



DISSERTAÇÃO DE MESTRADO

**Compressão de sinais ECG
utilizando DWT com quantização não-linear
e por sub-bandas**

Marcelo Adrián Campitelli

Brasília, Dezembro de 2015

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

**Compressão de sinais ECG
utilizando DWT com quantização não-linear
e por sub-bandas**

Marcelo Adrián Campitelli

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Mestre em Engenharia de Sistemas Eletrônicos e Automação*

Banca Examinadora

Prof. Francisco Assis Nascimento, ENE/UnB

Orientador

Prof. João Luiz Azevedo de Carvalho, ENE/UnB

Examinador interno

Prof. Bruno Luigi Macchiavello, CIC/UnB

Examinador externo

*A meus pais Juan Adrián Campitelli e Doris Ana Maria Morandini,
que me deram as ferramentas necessárias para ter uma boa educação,
e sempre apoiaram minha vida acadêmica. À minha parceira na vida, Carla da Silva Motta,
por me dar um apoio incondicional e me permitir poder fazer esse mestrado.*

Marcelo Adrián Campitelli

AGRADECIMENTOS

Agradeço ao meu orientador, professor Francisco Assis de Oliveira Nascimento, pelo apoio e a amizade que teve comigo durante esse trabalho.

Agradeço também ao meu co-orientador, professor Cristiano Jacques Miosso Rodrigues Mendes, pela constante presença no seguimento da minha dissertação, pela assessoria e conselhos acadêmicos, e mais ainda pela amizade.

Agradeço aos meus queridos sogros Fátima Motta e Arthur Motta, pela ajuda e carinho incondicional dado todos os dias, que permitiram que meu mestrado no Brasil fosse uma experiência excepcional.

A todos meus amigos do programa de pos-graduação de engenharia em sistemas eletrônicos e de automação (PGEA) pelos conselhos, conversas, e companheirismo, especialmente ao Gustavo ao Lucas.

Quero agradecer também à CAPES pelo financiamento desse mestrado, que ajudou e facilitou a realização dele.

FICHA CATALOGRÁFICA

CAMPITELLI, MARCELO ADRIÁN

Compressão de sinais ECG utilizando DWT com quantização não-linear e por sub-bandas [Distrito Federal] 2015.

xvi, 88 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2015).

Dissertação de Mestrado - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

- | | |
|---------------|----------------------|
| 1. Compressão | 2. Sinais biomédicos |
| 3. Wavelets | 4. ECG |
| I. ENE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

CAMPITELLI, M. A. (2015). *Compressão de sinais ECG utilizando DWT com quantização não-linear e por sub-bandas*. Dissertação de Mestrado, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 88 p.

CESSÃO DE DIREITOS

AUTOR: Marcelo Adrián Campitelli

TÍTULO: Compressão de sinais ECG utilizando DWT com quantização não-linear e por sub-bandas.

GRAU: Mestre em Engenharia de Sistemas Eletrônicos e Automação ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte dessa Dissertação de Mestrado pode ser reproduzida sem autorização por escrito dos autores.

Marcelo Adrián Campitelli

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

RESUMO

Com o desenvolvimento cada vez mais intenso da tecnologia de dispositivos biomédicos, há cada vez mais acesso aos sinais bioelétricos, o que permite grande avanço na realização de diagnósticos, planejamento de tratamentos e monitoração de pacientes. Particularmente, o eletrocardiograma (ECG) tem sido usado para muitos propósitos. Além disso, novas formas simples e de baixo custo para adquirir o ECG tem sido descobertas. Entretanto, esses avanços exigem o melhoramento dos processos de codificação do sinal de ECG, de forma a permitir seu armazenamento e transmissão eficientemente em termos de requisitos de memória e consumo de energia. Neste contexto, o presente trabalho propõe duas contribuições. Em primeiro lugar, apresenta um algoritmo de compressão de sinais ECG, utilizando transformadas wavelets, e propondo um processo de quantização novo, não encontrado na literatura. Nesse processo, a transformação é realizada utilizando a transformada discreta de wavelets (DWT) e a quantização consiste em um re-ordenamento não-linear das magnitudes dos coeficientes transformados (correção gamma) em conjunto com uma quantização por sub-bandas. A segunda contribuição consiste num estudo sistemático do desempenho das diferentes famílias de wavelets nos resultados obtidos pelo algoritmo proposto, calculando também os parâmetros ótimos de quantização para cada família wavelet utilizada. Para a análise desses métodos, foram realizados testes avaliando o desempenho do algoritmo proposto e comparando os resultados com outros métodos apresentados na literatura. Nesses testes, foram usados como referências os sinais da base de dados do Instituto de Tecnologia de Massachusetts e do Hospital Beth Israel de Boston (MIT-BIH). Uma parcela do banco de dados foi utilizada para otimizar os parâmetros de cada família wavelet no algoritmo proposto, e o desempenho final foi avaliado com todos os sinais restantes. Especificamente, para o sinal 117 do MIT-BIH, que é o sinal mais utilizado para comparar resultados na literatura, o método proposto levou a um fator de compressão (CR) de 11,40 e uma raiz da diferença média percentual (PRD) de 1,38. Demonstrou-se que o algoritmo gera melhores resultados de compressão quando comparado com a maioria dos métodos do estado-da-arte. Também se destaca a simplicidade na implementação do algoritmo em relação a outros encontrados na literatura.

ABSTRACT

With the increasing development of biomedical devices technology, there is more access to bioelectrical signals. That allows great advances in reaching diagnostics, planning treatments and monitoring patients. Particularly, the electrocardiogram (ECG) has been used for many purposes. Besides that, simple and low-cost ways to acquire the ECG have been found. Nevertheless, those advances require the improvement of the ECG signal coding processes, in a way that allows its efficient storage and transmission in terms of memory requirements and energy consumption. In this context, this dissertation proposes two contributions. Firstly, it presents an ECG signal compression algorithm, using wavelet transforms, and proposing a novel quantization process, not found in the literature. In said process, the transformation is done using the discrete wavelet transform (DWT) and the quantization consists of a non-linear re-ordering of the transformed coefficients magnitudes (gamma correction) in tandem with a sub-band quantization. The second contribution consists in a systematic study of the performance of the different wavelet families through the results obtained by the proposed algorithm, also calculating the optimum quantization parameters for each wavelet family. For the analysis of these methods, tests were done evaluating the performance of the proposed algorithm, comparing its results with other methods presented in the literature. In said tests, signals from the Massachusetts Institute of Technology and Boston's Beth Israel Hospital database (MIT-BIH) were used as reference. A part of the database was utilized to optimize the parameters of each wavelet family, and the final performance was evaluated with the remaining signals from the database. Specifically, for signal 117 of the MIT-BIH database, which is the most used signal to compare results in the literature, the proposed method led to a compression factor (CR) of 11,40 and a percentage root-mean-square difference (PRD) of 1,38. It was demonstrated that the algorithm generates better compression results when compared to the majority of state-of-the-art methods. The simplicity of the algorithm's implementation also stands out in relation to other algorithms found in the literature.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	PROBLEMATIZAÇÃO E PROPOSTA	2
1.3	OBJETIVOS	5
1.4	CONTRIBUIÇÕES	6
1.5	JUSTIFICATIVA	6
1.6	ESTRUTURA DA DISSERTAÇÃO	7
2	FUNDAMENTAÇÃO TEÓRICA	8
2.1	GERAÇÃO E CARACTERIZAÇÃO DO SINAL ECG	8
2.2	SINAIS EM DOMÍNIO CONTÍNUO E DISCRETO	10
2.3	TRANSFORMADAS LINEARES DE SINAIS EM TEMPO DISCRETO	11
2.3.1	NOTAÇÃO MATRICIAL	12
2.4	TRANSFORMADA WAVELETS	13
2.4.1	TRANSFORMADA DE WAVELETS EM DOMÍNIO CONTÍNUO	14
2.4.2	TRANSFORMADA DE WAVELETS EM DOMÍNIO DISCRETO	16
2.5	BANCO DE FILTROS MULTI-TAXA	18
2.6	COMPRESSÃO DE DADOS	19
2.6.1	COMPRESSÃO POR TRANSFORMADAS	20
2.6.2	QUANTIZAÇÃO	22
2.6.3	CODIFICAÇÃO POR ENTROPIA	24
2.7	MÉTODOS EXISTENTES DE COMPRESSÃO DE ECG	29
3	METODOLOGIA	36
3.1	DESENVOLVIMENTO DO SISTEMA DE COMPRESSÃO DE SINAIS DE ECG	36
3.1.1	TRANSFORMADA DE WAVELETS	36
3.1.2	DIVISÃO EM SUB-BANDAS	38
3.1.3	QUANTIZAÇÃO DAS SUB-BANDAS E CORREÇÃO GAMMA DOS COEFICIENTES	39
3.1.4	RLE (<i>run length encoding</i>)	44
3.1.5	CODIFICAÇÃO ARITMÉTICA	44
3.1.6	PROCESSO DE RECONSTRUÇÃO DO SINAL	45
3.2	PROCEDIMENTOS EXPERIMENTAIS PARA VALIDAÇÃO DO SISTEMA DESENVOLVIDO	47
3.2.1	BANCO DE DADOS	47
3.2.2	MÉTRICAS DE DESEMPENHO	47
3.2.3	TESTES REALIZADOS	48

3.3	MÉTODOS PARA A ANÁLISE DOS RESULTADOS.....	49
4	RESULTADOS E DISCUSSÕES.....	53
5	CONCLUSÃO.....	70
	REFERÊNCIAS BIBLIOGRÁFICAS.....	71
	APÊNDICES.....	78

LISTA DE FIGURAS

1.1	Sinal ECG típico. Adaptado de [2].	1
2.1	Geração do potencial de ação numa célula excitável. Adaptado de [29].	9
2.2	Gráfico de um ECG típico onde se observam as ondas P, QRS, T e U. Fonte: [1].	10
2.3	Resolução Tempo-Frequência. Adaptado de [45].	16
2.4	Diagrama representativo banco de filtros. Fonte: [36].	19
2.5	Diagrama de blocos típico usando diferentes tipos de compressão junto com a compressão por transformação.	21
2.6	Quantização uniforme (<i>midtread quantizer</i>). Adaptado de [7].	22
2.7	Quantização não-uniforme (<i>midrise quantizer</i>). Adaptado de [7].	23
2.8	Quantização escalar.	25
2.9	Quantização vetorial. Adaptado de [7].	26
2.10	Codificação Huffman. Adaptado de [6].	27
2.11	Codificação Aritmética. Adaptado de [6].	28
2.12	Exemplo de um ECG original e reconstruído utilizando o algoritmo AZTEC. Fonte: [10].	30
2.13	Exemplo de decomposição wavelet para 32 coeficientes com relação pai-filho.	33
3.1	Diagrama de blocos do sistema implementado.	37
3.2	Sinal 102 da base de dados MIT-BIH no domínio do tempo. Fonte: [5].	37
3.3	Decomposição wavelets do sinal 102 em 5 níveis com a família db4 com cada nível delimitado.	38
3.4	Decomposição wavelets do sinal 102 em 5 níveis representado por banco de filtros.	38
3.5	Divisão em 8 sub-bandas da transformada wavelet db4 do sinal 102.	39
3.6	Correção gamma.	41
3.7	Compressão e expansão na correção gamma.	41
3.8	Divisão em 8 sub-bandas da transformada wavelet db4 do sinal 102 com diferentes níveis de quantização para cada sub-banda.	43
3.9	Transformada de wavelets db4 do sinal 102 após o processo de quantização.	44
3.10	Diagrama de blocos do sistema de reconstrução.	45
3.11	(a) Sinal 102 original. (b) Sinal 102 reconstruído a partir do código comprimido usando db4, 5 níveis de decomposição, 32 sub-bandas, $\lambda = 0,82$, $\lambda_1 = 0,6560$, $\lambda_2 = 0,1640$, obtendo uma compressão de CR= 12,80 : 1 e erro de reconstrução PRD= 4,2%. (c) Resíduo entre o sinal 102 original e o sinal 102 reconstruído.	46
3.12	Função de distribuição de probabilidade normal.	50
4.1	Gráfico de barras mostrando o desempenho de cada família de wavelets respeito ao seu <i>FM</i> .	58

