,” in *IEEE-SBrT International Telecommunications Symposium-ITS-2002*, 2002. Citado na página 62.
- [51] MathWorks, “MATLAB, v9.2 (R2017a),” Disponível em:  
<http://www.mathworks.com/products/matlab.html>, Acesso em: Jan. 15, 2019. Citado na página 76.
- [52] Ellora Assets Corporation, “Freemake Audio Converter, v1.1.8,” Disponível em:  
[http://www.freemake.com/free\\_audio\\_converter/](http://www.freemake.com/free_audio_converter/), Acesso em: Jan. 15, 2019. Citado na página 80.
- [53] P. Kabal *et al.*, “An Examination and Interpretation of ITU-R BS. 1387: Perceptual Evaluation of Audio Quality,” *TSP Lab Technical Report, Dept. Electrical & Computer Engineering, McGill University*, pp. 1–89, 2002. Citado na página 82.
- [54] HörTech gGmbH, “PEMO-Q, v1.4.0 Free,” Disponível em:  
<http://www.hoertech.de/en/f-e-products/pemo-q.html>, Acesso em: Jan. 15, 2019. Citado na página 82.