4.2	Curva do FM médio (para os 7 primeiros sinais do banco de dados) obtido para cada família de wavelets utilizando os λ_1 e λ_2 ótimos de cada uma delas.....	59
4.3	Testes de Wilcoxon para cada grupo de famílias wavelets.....	65
4.4	Testes de Wilcoxon para as famílias wavelets com melhor desempenho.	67
4.5	(a) Sinal 117 original. (b) Sinal 117 reconstruído usando bior6.8, 5 níveis de decomposição, 32 sub-bandas, $\lambda = 0,66$, $K = 0,90$, $\lambda_1 = 0,594$, $\lambda_2 = 0,066$, obtendo uma compressão de CR= 11,40 : 1 e erro de reconstrução PRD= 1,38%. (c) Resíduo entre o sinal 117 original e o sinal 117 reconstruído.	68
4.6	Curva de PRD1 em função de CR para o sinal 117 utilizando a transformada Bi-ortogonal 6.8.....	69

LISTA DE TABELAS

4.1	Tabela com λ e K ótimos para cada família wavelet.....	53
4.2	Tabela com λ e K ótimos calculados a partir de 7 sinais, aplicados para cada família wavelet para os 41 sinais restantes do banco de dados.	55
4.3	Resultados de compressão para todo o banco de dados MIT-BIH.....	61
4.4	Valores de comparação para CR, PRD e FM do sinal 117 entre o método proposto e métodos de compressão do estado-da-arte	63

LISTA DE SÍMBOLOS

Símbolos Gregos

λ_1	Parâmetro que controla o re-ordenamento não linear das magnitudes dos coeficientes transformados
λ_2	Parâmetro que controla a quantização por sub-bandas
λ	$\lambda = \lambda_1 + \lambda_2$

Siglas e acrônimos

AZTEC	<i>Amplitude Zone Time Epoch Coding</i>
CORTES	<i>Coordinate Reduction Time Encoding System</i>
CT	<i>Cosine Transform</i> - Transformada de cossenos
CR	<i>Compression Ratio</i> - Fator de compressão
CS	<i>Compressed Sensing</i>
CWT	<i>Continuous Wavelet Transform</i> - Transformada contínua de wavelets
DCT	<i>Discrete Cosine Transform</i> - Transformada discreta de cossenos
DFT	<i>Discrete Fourier Transform</i> - Transformada discreta de Fourier
DPCM	<i>Differential Pulse Code Modulation</i>
DTWT	<i>Discrete Time Wavelet Transform</i> - Transformada wavelets de tempo discreto
DWT	<i>Discrete Wavelet Transform</i> - Transformada discreta de wavelets
ECG	Eletrocardiograma
EPE	<i>Energy packing efficiency</i> - Eficiência de empacotamento de energia
EZW	<i>Embedded zero-tree wavelet</i>
FAN/SAPA	<i>Fan and scan-along polygonal approximation</i>
FM	Fator de mérito
FFT	<i>Fast Fourier Transform</i> - Transformada rápida de Fourier
FT	<i>Fourier Transform</i> - Transformada de Fourier
HT	<i>Haar Transform</i> - Transformada de Haar
KLT	<i>Karhunen-Loeve Transform</i> - Transformada Karhunen-Loeve
LPC	<i>Linear Predictive Coding</i> - Codificação com predição linear
MIT-BIH	<i>Massachusetts Institute of Technology and Boston's Beth Israel Hospital</i>
MRA	<i>Multi-resolution Analysis</i> - Análise multi-resolução
P-QRS-R	Complexo PQRST no ECG
PRD	<i>Percentage root-mean square difference</i> - Raíz da diferença média percentual
QMF	<i>Quadrature Mirror Filter</i> - Filtro de espelho em quadratura
RLE	<i>Run Length Encoding</i>
SPIHT	<i>Set Partitioning in Hierarchical Trees</i>
STFT	<i>Short Time Fourier Transform</i> - Transformada de Fourier de tempo curto
TP	<i>Turning Point</i>
UTI	Unidade de terapia intensiva
VQ	<i>Vector Quantization</i> - Quantização vetorial
WHT	<i>Walsh-Hadamard Transform</i> - Transformada de Walsh-Hadamard
WT	<i>Wavelet Transform</i> - Transformada de wavelets

1 INTRODUÇÃO

O trabalho analisa algoritmos de compressão de sinais de eletrocardiografia (ECG) utilizando transformadas de wavelets, e faz uma análise sistemática das diferentes famílias usadas para compressão. Este capítulo detalha os objetivos e a justificativa desse estudo. Detalha ainda as contribuições desta análise, que sucintamente se referem a uma nova forma proposta para fazer a quantização em uma compressão por transformadas. Também se descreve uma forma de escolha sistemática de parâmetros que levam ao melhor desempenho de cada família, juntamente com a validação mediante a comparação das famílias de wavelets. Também é discutida a utilidade de compressão de sinais ECG, e o valor do método proposto frente a outras alternativas.

1.1 CONTEXTUALIZAÇÃO

O eletrocardiograma (ECG) constitui um dos sinais fisiológicos mais usados e de maior sucesso na engenharia biomédica, e serve para diagnosticar condições do coração [1]. Os músculos do coração, que constituem o chamado miocárdio, são excitados por impulsos elétricos provenientes dos neurônios e que se propagam até as células musculares do coração gerando diferentes potenciais de ativação. Diferentes células de diferentes áreas do coração são estimuladas gerando contrações musculares. A soma dos potenciais de ativação gerados formam o que conhecemos como um ECG [2], cuja forma padrão é mostrada na Figura 1.1.

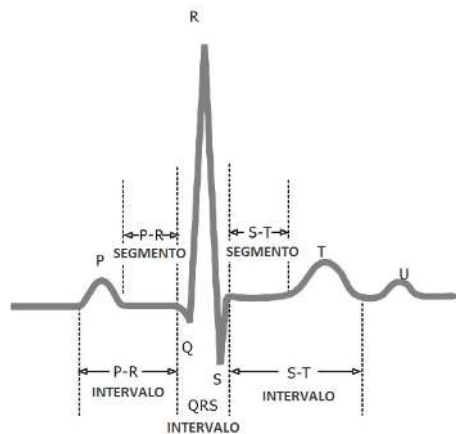


Figura 1.1: Sinal ECG típico. Adaptado de [2].

Um ECG é um sinal que mostra a atividade elétrica do coração. É uma ferramenta médica muito utilizada, e a informação que mostra é de alta relevância para um diagnóstico adequado dos possíveis maus funcionamentos do coração. Basicamente, consiste num registro gráfico da magnitude e na detecção temporal da atividade elétrica gerada pelos movimentos ventriculares e átrios.

Existem diferentes aplicações para um ECG. Ele é utilizado como medida preventiva, para autorizar a realização de alguns tipos de atividades que demandam uma alta exigência do coração, para diagnosticar possíveis doenças, tentar explicar problemas de saúde da pessoa, e outras. Um ECG pode ser executado por diferentes razões, por exemplo na suspeita de um ataque cardíaco, na suspeita de uma embolia pulmonar, arritmias cardíacas percebidas pelo médico, em casos de desmaios, em casos de convulsões, para monitorar o efeito que alguma medicação tem no coração, anormalidades eletrolíticas. Também são utilizados nos pacientes que estão sob o efeito de anestesia, em pacientes em cuidados críticos na unidade de terapia intensiva (UTI). Em casos particulares, podem ser utilizados para pessoal especializado em alguma área específica, por exemplo pilotos de avião, atletas de alta performance, etc. [3].

Encontram-se desenvolvidos e comercialmente disponíveis diferentes equipamentos para poder adquirir o ECG. A maioria deles têm algumas funções básicas como uma tela de alta resolução, que mostra o ECG enquanto é adquirido, ou seja, de forma online; uma impressora de alta resolução, interpretação da forma de onda (detecção de arritmias, malfuncionamentos dos processos cardíacos, ataque cardíaco, etc.), armazenamento do ECG, e até bateria para poder ser portáteis [4]. As mais recentes inovações em tecnologia do ECG incluem conexão sem fio a outros sistemas para a transmissão eletrônica dos ECG. Esses equipamentos são utilizados em clínicas, em unidades de terapia intensiva, ou, por exemplo, para realizar diagnósticos ou medir o desempenho durante o exercício.

Com os mais recentes avanços da tecnologia do ECG, temos ainda mais aplicações e equipamentos. Existem equipamentos portáteis, que facilitam o lugar de captura do ECG e permitem observar a atividade cardíaca do paciente as 24 horas do dia. Há pouco tempo, foram desenvolvidos sistemas de equipamento/software que permitem utilizar um smartphone como unidade de processamento com um software adequado, em conjunto com um periférico de baixa complexidade, para realizar o ECG. Todas essas novidades no âmbito do ECG deram lugar a uma nova área de interesse na medicina, chamada telemedicina. Esta área tem como objetivo poder realizar os estudos pertinentes, no nosso caso o ECG (telecardiologia), remotamente, e poder transmitir essa informação ao local de interesse, como um banco de dados, uma central de monitoramento de um hospital, uma central de estudos especializados, para observação do médico, entre outros usos. Outro uso do ECG que se observa no cotidiano é num marca-passos. Nele, está-se constantemente mensurando a atividade elétrica do coração para realizar ações de ajuste sobre o músculo cardíaco se for necessário. Esses avanços requerem uma forma de representação do ECG que seja eficiente, para o menor gasto de energia, menor uso de processamento, e menor necessidade de capacidade de armazenamento.

1.2 PROBLEMATIZAÇÃO E PROPOSTA

O armazenamento dos sinais ECG é de suma importância para registros médicos dos pacientes, estudos de doenças cardíacas do paciente, transferência a outros bancos de dados, etc. Por

esse motivo, a capacidade de armazenamento é um tópico de interesse. Pensemos por exemplo em um ECG de 24 hs de monitoramento de um paciente de 1 canal. Utilizando a frequência de amostragem e a resolução igual à utilizada no banco de dados do MIT-BIH (Arrhythmia Database) [5] teríamos 360 amostras por segundo, com resolução de 11 bits, durante 24 hs ($360 \times 11 \times 24 \times 60 \times 60 = 342144000$ bits = 42MB). No exemplo dado foi utilizado apenas 1 canal, porém, nos ECGs padrão, costuma-se utilizar 12 canais. Além disso taxas de amostragem e resoluções maiores são mais comuns hoje, elevando ainda mais o requisito de memória para a representação de cada sinal. É por essa razão que se procura reduzir o número de bits necessários para representar o sinal ECG com uma dada taxa de amostragem e um número de canais, o que é possível por meio de técnicas de compressão. O objetivo da compressão é conseguir reduzir o máximo possível a quantidade de bits necessária para representar o ECG, sem perder informação relevante dele.

Dessa forma, há necessidade de redução da memória que se precisa para o armazenamento local. Também é necessária a redução da memória utilizada para que se possa transmitir os dados com menor largura de banda. Outro aspecto a se ter em conta é a redução do uso de energia de alguns dispositivos, como por exemplo um marca-passos ou um dispositivo de monitoramento do paciente. Nesses dispositivos, com os dados comprimidos, é possível armazená-los e manipulá-los com menos energia, e portanto menos uso de bateria.

Quando falamos de uma técnica de compressão, estamos na verdade falando de dois algoritmos. Um para o processo de compressão, o outro para a processo de reconstrução. No processo de compressão opera-se sobre a fonte original para obter uma versão comprimida. No processo de reconstrução, opera-se sobre a versão comprimida, para obter uma versão reconstruída da fonte original. Tendo em conta essa diferenciação, podemos falar de compressão sem perdas (*lossless*), que acontece quando a versão reconstruída é igual à fonte original, e podemos falar de compressão com perdas (*lossy*), que acontece quando a versão reconstruída tem diferenças com relação à fonte original, porém alcança um nível maior de compressão [6, 7]. Existem algumas aplicações nas quais a compressão sem perdas é útil, em casos nos quais é muito importante que a informação recuperada seja idêntica à original. Contudo, no caso da compressão de sinais ECG, considerando que se precisa de uma redução de memória muito grande, as técnicas utilizadas são quase sempre com perdas, levando em conta um limite na compressão da informação para ter uma qualidade mínima no sinal reconstruído. De qualquer forma, existem trabalhos feitos nos quais a compressão sem perdas do sinal ECG é estudada [8], nos que se analisa quais fatores influenciam em maior proporção na compressão do ECG.

A compressão de dados é o processo de detectar e eliminar as redundâncias de um dado conjunto de dados. Falando de técnicas de compressão de sinais, elas têm sido classificadas principalmente em três categorias: compressão de dados no domínio do tempo, métodos transformados e técnicas de extração de parâmetros [9]. Os métodos no domínio do tempo contêm informação própria do sinal e os métodos transformados contêm informação transformada do sinal, nos quais os dados originais são reconstruídos por um processo inverso. Os compressores no domínio de tempo detectam as redundâncias na análise direta das amostras próprias do sinal. Os com-

pressores transformados utilizam uma representação do sinal em outro domínio para analisar as distribuições espectrais e de energia, conseguindo assim a detecção de redundâncias. O método de extração de parâmetros é irreversível, no qual uma característica do sinal é extraída [9]. Na maior parte da literatura que estuda a compressão dos sinais ECG, encontramos técnicas de compressão diretas ou transformadas. Os métodos diretos não precisam do domínio transformado e podem conseguir níveis de compressão altos. No entanto não tem controle sobre a qualidade do sinal recuperado, e nesse quesito, costumam ter resultados pobres [9].

Os métodos no domínio do tempo, que são os métodos clássicos, e foram os primeiros a ser utilizados, fazem uso de algoritmos de extrapolação e interpolação. Os algoritmos de extrapolação usam informação *a priori* de algumas amostras passadas, e os algoritmos de interpolação usam informação *a priori* de amostras passadas e futuras. Dividimos essas técnicas em três grandes grupos: tolerance-comparison compression, differential pulse code modulation (DPCM) e entropy coding (codificador de entropia). Tolerance-comparison compression utiliza extrapoladores polinomiais e interpoladores polinomiais, e os mais utilizados são os de ordem zero, e os de ordem um. A idéia do DPCM consiste em, ao se estimar as amostras, o erro entre a amostra atual e a estimada é quantizada e transmitida ou armazenada. Nos codificadores de entropia, que é um tipo de codificador sem perdas, a compressão se obtém explorando a redundância presente no sinal. Existem diferentes tipos desses codificadores. Em um dos principais tipos de codificadores de entropia, se assinam palavras código de diferente comprimento a cada símbolo de uma seqüência de dados dependendo da sua frequência de ocorrência [9]. Dentro dos métodos no domínio do tempo, temos os que foram desenvolvidos propriamente para a compressão de sinais ECG. Encontramos os algoritmos AZTEC [10], FAN/SAPA [11], CORTES [12], TP (Turning Point) [13] que são baseados no tolerance-comparison compression. Eles serão explicados no Capítulo 2.

Nos métodos transformados se realiza algum tipo de processamento ao sinal de entrada por meio de uma transformação linear ortogonal, e se realiza uma codificação dessa saída transformada, reduzindo a quantidade de informação necessária para representar o sinal original [9]. Existem várias transformadas ortogonais que são utilizadas no processamento digital de sinais como a transformada Karhunen-Loeve (KLT), a transformada de Fourier (FT), a transformada de cossenos (CT), a transformada Walsh-Hadamard (WHT), a transformada de Haar (HT), transformada wavelets (WT), etc. A transformada ótima é a KLT, pois é a que utiliza o menor número possível de funções ortonormais para representar o sinal de entrada, e ainda os coeficientes transformados são descorrelacionados, o que resulta numa minimização da entropia. No entanto, deve se enviar a tabela com as funções de base calculadas ao decodificador, além do custo computacional para calcular os vetores de base ser muito alto. Isso faz que na prática se utilizem transformadas sub-ótimas como as mencionadas anteriormente, as quais possuem vetores de base independentes do sinal de entrada, o que faz que elas não tenham que ser enviadas nem calculadas cada vez que se quer comprimir um sinal de entrada novo, e, portanto, tornam-se mais rápidas na execução. Podemos inferir pelo mencionado que a performance das transformadas tem um máximo teórico delimitado pela KLT [9].

A correta leitura dos eletrocardiogramas é necessária para a identificação dos seus diferentes componentes, a saber, P-QRS-T, que podem variar em magnitude e intervalo. A partir dessa leitura, será possível constatar irregularidades que podem ser identificadas, como doenças crônicas do coração, arritmias, variação do ritmo cardíaco, etc. Devido a isso, a compressão de um sinal ECG, não pode ter como único objetivo a redução dos bits necessários para representá-lo; também deve ter em conta a fidelidade que o sinal reconstruído terá com o sinal original, após sua descompressão.

Dessa maneira, podemos afirmar que existe uma relação estreita entre a capacidade de compressão em bits, e a fidelidade do sinal ECG recuperado após a descompressão. É esse compromisso, entre compressão e fidelidade do sinal ECG recuperado, que as diferentes técnicas de compressão de ECG têm como foco, e tentam, mediante diferentes métodos, conseguir a melhor relação para uma determinada aplicação. Na atualidade, são utilizados quase exclusivamente métodos transformados, que têm os melhores resultados. Existem métodos nos quais se utilizam diferentes transformadas, como a DCT, a DFT [14], métodos de duas dimensões [15] [16] [17], e mais recentemente, a transformada wavelet (WT) [18] [19]. Tendo em conta que existem trabalhos aplicando a WT [18, 19, 20, 21, 22, 23, 24, 25, 26, 27], não se achou na literatura um no qual a estratégia de compressão dos coeficientes transformados fosse dependendo das sub-bandas de decomposição além de levar em consideração a magnitude dos coeficientes (independentemente da sub-banda na qual se encontre). Não há portanto uma análise sistemática comparando as várias famílias de transformadas conhecidas para a fase de representação com essa estratégia, especificamente para sinais de ECG.

Em consequência, no projeto se compararão as transformadas de wavelets mais utilizadas, no que se refere à representação de sinais de ECG, utilizando a estratégia de realizar modificações sobre os coeficientes com base na sub-banda na qual se encontrem e nas amplitudes daqueles coeficientes. Assim, se analisaram as métricas de desempenho de algoritmos de compressão usando as diferentes transformadas testadas, avaliando a da diferença raiz quadrática média percentual (*percentage root mean square difference* - PDR) [28] em função da taxa de compressão (*compression ratio* - CR) [28] que são as métricas mais utilizadas na literatura. Em resumo, será proposto um método de compressão do sinal de eletrocardiografia utilizando transformadas wavelet para conseguir uma representação esparsa do mesmo, se realizará um re-ordenamento não linear (correção gamma) dos coeficientes em combinação com uma quantização por sub-bandas no domínio transformado, *run length encoding* (RLE) do resultado e, por último, se aplicará um codificador aritmético para minimizar a redundância do sinal resultante.

1.3 OBJETIVOS

O presente trabalho propõe um algoritmo de compressão de sinais ECG que utiliza o método de compressão por transformação, utilizando a transformada wavelet, particularmente a transformada wavelet discreta (DWT), para realizar a transformação. No estágio de quantização dos

coeficientes transformados se utiliza um método novo não achado na literatura, na qual se combina a quantização por sub-bandas, tendo em conta a concentração de energia de cada sub-banda (EPE - energy packing efficiency), com um re-ordenamento das magnitudes dos coeficientes utilizando uma correção não-linear (gamma correction). Por último no algoritmo é utilizado um codificador de entropia, particularmente uma codificação aritmética, para eliminar a redundância restante.

Assim, os objetivos deste trabalho são implementar e avaliar o desempenho de diferentes famílias wavelets na compressão de sinais ECG usando uma nova técnica de quantização, assim como avaliar o desempenho de compressão utilizando diferentes parâmetros para esse novo método de quantização proposto. Com esta análise proposta, pretende-se determinar quais são os parâmetros de quantização para cada transformada que levam ao melhor desempenho de compressão do algoritmo, e, de todas as transformadas assim configuradas (ou seja, sabendo quais são os parâmetros ótimos para a compressão), qual permite a melhor compressão e menor erro de reconstrução.

Posteriormente, com os resultados obtidos, se comparará, para avaliar as vantagens e desvantagens da técnica aplicada, com outros métodos descritos na literatura, do estado-da-arte. Também se avaliará a invariabilidade dos resultados testando o algoritmo com os parâmetros calculados para outros sinais e observando o comportamento dos resultados obtidos.

1.4 CONTRIBUIÇÕES

Como contribuições que este trabalho outorga se destaca a análise sistemática entre as diferentes famílias de wavelets e a proposta de um novo método de quantização não achado na literatura, para o qual se calculam também os parâmetros ótimos para cada família wavelet utilizada. O método empregado traz simplicidade na implementação quando é comparado com as técnicas atuais do estado-da-arte. Também se comparam os resultados de desempenho com as diferentes técnicas de compressão por transformadas, mostrando a utilidade do método proposto frente a outras alternativas.

1.5 JUSTIFICATIVA

O trabalho proposto introduzirá uma nova técnica a se ter em conta na compressão de sinais ECG, junto com abordagens diferentes para obter a compressão. Isso é valioso devido à ampliação do panorama atual sobre compressão por transformadas, na qual não tem se observado na literatura uma redistribuição não linear dos coeficientes a serem comprimidos. Por tanto, novas implementações específicas de ECG poderiam utilizar os resultados fundamentados apresentados no presente trabalho.

Devido ao fato de que o método proposto é um sistema computacionalmente simples, onde não temos que realizar um pre-processamento do sinal nem fazer um cálculo separado para cada sinal a ser comprimido (uma vez que calculemos os parâmetros ótimos do algoritmo), permitirá melhorar as aplicações de baixo consumo, importante em implementações com bateria, como um marca-passo, ou dispositivos que podem se levar com a pessoa (*wearable devices*). Também melhorará a performance da velocidade de transmissão e uso de largura de banda em outras aplicações, como centros médicos de estudos, postos remotos de monitoramento, ou transmissões desde sensores portáteis para o especialista que analisa o sinal.

1.6 ESTRUTURA DA DISSERTAÇÃO

Até agora foi dada uma pequena introdução sobre o contexto de compressão de sinais de ECG, e por que razão é de interesse conseguir bons resultados. No Capítulo 2 se dá um contexto teórico ao leitor para poder entender as técnicas aplicadas no trabalho, e entender o motivo das técnicas utilizadas. O capítulo começa mostrando como um sinal de ECG é gerado, e o que ele representa. Posteriormente trata-se da teoria de sinais, e sobre transformações lineares de sinais. Isso é explicado como introdução para uma melhor leitura da explicação de transformadas de wavelets, transformada utilizada no presente trabalho através de banco de filtros. Também se explica brevemente a teoria de compressão de dados, explicando o esquema básico de compressão por transformadas, quantização e codificação de entropia. Por último se explicam diversos métodos do estado-da-arte de compressão de sinais ECG.

No Capítulo 3 se descreve a metodologia adotada. Na Seção 3.1 se mostra o diagrama de blocos do algoritmo implementado, explicando cada uma das suas etapas. Elas são transformada wavelet e divisão em sub-bandas, quantização das sub-bandas e redistribuição não-linear da magnitude dos coeficientes, *run length encoding* (RLE) e codificação aritmética. Na Seção 3.2 se explicam os testes realizados e os motivos pelos quais esse tipo de testes foram aplicados. Finalizando, na Seção 3.3, é explicado como serão analisados os testes feitos, e como poderemos tirar conclusões a partir deles.

No Capítulo 4 se mostram os resultados dos testes explicados no Capítulo 3, fazendo uma análise deles, gerando discussões e explicando se o resultado é de acordo ao esperado. No Capítulo 5 se dá uma conclusão de todo o exposto, fazendo um sumário do que foi visto durante o trabalho, discutindo os resultados e o algoritmo em si, propondo trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 GERAÇÃO E CARACTERIZAÇÃO DO SINAL ECG

O coração é composto por diferentes tipos de músculos. As células musculares compartilham uma característica importante com as células nervosas, que é o fato de possuírem membranas excitáveis. A membrana celular é a parte que divide os fluidos intracelulares dos extracelulares onde encontramos diferentes concentrações iônicas. Essa membrana possui uma permeabilidade diferente para cada tipo de ion e, portanto, atua como um filtro iônico seletivo. O comportamento das células musculares é similar ao das células nervosas, porém a membrana de uma célula muscular é mais complexa que a membrana de uma célula nervosa, têm capas, e parâmetros elétricos distintos aos de uma célula nervosa. Por questões de simplicidade, na literatura é utilizado o modelo que descreve membranas de células nervosas como uma primeira aproximação para explicar o funcionamento das membranas de células musculares. Esse modelo foi proposto por Hodgkin e Huxley em 1952 [29].

Existe um equilíbrio importante no tecido cardíaco entre íons de sódio Na^+ e de potássio K^+ . O coração é um músculo, conhecido como o músculo cardíaco, composto por células musculares cardíacas, de comportamento similar às células musculares genéricas, e portanto explicaremos seu comportamento desenvolvendo o comportamento das células musculares. Quando uma das células musculares é estimulada, a permeabilidade iônica da membrana é alterada. Na primeira fase temos um aumento grande na permeabilidade ao Sódio Na^+ e, devido ao aumento no fluxo de Na^+ , a corrente iônica aumenta. Isto causa que o potencial da membrana, de -70 mV no repouso (ou aproximadamente -90 mV para o caso da membrana cardíaca), aumente na tentativa de equilibrar as correntes iônicas do sódio e do potássio. Ao atingir esse equilíbrio, um potencial de 20 mV é alcançado. Neste ponto, a célula se denomina despolarizada e a permeabilidade ao sódio se encontra num nível normal, porém a permeabilidade ao potássio está aumentada. É neste momento que os ions K^+ saem em suficiente quantidade para trazer o potencial da membrana celular novamente ao estado de repouso (repolarização). Assim, tomam lugar processos metabólicos que fazem o nivelamento final de íons Na^+ e K^+ para levar a célula às concentrações iônicas de repouso. Agora a célula se encontra em descanso e é capaz de ser re-estimulada. Todas essas atividades celulares descritas ocorrem no coração e formam parte de um processo de transmissão de pulsos elétricos através do coração que, quando são medidos na superfície da pele, formam o eletrocardiograma (ECG) [30]. Podemos observar o processo descrito na Figura 2.1 [29].

Existem, no caso do coração, outros íons que cumprem um papel no processo de equilíbrio iônico entre o interior e o exterior da célula. São os íons de cálcio Ca^{++} . Eles praticamente não entram na célula, exceto durante os potenciais de ação (por exemplo durante a despolarização) quando a permeabilidade ao cálcio aumenta. O cálcio que entra na célula deve ser tirado da célula, porque uma acumulação de cálcio causaria disfunção da célula. Para isso existem dois

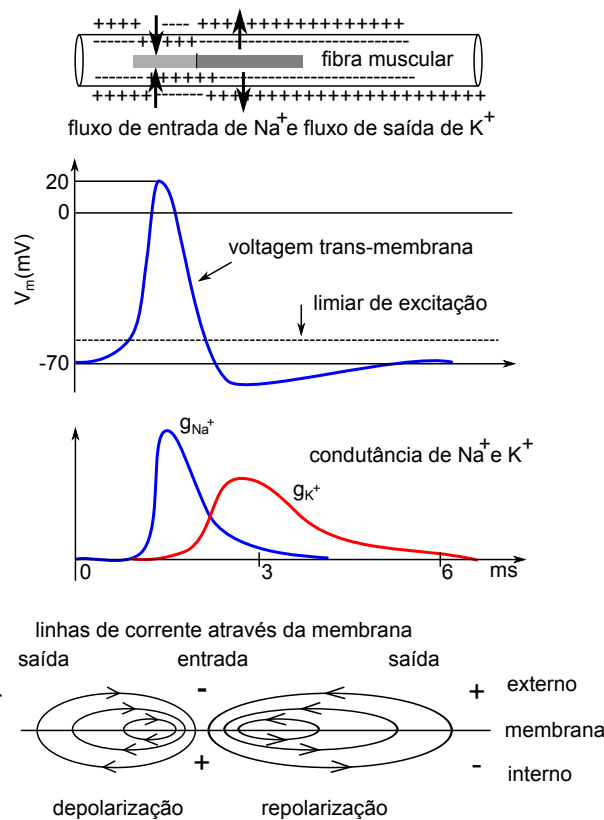


Figura 2.1: Geração do potencial de ação numa célula excitável. Adaptado de [29].

mecanismos. Um deles envolve uma bomba que tira cálcio da célula ativamente. O segundo mecanismo é do intercâmbio entre sódio-cálcio. Não está claro o funcionamento do segundo mecanismo, mas se sabe que três íons de sódio são trocados por cada um de cálcio [31].

Tendo sido explicado o funcionamento das células que compõem o coração, estamos em posição de poder explicar como o ECG é gerado. O coração tem quatro câmaras; as duas câmaras superiores se denominam átrios e as duas câmaras inferiores se denominam ventrículos. O átrio é uma bomba de baixa pressão que recebe sangue da circulação das veias. Pelo processo iônico descrito anteriormente, que acontece através das membranas celulares, é gerado um potencial extra-celular que logo excita células vizinhas, produzindo-se uma propagação elétrica célula a célula. Como o corpo atua como um resistor, esse potenciais extra-celulares se estendem até a superfície do corpo. Os potenciais gerados por pequenas massas de tecido não são apreciadas no ECG. Quando a onda elétrica de ativação chega ao músculo átrio, começa-se observar atividade elétrica na superfície do corpo, e podemos ver a primeira onda do ECG. Ela é chamada de onda P, e representa a ativação do átrio. O pulso elétrico prossegue do átrio através de células cardíacas especializadas que têm massas muito pequenas e não geram um sinal suficientemente grande para ser observado no ECG. Quando os ventrículos são excitados, eles se contraem e eles fornecem a principal força para circular sangue aos órgãos do corpo. A onda gerada aqui, na superfície do corpo, tem várias componentes, conhecidas como o complexo QRS. A polaridade e presença do complexo QRS depende da posição dos eletrodos no corpo e anormalidades que possam existir. Após isso, os ventrículos voltam ao seu estado de repouso elétrico e pode-se observar uma onda

de repolarização chamada onda T. Dependendo da pessoa, pode-se apresentar também uma onda U após a onda T [1]. O conjunto de ondas descrito gera como resultados o que conhecemos como ECG. Podemos observar o resultado no tempo dos potenciais elétricos mencionados na Figura 2.2, que dão como resultado o ECG.

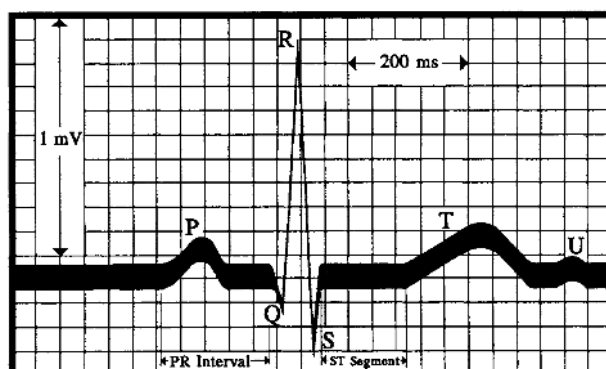


Figura 2.2: Gráfico de um ECG típico onde se observam as ondas P, QRS, T e U. Fonte: [1].

2.2 SINAIS EM DOMÍNIO CONTÍNUO E DISCRETO

No caso do ECG, que é o sinal sendo estudado no presente trabalho, a variável independente é chamada tempo. Se a variável independente é contínua, o sinal é chamado sinal de tempo contínuo e se a variável independente é discreta, o sinal é chamado sinal de tempo discreto. Um sinal de tempo contínuo é definido para todos os instantes de tempo. Por outro lado, um sinal de tempo discreto é definido para instantes discretos de tempo e, portanto, é uma sequência de números. Na maioria das aplicações, um sinal de tempo discreto se obtém a partir de um sinal de tempo contínuo amostrando este em intervalos uniformes de tempo. No caso do ECG, as versões digitais com as que trabalhamos na atualidade são versões amostradas do sinal elétrico contínuo produzido na superfície da pele pela atividade elétrica do coração. Se os instantes nos que o sinal de tempo discreto é definido estão espaçados uniformemente, a variável independente discreta n pode ser normalizada para tomar valores inteiros [32].

Com a disponibilidade de computadores digitais para realizar as tarefas de processamento de sinais, o processamento digital de sinais passou a ter cada vez mais importância. Consiste em três passos principais: conversão do sinal analógico para uma versão digital dele, processamento da versão digital, e, se for desejado para a aplicação particular, conversão do sinal digital processado novamente à sua forma analógica [32]. Para converter o sinal analógico em um sinal digital se utiliza um conversor analógico-digital (A/D). Nesse processo se tomam amostras do sinal analógico para formar o sinal digital. Pode-se demonstrar que se o espectro do sinal analógico é limitado em banda por B Hz, então pode-se reconstruir ele a partir das suas amostras se elas forem tomadas uniformemente a uma frequência $R > 2B$ Hz. Em outras palavras, a frequência mínima de amostragem é $f_s > 2B$ Hz [33]. A frequência de amostragem mínima mencionada se conhece como

taxa de Nyquist-Shannon e o intervalo de tempo correspondente $T_s = \frac{1}{f_s} = \frac{1}{2B}$ se conhece como intervalo de Nyquist-Shannon [33]. Por questões de simplicidade, assumiremos que o leitor está familiarizado com os conceitos e não se demonstrará a justificação da taxa de Nyquist-Shannon.

Na digitalização de um sinal contínuo, não existe apenas o processo de amostragem. Quando o valor contínuo é amostrado, ele possui uma amplitude que não sempre é um valor inteiro. Os valores representados digitalmente devem ter uma amplitude inteira definida, motivo pelo qual se realiza uma quantização do valor contínuo amostrado, arredondando o mesmo ao inteiro mais próximo. Podemos ver então que, durante a digitalização de sinais contínuos, realizam-se os processos de amostragem e quantização.

No caso próprio do ECG, a frequência de amostragem tem sido elemento de foco de vários estudos. Dependendo do tipo de uso que se queira realizar com o ECG, como análise, compressão, monitoramento, medição de variação dos batimentos cardíacos, etc., a frequência de amostragem utilizada pode variar. Ainda continua sendo um elemento de estudo, tendo-se justificado o uso de até 4000 Hz de taxa de amostragem; porém, na maioria das aplicações, é aceita uma taxa de amostragem que pode variar entre 250 Hz até 500 Hz [34, 35].

2.3 TRANSFORMADAS LINEARES DE SINAIS EM TEMPO DISCRETO

Um sinal ou função $x[n]$ da forma

$$x : \{0, 1, 2, \dots, N - 1\} \rightarrow \mathbb{R} \quad (2.1)$$

pode ser analisado, descrito, ou processado de uma forma mais eficiente, em algumas aplicações, se é expressado como uma decomposição linear da forma

$$x[n] = \sum_{k=0}^{N-1} \hat{X}[k] b_k[n], \quad (2.2)$$

em que k é um índice inteiro finito, $\hat{X}[k]$ são os coeficientes transformados, e as funções b_k formam o denominado conjunto de expansão. Caso esse conjunto de expansão permita a expansão 2.2 para todo sinal da forma 2.1, e supondo que as funções b_k são independentes, então o conjunto de expansão é chamado de base para essa classe de funções. Cabe observar que no, caso desse conjunto de expansão ser uma base, a expansão 2.2 é única.

Dizer que é uma base significa que é um conjunto tal que qualquer sinal do espaço é uma combinação linear dos elementos desse conjunto. Se essa base é ortogonal, ou seja:

$$\langle b_k[n], b_l[n] \rangle = \sum_{n=0}^{N-1} b_k[n] b_l^*[n] = 0 \quad \forall k \neq l, \quad (2.3)$$

então os coeficientes transformados podem ser calculados mediante o produto interno [36]

$$\hat{X}[k] = \langle x[n], b_k[n] \rangle = \sum_{n=0}^{N-1} x[n] b_k^*[n], \quad (2.4)$$

em que f é um fator de normalização.

Em vários cenários, é desejável decompor uma função em alguma combinação de funções mais elementares, chamadas funções de base. Fazendo isso, algumas operações matemáticas, ou cálculos a serem feitos sobre o sinal, são amplamente simplificados e, além disso, uma decomposição como a mencionada permitiria obter informações valiosas para um problema dado, que inicialmente não eram evidentes. [37]

2.3.1 Notação Matricial

Podemos expressar as transformadas previamente mencionadas através de uma notação matricial [36]. Para o caso de

$$\hat{X}[k] = f \sum_{n=0}^{N-1} x[n] b_k^*[n], \quad (2.5)$$

podemos re-escrever a expressão como:

$$f \begin{bmatrix} b_0^*[0] & b_0^*[1] & b_0^*[2] & \dots & b_0^*[N-1] \\ b_1^*[0] & b_1^*[1] & b_1^*[2] & \dots & b_1^*[N-1] \\ b_2^*[0] & b_2^*[1] & b_2^*[2] & \dots & b_2^*[N-1] \\ & & \ddots & & \\ b_{N-1}^*[0] & b_{N-1}^*[1] & b_{N-1}^*[2] & \dots & b_{N-1}^*[N-1] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} = \begin{bmatrix} \hat{X}[0] \\ \hat{X}[1] \\ \hat{X}[2] \\ \vdots \\ \hat{X}[N-1] \end{bmatrix} \quad (2.6)$$

que pode ser interpretada como

$$\hat{X} = fTx. \quad (2.7)$$

Para o caso de

$$x[n] = \sum_{k=0}^{N-1} \hat{X}[k] b_k[n], \quad (2.8)$$

podemos re-escrever a expressão como:

$$f \begin{bmatrix} b_0[0] & b_1[0] & b_2[0] & \dots & b_{N-1}[0] \\ b_0[1] & b_1[1] & b_2[1] & \dots & b_{N-1}[1] \\ b_0[2] & b_1[2] & b_2[2] & \dots & b_{N-1}[2] \\ & & \ddots & & \\ b_0[N-1] & b_1[N-1] & b_2[N-1] & \dots & b_{N-1}[N-1] \end{bmatrix} \begin{bmatrix} \hat{X}[0] \\ \hat{X}[1] \\ \hat{X}[2] \\ \vdots \\ \hat{X}[N-1] \end{bmatrix} = \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (2.9)$$

que pode ser interpretada como

$$x = I_n \hat{X}, \quad (2.10)$$

em que

$$I_n = \frac{1}{f} T^{-1}. \quad (2.11)$$

Esta notação será de utilidade na discussão acerca das transformadas wavelet, apresentadas a seguir.

2.4 TRANSFORMADA WAVELETS

Os métodos utilizados para estudar o ECG eram baseados no domínio do tempo. Não obstante, esse método não é sempre suficiente para estudar todas as características dos sinais de ECG. A representação em frequência é requerida para obter outras informações importantes. Com esse propósito, é aplicada a Transformada de Fourier (FT), que, como o sinal representado num meio digital é discreto, é implementada utilizando a Transformada Discreta de Fourier (DFT) através do algoritmo de implementação rápida chamado Transformada Rápida de Fourier (FFT), que diminui consideravelmente a complexidade do algoritmo $[O(n \log n)]$ [37]. Entretanto, não consegue fornecer informação sobre a localização exata dos componentes de frequência no tempo. Em outras palavras, esse tipo de transformada é útil quando se trabalha com sinais estacionários, ou seja, quando as componentes de frequência que os compõem se encontram presentes sempre, em qualquer instante do tempo. Por outro lado, no caso da maioria dos sinais que encontramos na prática, por exemplo na bio-medicina, nos encontramos com sinais não-estacionários. Nesse caso, as componentes de frequência que compõem o sinal variam dependendo do instante que esteja sendo analisado [38].

No caso da FT, ela calcula as componentes de frequência que formam o sinal para todos os instantes de tempo ($X(f) = \int_{-\infty}^{\infty} x(t)e^{-2j\pi ft} dt$) [39], o que quer dizer que essa transformada não leva em consideração os instantes em que se encontram essas componentes de frequência, o que, no que se refere à análise de sinais não-estacionários, não é de muita utilidade, exceto em algumas aplicações nas quais não é importante ter informação sobre os instantes de ocorrência das componentes de frequência, como por exemplo filtragem de ruído de um sinal, análise das componentes espectrais de um sinal, entre outras. Por essas razões, e considerando que o ECG é variável no tempo (sinal não-estacionário), é essencial uma descrição concreta das componentes de frequência conforme a sua localização no tempo. Por isso, é justificado o uso de uma representação em tempo-frequência [40].

A ferramenta disponível para este objetivo é a Transformada de Fourier de Tempo Curto (STFT), que consiste de forma resumida em aplicar uma FT a versões janeladas do sinal, assumindo que o tamanho da janela é o suficientemente pequeno para considerar que o sinal dentro dela se comporta como um sinal estacionário. Apesar disso, o problema existente com esse enfoque é a resolução tempo-frequência, que não é ótima, devido a que possui a mesma precisão temporal para componentes de alta frequência e baixa frequência (janelas de igual tamanho). É intuitivo observar que, para componentes de alta frequência, é desejável uma alta precisão no tempo (janelas de curta duração) e, para componentes de baixa frequência, não é necessária essa alta precisão no tempo (janelas de longa duração) [41]. Novamente, nos encontramos frente a um problema para o análise do ECG, tendo em vista que uma boa resolução tempo-frequência é necessária. Por esse motivo, precisa-se utilizar uma técnica mais adequada. Entre as várias transformações tempo-frequência existentes, a transformada wavelet (WT) é de especial utilidade para a tarefa em questão.

A WT é especialmente valiosa porque permite esclarecer simultaneamente informação espec-

tral e temporal de uma maneira mais flexível que a STFT, já que utiliza uma janela de largura variável. Por essa razão, a WT produz uma decomposição tempo-frequência do sinal mais efetiva que a STFT. Utilizando a WT, conseguimos informação de curta duração, alta frequência e também informação de longa duração, baixa frequência, ambas simultaneamente. Em consequência, a WT é útil para a análise de características de sinais não-estacionárias, como, por exemplo, o ECG [19].

2.4.1 Transformada de Wavelets em domínio Contínuo

Para poder explicar como a WT funciona, primeiro devemos entender uma série de princípios que serão explicados a seguir. O problema da resolução tempo-frequência mencionado anteriormente se deve a um fenômeno físico chamado Princípio de Incerteza de Heisenberg [41]. Inicialmente, este princípio era utilizado para explicar o momento e a localização de partículas em movimento, mas também pode ser aplicado à informação tempo-frequência de um sinal. Basicamente, estabelece que não podemos saber a representação tempo-frequência exata de um sinal, ou seja, não podemos saber quais componentes espectrais existem em um instante dado. O que de fato podemos saber são os intervalos de tempo onde bandas de frequência existem. É possível analisar o sinal utilizando outro enfoque chamado análise multi-resolução (MRA). A idéia do MRA é analisar o sinal em diferentes frequências utilizando diferentes resoluções [42]. Em outras palavras, boa resolução no tempo e baixa resolução em frequência para as altas frequências, e boa resolução em frequência e baixa resolução no tempo para baixas frequências. Esse enfoque guarda pertinência com o fato de que as altas frequências têm curta duração e as baixas frequências têm longa duração. Na prática, a maioria dos sinais que encontramos apresentam essas características [43].

A WT faz uso do princípio MRA e consiste em utilizar wavelets como funções de base para realizar a transformação. O nome wavelet significa onda pequena. Pequena se refere a que a função de base utilizada é de duração finita. Onda se refere a que a função de base é de caráter oscilatório, ou seja, que sua integral no intervalo de duração da função é igual a 0. Existe outro termo importante denominado wavelet mãe, que é utilizado para explicar que as diferentes funções de base (de diferente comprimento) usadas no processo de transformação são obtidas a partir de uma função principal, chamada wavelet mãe. Ela serve como protótipo para gerar as outras funções de base. Para obter as funções de base, se realizam translações e escalonamentos da wavelet mãe. As translações se referem a informação do tempo no domínio transformado. Os escalonamentos se referem à inversa da frequência ($1/f$) que chamaremos escala, que nos dará informação indireta da frequência (quanto menor a escala é, maior a frequência, e quanto maior a escala é, menor a frequência) [44].

Quando se calcula a WT, a wavelet mãe serve de protótipo para calcular todas as funções de base, que chamaremos de janelas. Todas as janelas que se utilizam para realizar a transformada são versões dilatadas ou comprimidas, e deslocadas da wavelet mãe. Primeiro se realiza o produto da função wavelet mais comprimida (a de maior frequência) pelo sinal e se integra para todo o

intervalo de tempo, e o resultado é multiplicado por um número constante de normalização, para que o sinal transformado tenha a mesma energia para cada escala. Logo, essa mesma wavelet comprimida é deslocada à direita e se realizam novamente todas as operações mencionadas. Os deslocamentos dessa wavelet são feitos até ela chegar ao final do sinal. Nessa instância, a wavelet é dilatada por um valor pequeno, volta até o começo do sinal e se realiza novamente todo o processo de produtos e deslocamentos até chegar ao final do sinal novamente, onde se realiza uma nova dilatação pequena da wavelet, e começa novamente o processo. Quando o processo é completado para todas as dilatações da wavelet, a WT foi calculada, a partir da relação

$$\Psi_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt, \quad (2.12)$$

em que τ é o parâmetro de translação, s é o parâmetro de dilatação/compressão, ou escala como mencionamos previamente (cabe destacar que a compressão da que se fala aqui não é compressão da informação da qual se falará em outra subseção), ψ é a função de transformação, que denominamos wavelet mãe, e $\frac{1}{\sqrt{|s|}}$ é o fator de normalização [40].

Para ilustrar e entender melhor o explicado sobre a WT, observemos a Figura 2.3. Podemos ver que a largura e altura das caixas muda, porém sua área é constante, ou seja, cada caixa representa um área igual do plano tempo-frequência, mas com diferentes proporções. É interessante observar que para baixas frequências, a altura das caixas é mais curta (melhor resolução em frequência) mas suas larguras são mais compridas (pior resolução no tempo). O contrario acontece para altas frequências. No caso da STFT mencionada anteriormente, como as janelas são de largura constante, todas as caixas nessa ilustração seriam do mesmo tamanho.

A WT tem uma inversa com a qual se pode obter novamente o sinal no tempo. Utiliza-se a seguinte equação:

$$x(t) = \frac{1}{C_\psi^2} \int_s \int_\tau \Psi_x^\psi(\tau, s) \frac{1}{s^2} \psi \left(\frac{t - \tau}{s} \right) d\tau ds, \quad (2.13)$$

em que C_ψ é a constante de admissibilidade, que depende da wavelet sendo utilizada [40]. Para a transformada inversa existir e ter-se uma reconstrução perfeita do sinal original, tem que se cumprir a condição de admissibilidade, que diz:

$$C_\psi = \left[2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{|\xi|} d\xi \right]^{\frac{1}{2}} < \infty, \quad (2.14)$$

em que $\hat{\psi}(\xi)$ é a FT de $\psi(t)$. A equação 2.14 implica que $\hat{\psi}(0) = 0$, ou seja, deve-se cumprir

$$\int \psi(t) dt = 0, \quad (2.15)$$

que significa que a wavelet deve ser de caráter oscilatório e, como temos visto, muitas famílias wavelets têm essa característica, ou seja que a condição de admissibilidade não é muito restritiva para as funções wavelets.

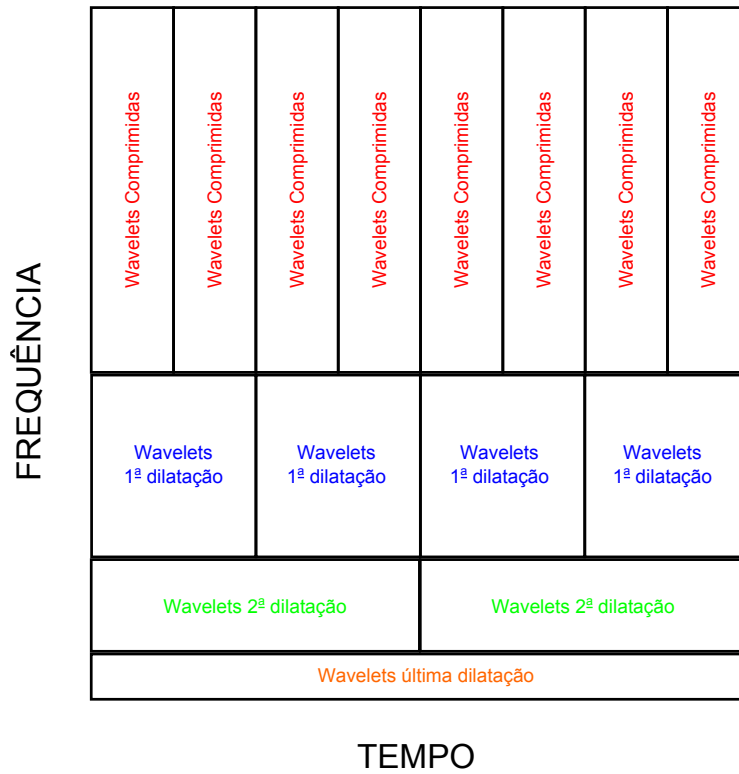


Figura 2.3: Resolução Tempo-Frequência. Adaptado de [45].

Até agora vimos que a WT é calculada fazendo dilatações da janela de análise (variando s), fazendo deslocamentos no tempo (variando τ), multiplicando pelo sinal e integrando sobre o intervalo de tempo. Essa forma de definir a WT através das equações mencionadas, é denominada CWT. Na prática, pode se utilizar uma versão amostrada da CWT, que seria uma versão aproximada da CWT, usando amostras equi-distantes da CWT, para não ter que realizar o cálculo nos infinitos pontos no intervalo de tempo, devido a que isso seria impraticável. Chamamos a essa variação de transformada wavelets de tempo discreto (DTWT), que é muito utilizada em processamento de imagens por exemplo. Existe também outra versão chamada a Transformada Discreta de Wavelet (DWT) na qual se restringe o valor de s na equação 2.12 a valores discretos 2^j . A DWT tem a particularidade de possuir uma boa concentração de energia do sinal em poucos coeficientes, o que faz dela uma ferramenta muito utilizada na compressão de sinais.

2.4.2 Transformada de Wavelets em Domínio Discreto

O explicado nas Seções 2.3 e 2.3.1 servirá para poder explicar a DWT. Se observamos a equação matricial 2.6 na Seção 2.3.1 e a equação 2.4 na Seção 2.3, será fácil observar a similaridade com a CWT. Por questões de simplicidade, re-escreveremos a equação de transformação linear discreta e a equação da CWT.

$$\hat{X}[k] = f \sum_{k=0}^{N-1} x[n]b_k^*[n], \quad (2.16)$$

$$\Psi_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt. \quad (2.17)$$

Podemos ver a equivalência entre as equações. O fator f da transformação linear seria o fator $\frac{1}{\sqrt{|s|}}$ da CWT. O sinal $x[n]$ da transformação linear seria o sinal $x(t)$ da CWT. O conjunto de funções de base $b_k^*[n]$ da transformação linear seriam as wavelets $\psi^* \left(\frac{t - \tau}{s} \right)$ dilatadas e transladadas da CWT. A diferença está em que no caso da transformação linear estamos falando de um caso discreto, e na CWT de um caso contínuo. Para a CWT virar discreta, devemos discretizar s para $s = s_0^j$ e τ para $\tau = k \cdot s_0^j \cdot \tau_0$, em que $s_0 > 1$ e $\tau_0 > 0$ [41]. Se pensamos na função contínua de wavelet como

$$\psi_{\tau, s} = \frac{1}{\sqrt{s}} \psi \left(\frac{t - \tau}{s} \right), \quad (2.18)$$

e se substituirmos os valores de τ e s pelos mencionados obtemos:

$$\psi_{j, k} = s_0^{-\frac{j}{2}} \psi(s_0^{-j} t - k \tau_0), \quad (2.19)$$

em que j é o índice associado às dilatações que a função wavelet terá e k é o índice associado às translações para cada nível de dilatação da função wavelet [42].

Para o caso da DWT $s_0 = 2$ e $\tau_0 = 1$, o que nos deixa com $s = 2^j$ e $\tau = k \cdot 2^j$. Isso quer dizer que apenas as escalas $s = 2, 4, 8, 16, \dots$, etc. são utilizadas. No eixo do tempo, como as escalas discretas variam por fatores de 2, a taxa de amostragem é reduzida por um fator de 2, para cada escala. Por exemplo, se $s = 2$, então $\tau = 2, 4, 6, 8, \dots$, se $s = 4$ então $\tau = 4, 8, 12, 16, \dots$, se $s = 8$ então $\tau = 8, 16, 24, 32, \dots$, e assim sucessivamente. Ou seja, agora temos um número finito de dilatações e translações para calcular a DWT.

Se observamos a equação

$$f \begin{bmatrix} b_0^*[0] & b_0^*[1] & b_0^*[2] & \dots & b_0^*[N-1] \\ b_1^*[0] & b_1^*[1] & b_1^*[2] & \dots & b_1^*[N-1] \\ b_2^*[0] & b_2^*[1] & b_2^*[2] & \dots & b_2^*[N-1] \\ & & \ddots & & \\ b_{N-1}^*[0] & b_{N-1}^*[1] & b_{N-1}^*[2] & \dots & b_{N-1}^*[N-1] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} = \begin{bmatrix} \hat{X}[0] \\ \hat{X}[1] \\ \hat{X}[2] \\ \vdots \\ \hat{X}[N-1] \end{bmatrix}, \quad (2.20)$$

podemos ver que os coeficientes $b_{N-1}^*[n]$ correspondem ao primeiro deslocamento da wavelet de maior frequência $s = 2$, $b_{N-2}^*[n]$ correspondem ao segundo deslocamento da wavelet de maior frequência $s = 2$, $b_{N-3}^*[n]$ correspondem ao terceiro deslocamento da wavelet de maior frequência $s = 2$ e assim sucessivamente até $b_{N-\frac{N}{2}}^*[n]$. $b_{N-\frac{N}{2}-1}^*[n]$ correspondem ao primeiro deslocamento da wavelet da frequência $s = 4$, $b_{N-\frac{N}{2}-2}^*[n]$ correspondem ao segundo deslocamento da wavelet da frequência $s = 4$ e assim sucessivamente até $b_{N-\frac{N}{2}-\frac{N}{4}}^*[n]$. $b_{N-\frac{N}{2}-\frac{N}{4}-1}^*[n]$ correspondem ao primeiro deslocamento da wavelet da frequência $s = 8$, e assim sucessivamente até completar o sinal.

2.5 BANCO DE FILTROS MULTI-TAXA

A DWT pode ser calculada de maneira eficiente utilizando técnicas da teoria de banco de filtros. Neste caso, se utilizam filtros com diferentes frequências de corte para analisar as diferentes escalas. Se utilizam filtros passa-alta para analisar as altas frequências e filtros passa-baixa para analisar as baixas frequências. O procedimento começa filtrando o sinal $x[n]$ com um filtro com resposta ao impulso $h[n]$. Filtrar um sinal é equivalente a fazer uma convolução no domínio do tempo entre $x[n]$ e $h[n]$. A convolução se define como:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]. \quad (2.21)$$

Se o sinal $x[n]$ foi amostrado utilizando o critério de Nyquist [39] então a frequência máxima do sinal no domínio de frequências discreto é π rad (as frequências no domínio discreto se expressam em radianos). Após fazer uma filtragem passa-baixa da metade da banda, a frequência mais alta será $\frac{\pi}{2}$, então, de acordo com Nyquist, a metade das amostras podem ser eliminadas, ou seja, realiza-se uma sub-amostragem por um fator de 2. Este processo tem duas consequências: a filtragem passa-baixa divide pela metade a resolução, mas não modifica a escala s . A sub-amostragem pelo fator de 2 dobra a escala s . Esse procedimento pode se expressar como [36]:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[2n-k]. \quad (2.22)$$

A DWT analisa diferentes bandas de frequência com diferentes resoluções, decompondo o sinal numa aproximação macro (baixa frequência) e detalhes (alta frequência). A decomposição do sinal em diferentes bandas de frequência é obtida mediante filtros passa-alta e passa-baixa sucessivos. O sinal $x[n]$ primeiro passa por um filtro passa-alta $g[n]$ e um filtro passa-baixa $h[n]$. Após a filtragem, metade das amostras podem ser descartadas devido ao teorema de Nyquist-Shannon, devido a que a frequência mais alta agora é $\frac{\pi}{2}$, e portanto, se sub-amostra por 2 descartando cada outra amostra. Pode ser expressado como:

$$y_{high}[k] = \sum_n g[n]x[2k-n], \quad (2.23)$$

$$y_{low}[k] = \sum_n h[n]x[2k-n]. \quad (2.24)$$

Podemos ver que essa decomposição reduz pela metade a resolução no tempo devido a que estão presentes apenas a metade de amostras. Porém, a resolução em frequência é dobrado. O processo descrito pode ser repetido novamente na saída do filtro passa baixa obtendo cada vez mais precisão em frequência e piorando a resolução temporal. É bom observar que, devido às sucessivas sub-amostragens por 2, existe um máximo de decomposições que poderão ser realizadas. Para um sinal de 1024 amostras por exemplo, podem-se realizar 10 níveis de decomposição ($2^{10} = 1024$).

Existe uma propriedade importante na relação entre os filtros passa-baixa e passa-alta empregados na decomposição do sinal. Eles não são independentes um do outro, cumprem a relação

$$g[L - 1 - n] = (-1)^n h[n], \quad (2.25)$$

em que $g[n]$ é o passa-alta, $h[n]$ o passa-baixa e L o comprimento do filtro. Filtros que cumprem essa condição são chamados de filtros QMF (Quadrature Mirror Filters) [40]. Como os filtros mencionados são filtros de metade de banda, eles formam bases ortonormais, o que facilita a reconstrução do sinal. Para realizar a reconstrução, simplesmente aplicamos o processo inverso. Os sinais em cada nível são sobre-amostrados por 2 através de filtros de reconstrução $g[n]$ e $h[n]$ (passa-alta e passa baixa) e depois são somados, em cada nível. Assim a fórmula vira (para cada nível):

$$x[n] = \sum_{k=-\infty}^{\infty} (y_{high}[k]g[-n + 2k]) + (y_{low}[k]h[-n + 2k]). \quad (2.26)$$

O processo explicado pode se observar na Figura 2.4. Aqui, v_3 é o sinal original, h_1 é um filtro passa-alta, h_0 é um filtro passa-baixa (ambos cumprem as condições mencionadas para QMF) e w_2 , w_1 , w_0 e v_0 são decomposições do sinal para diferentes frequências.

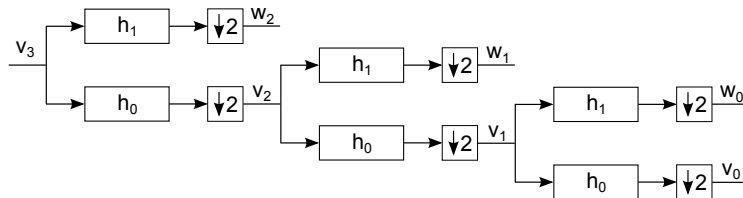


Figura 2.4: Diagrama representativo banco de filtros. Fonte: [36].

2.6 COMPRESSÃO DE DADOS

Até agora temos explicado como se gera um sinal de ECG, de onde ele vem, por que ele é útil, e por que é interessante comprimir esse sinal. Com esse propósito temos explicado as diferenças entre sinais contínuos e discretos, como realizar uma transformada linear de um sinal em tempo discreto, em que consistem as transformadas de wavelets, e sua aplicação mediante banco de filtros multi-taxa. O que resta é explicar o conceito de compressão, que significa comprimir um sinal, e como obtemos essa compressão, assim como falar dos diferentes tipos de compressão, e as técnicas particulares que serão utilizadas neste trabalho.

Os algoritmos de compressão de dados são utilizados para reduzir o número de bits requeridos para representar um sinal, por exemplo, uma imagem, uma sequência de vídeo, áudio ou sinais bio-médicos. Precisamos da compressão de dados pelo fato de que, com os avanços tecnológicos, geramos mais informação, na forma de valores representados por bytes de dados, e precisamos da redução do armazenamento necessário para manter esses dados [7]. Como a quantidade de bytes num meio de armazenamento físico é limitada, bem como a banda de transmissão, é de suma importância possuir algoritmos de compressão que reduzam o espaço que os arquivos mencionados

ocupam. Por outro lado, a necessidade de armazenamento massivo e transmissão aumentam mais rápido que as capacidades tecnológicas de armazenamento e compressão que desenvolvemos. Por essa razão vemos também como a compressão de dados é necessária.

Podemos achar vários métodos de compressão de dados. Eles são baseados em diferentes ideias e são úteis para diferentes tipos de dados, mas todos, de uma forma ou outra, se baseiam no mesmo princípio, que é comprimir dados removendo a redundância presente nos dados originais [6]. Qualquer fonte de dados não aleatória tem estrutura, e podemos explorar essa estrutura para obter uma representação mais comprimida dos dados. É importante ressaltar que para melhorar a eficiência de um algoritmo de compressão, a natureza dos dados deve ser levada em consideração. Para comprimir uma sequência de dados, o algoritmo deve examinar os dados, achar redundâncias, e tentar tirar essas redundâncias. Porém, elas dependem do tipo de dados presente (texto, imagens, vídeo, sinais biomédicos, etc.) motivo pelo qual deve-se desenvolver um novo método de compressão, que terá o melhor desempenho para esse tipo de dados [6].

Existem diversas formas de dividir os tipos de compressão de dados, levando em consideração diferentes objetivos que se buscam com a compressão, ou diferentes aspectos do sinal que são explorados para obter a compressão. Utilizaremos a divisão clássica de compressão de dados, que é baseada nos requerimentos da reconstrução: compressão sem perdas, em que o sinal reconstruído (y) é igual ao sinal original (x), e compressão com perdas, em que obtemos uma compressão maior, porém y pode ser diferente de x . Na compressão sem perdas, não há perda nenhuma de informação, ou seja, se essa técnica foi utilizada, então os dados originais podem ser recuperados exatamente a partir dos dados comprimidos. Um exemplo seria compressão de texto, em que pequenas diferenças podem mudar completamente o significado de uma frase. Outros usos são, por exemplo, em imagens médicas, dados obtidos de satélites e dados bancários [7]. No caso de compressão com perdas, existe uma perda de informação e os dados originais não podem ser recuperados exatamente. Entretanto, se essa diferença no sinal reconstruído pode ser tolerada, geralmente obtemos uma compressão muito mais alta que a conseguida com uma compressão sem perdas. Conceitualmente, na compressão sem perdas se exploram os dados redundantes para obter a compressão, e, na compressão com perdas, se exploram os dados irrelevantes. De fato, a perda de informação na compressão com perdas não é um problema para diversas aplicações. Por exemplo, ao transmitir ou armazenar voz, não precisamos do valor exato de cada amostra no sinal recuperado para entender o que foi dito pela pessoa na mensagem original.

2.6.1 Compressão por Transformadas

Dentro dos diferentes métodos de compressão, existe a compressão por transformadas. Se tomamos uma sequência de dados de entrada e transformamos eles em outra sequência em que a maior parte da informação se encontra contida em poucos elementos, podemos então codificar essa outra sequência resultando em compressão de dados. Em outras palavras, se compactamos a maior parte da informação de uma sequência fonte em alguns poucos elementos de uma sequência transformada utilizando uma função de transformação (que é inversível), e descartando os

elementos da sequência transformada que não contêm muita informação, podemos obter uma grande quantidade de compressão [7].

O processo de codificação mais comumente utilizado, fazendo uso da compressão por transformada, consiste em três tipos de compressão:

1. *Transformação*
2. *Quantização*
3. *Codificação de entropia*

A transformação consiste em mapear a sequência de dados fonte em uma sequência transformada, utilizando um mapeamento inversível, como foi explicado na Seção 2.3, na qual a maior parte da energia se encontrará em uma menor quantidade de elementos que na sequência fonte, podendo como consequência remover elementos da sequência transformada sem ter um impacto grande no fidelidade do sinal reconstruído, uma vez aplicada a transformada inversa. Essa transformação pode se aplicar à sequência de dados fonte toda, ou a blocos de tamanho N , nos quais a sequência de dados fonte é dividida. O processo de quantização é aplicado à sequência transformada. Consiste em diminuir a resolução em bits dos coeficientes, e existem diversas técnicas, dependendo da aplicação particular (o conceito será explicado em mais detalhe na Seção 2.6.2). Finalmente, os valores quantizados precisam ser codificados. Podem ser codificados utilizando métodos simples como um código de comprimento fixo, ou algo mais complexo como uma combinação de run-length encoding (RLE) e codificação aritmética (o conceito será explicado em mais detalhe na Seção 2.6.3) [7]. Se observamos a Figura 2.5, podemos ver um macro diagrama onde se mostram os diferentes estágios que se atravessam ao se realizar o processo mencionado, junto com o processo de reconstrução.

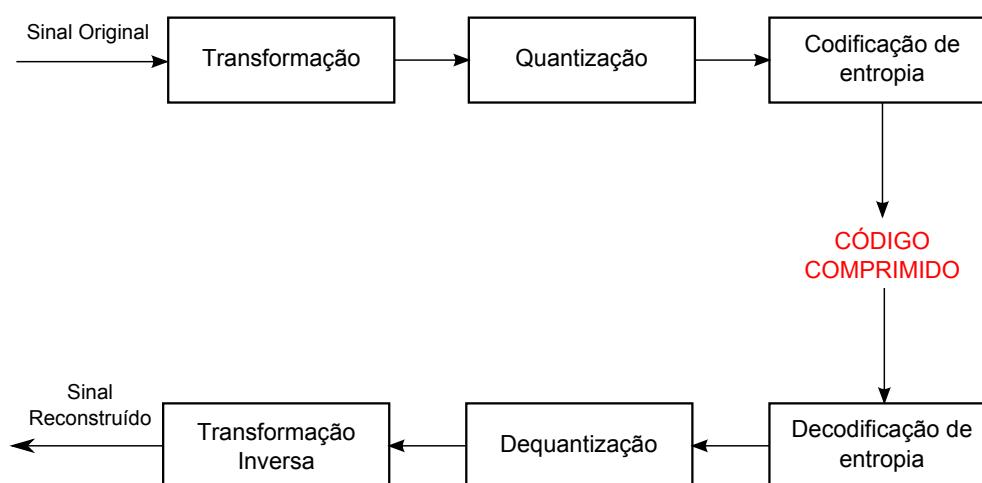


Figura 2.5: Diagrama de blocos típico usando diferentes tipos de compressão junto com a compressão por transformada.

2.6.2 Quantização

Visto de uma maneira conceitual, o processo de representar um conjunto grande de valores (possivelmente infinito) com um conjunto muito menor de valores é chamado de quantização. Por exemplo, se pensamos numa sequência fonte, e calculamos sua transformada, e arredondamos os coeficientes transformados ao inteiro mais próximo, estaríamos realizando uma quantização, pois diminuimos o conjunto de valores possíveis que os coeficientes transformados têm. Entretanto, se perdeu informação dos mesmos, pois não sabemos exatamente o valor original deles, uma vez que foi feito o arredondamento (imaginemos obter o número 3, não sabemos se ele veio de um 2,95 ou um 3,14 por exemplo). Essa perda de informação é o motivo pelo qual usamos a expressão com perdas quando nos referimos a esquemas de compressão com perdas, ou seja, o processo de quantização é o que gera perdas nos esquemas de compressão com perdas [7].

Existem duas formas gerais de dividir os diferentes tipos de quantizadores. Se as entradas e saídas de um quantizador são escalares, se define que esse quantizador é um quantizador escalar [7], e, se as entradas e saídas são vetores, se define o quantizador como um quantizador vetorial [7]. A outra forma de dividir os tipos de quantizadores é em quantizadores uniformes e quantizadores não-uniformes [7].

O caso mais simples é o dos quantizadores uniformes. Todos os intervalos de quantização têm o mesmo tamanho, e se possuem o valor zero como um dos seus níveis de representação, é denominado *midtread quantizer*, e no caso de não ter o zero como nível de representação é chamado *midrise quantizer* [7]. Na Figura 2.6, pode-se ver o diagrama de funcionamento de um *midtread quantizer*. Para valores de entrada que variam entre $[-0,5; 0,5]$ a saída é 0, para valores que variam entre $(0,5; 1,5]$ a saída é 1, e assim sucessivamente.

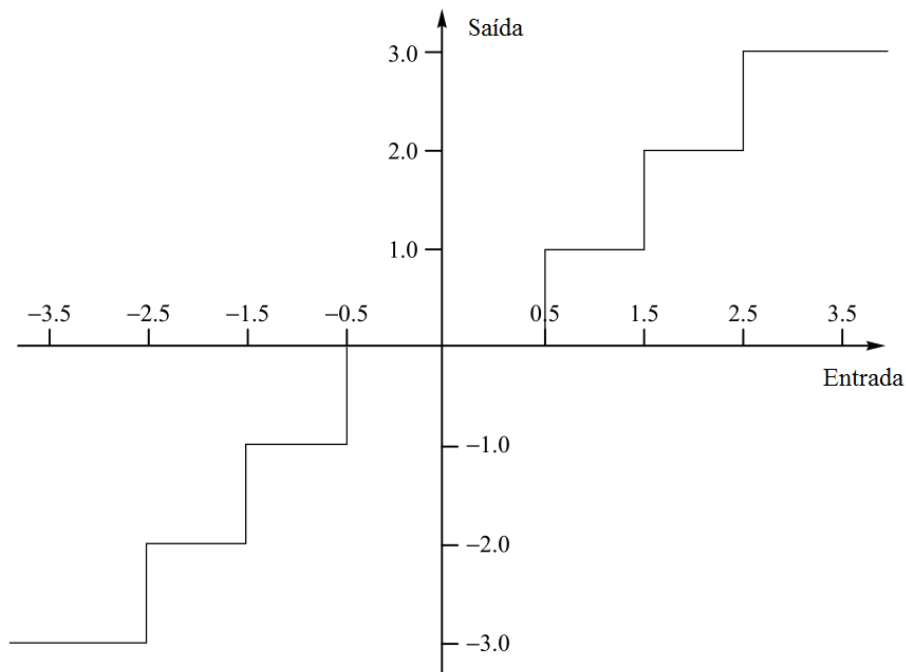


Figura 2.6: Quantização uniforme (*midtread quantizer*). Adaptado de [7].

Segundo o que temos explicado até agora, é lógico observar que, uma vez feito o processo de quantização, há informação que foi perdida e não pode ser recuperada, ao se fazer o arredondamento por exemplo. Essa característica faz que, ao querer recuperar o sinal original, exista uma distorção, porque não existe forma de saber exatamente qual era o valor original uma vez que esse valor foi quantizado. Levando isso em consideração, se pensamos num sinal qualquer a ser quantizado, existem valores que ocorrem com maior probabilidade que outros, e essa característica é explorada pelos quantizadores não-uniformes. Para poder diminuir a distorção do sinal reconstruído, esses quantizadores fazem que os intervalos de quantização sejam menores para regiões que têm mais probabilidade, e fazem que as regiões com valores que acontecem poucas vezes (baixa probabilidade) sejam maiores. O erro de quantização em intervalos pequenos é menor, e se os valores do sinal original que mais ocorrem são quantizados com intervalos pequenos, isso trará como consequência uma melhoria na fidelidade do sinal reconstruído. Essa melhora nos valores mais prováveis é paga com erros maiores nos valores menos prováveis (intervalos de quantização maiores), porém, como os valores mais prováveis tem agora menor erro, então na média, a distorção será menor que se tivéssemos usado um quantizador uniforme [7]. Na Figura 2.7 podemos observar um quantizador não-uniforme, que não possui o zero como valor de saída (midrise).

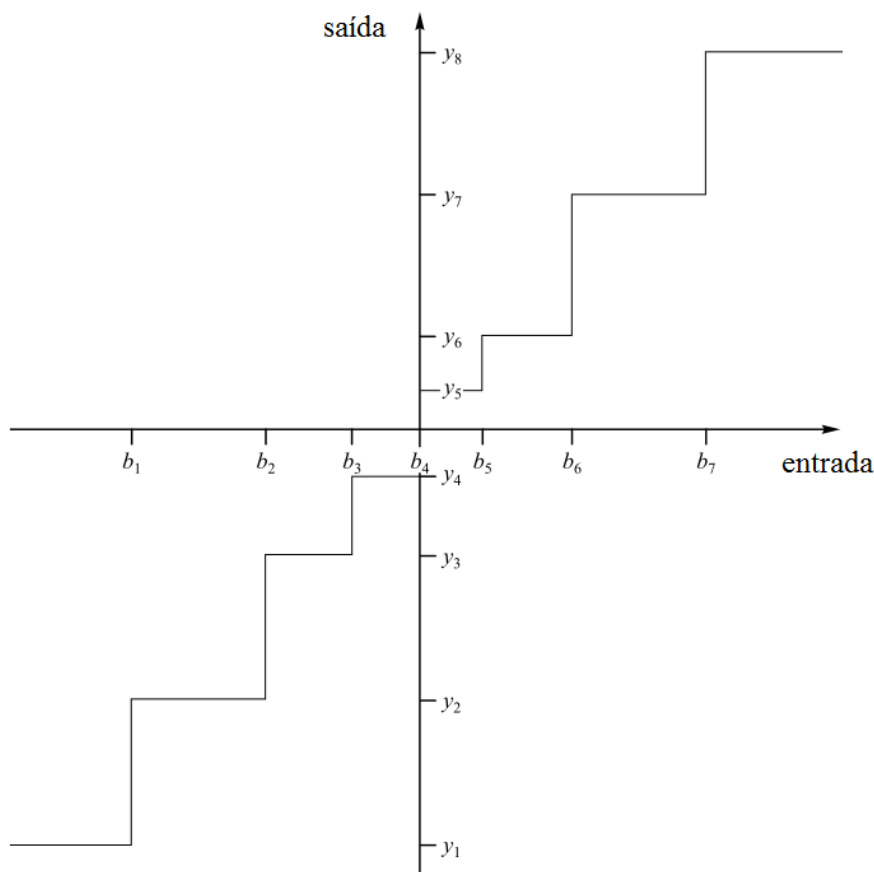


Figura 2.7: Quantização não-uniforme (*midrise quantizer*). Adaptado de [7].

O caso mais simples e mais utilizado na prática por causa da sua simplicidade na implemen-

tação é o quantizador escalar. Esse quantizador está representado por uma operação matemática muito simples da forma:

$$V_q = \left] \frac{V}{Q} \right[, \quad (2.27)$$

em que V_q representa o valor quantizado, V representa o valor de entrada e Q é o fator de quantização, e o operador $\left] \right[$ representa a operação de arredondamento. O quantizador simplesmente divide um valor de entrada V por um valor Q e arredonda o resultado. Pode-se observar que, quanto maior o valor de Q , maior o nível de quantização. O algoritmo de reconstrução então é:

$$V_{reconstruido} = V_q \times Q. \quad (2.28)$$

Vemos, pela natureza da operação matemática realizada, que temos perda de informação. Ao se fazer o arredondamento na quantização, existe informação que não pode ser recuperada na reconstrução. Por exemplo, imaginemos um $Q = 3$, para um valor de entrada $V = 10$, então a saída do quantizador seria $V_q = \left] 3,33 \right[= 3$. Ao querer reconstruir o valor original fazemos a operação $V_{reconstruido} = 3 \times 3 = 9$. Podemos observar que o valor reconstruído não é igual ao valor original, devido à perda de informação no arredondamento na quantização. Também podemos ver que, a medida que aumentamos o valor de Q , cada vez mais valores são quantizados para $V_q = 0$, ou seja que mais informação é perdida. Portanto, quanto maior é Q , maior é a perda de informação, e maior o nível de quantização. Podemos apreciar o processo descrito na Figura 2.8.

No caso da quantização vetorial, não entraremos em muito detalhe na sua explicação, uma vez que esse tipo de quantização não foi utilizada neste trabalho, e já foram explicados os conceitos necessários para entender o que será implementado no Capítulo 3. De qualquer forma, diremos que a quantização vetorial agrupa os valores da fonte em blocos ou vetores. Um vetor de valores da fonte (conjunto de valores da fonte) forma a entrada para o quantizador vetorial. Tanto no codificador como no decodificador do quantizador vetorial, existe um conjunto de vetores chamado *codebook*. Os vetores nesse *codebook* são chamados de vetores-código, e são escolhidos para representar da melhor forma o vetor fonte. Cada vetor-código tem um índice binário. No codificador, o vetor de entrada é comparado com cada vetor-código para escolher aquele vetor-código que é mais similar com o vetor de entrada. Para o decodificador saber qual vetor-código foi escolhido, se transmite ou armazena o índice binário desse vetor-código. O decodificador tem o mesmo *codebook* que o codificador [7]. Um diagrama que mostra graficamente o conceito por trás da quantização vetorial é mostrado na Figura 2.9.

2.6.3 Codificação por Entropia

O que explicamos até agora na Seção 2.6 se refere a um tratamento prévio da informação para deixar ela de uma forma que terá melhores resultados na codificação de entropia, ou chamada também codificação estatística. Uma vez que se tem a informação que se quer codificar, precisa-se gerar um código para representar essa informação. Existem códigos de tamanho fixo, por exemplo, o standard ASCII, que utiliza um código de 8 bits para representar cada caracter. Os

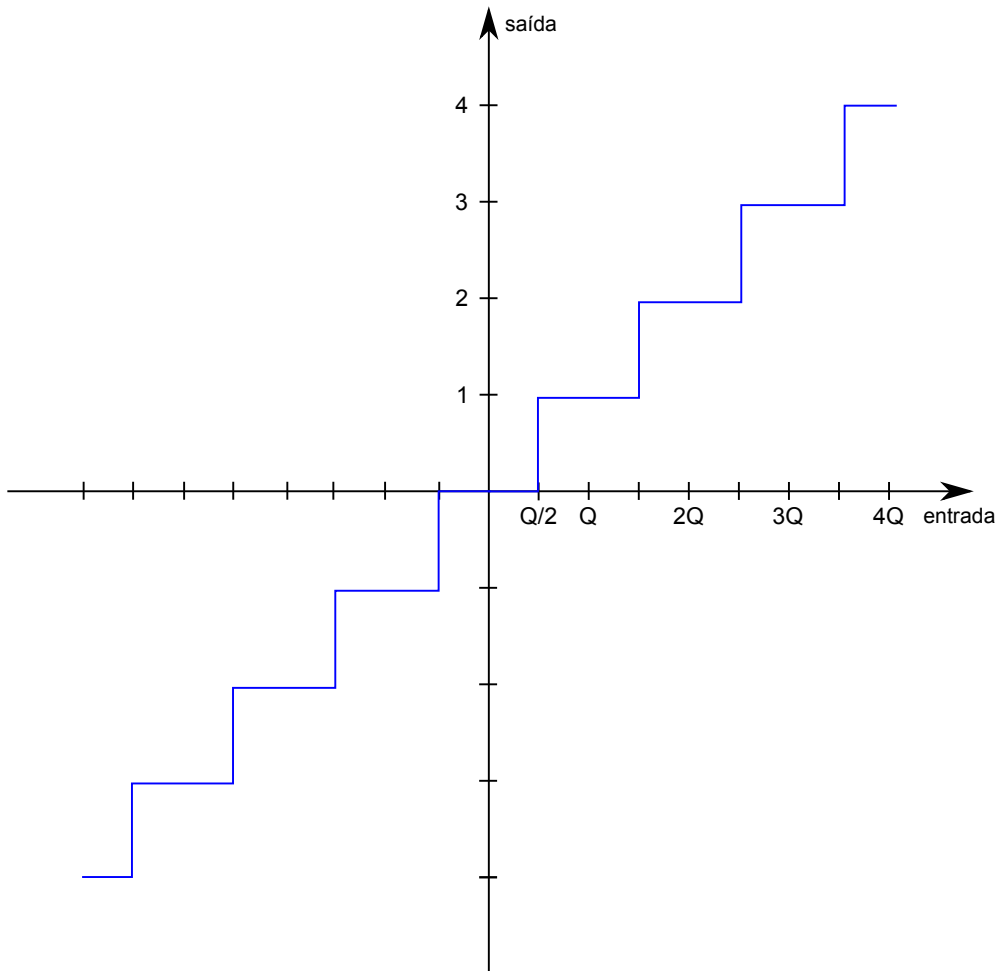


Figura 2.8: Quantização escalar.

métodos estatísticos utilizam códigos de tamanho variável, em que se atribuem os códigos mais curtos a símbolos que ocorrem com mais frequência na informação (ou seja que têm maior probabilidade de acontecer) e códigos mais longos para símbolos que acontecem com menor frequência (menor probabilidade) [6]. Esse método é um tipo de codificação por entropia. Esses códigos de tamanho variável afrontam dois problemas: atribuir códigos que podem ser decodificados univocamente (uma palavra código só pode representar um valor) e atribuir códigos que darão o mínimo tamanho médio.

Definamos agora a entropia de uma sequência de dados. A quantidade de informação contida em um símbolo de um alfabeto de tamanho n (ou seja que existem n símbolos possíveis a serem representados) é representada por:

$$H = - \sum_{i=1}^n P_i \log_2 P_i, \quad (2.29)$$

em que H representa o menor número de bits necessários para representar, em média, qualquer símbolo do alfabeto, e P_i representa a probabilidade do i -ésimo símbolo acontecer [6]. Outra informação útil que pode ser extraída daqui é que um símbolo com probabilidade P_i deveria idealmente ser representado por um código de $-\log_2 P_i$ bits, que é denominada como a auto-

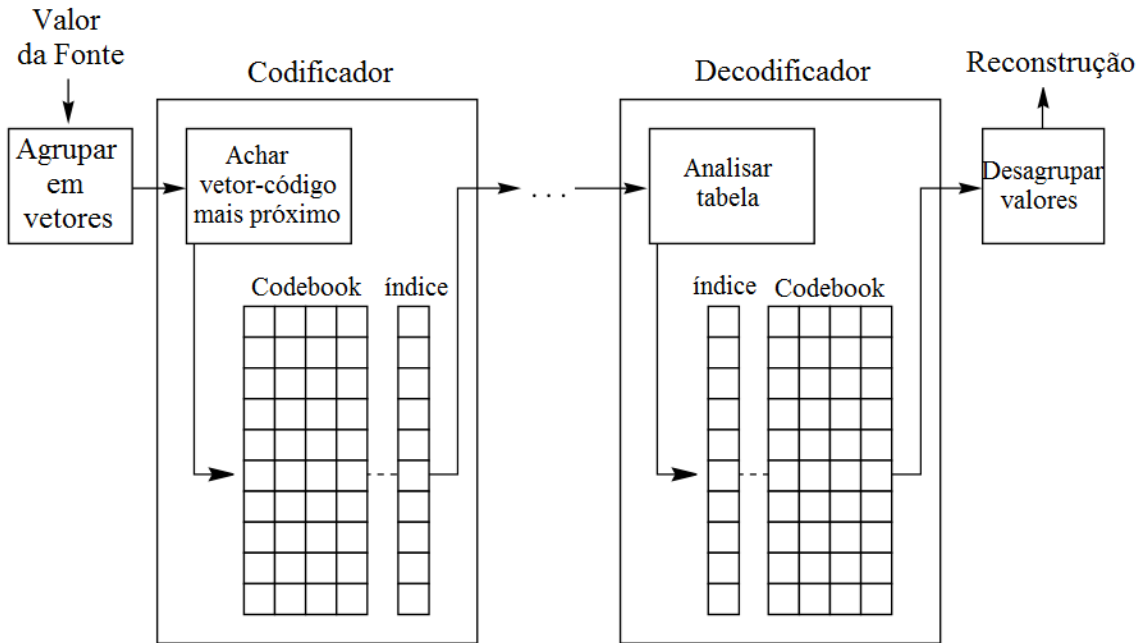


Figura 2.9: Quantização vetorial. Adaptado de [7].

informação desse símbolo [6]. O teorema principal provado por Shannon explica que uma sequência de n símbolos pode ser comprimida a nH bits, sendo esse o limite de compressão sem perdas. Também explica que existem compressores quase ótimos que podem chegar muito próximo do nível de compressão mencionado (nH).

Entre os diferentes métodos de codificação por códigos de comprimento variável se destacam a codificação Huffman e a codificação aritmética, pois são os métodos mais utilizados em algoritmos de compressão. Existem outros métodos menos utilizados como os códigos Golomb, os códigos Tunstall, códigos Shannon-Fano, e outros. Porém, explicaremos apenas os códigos Huffman e a codificação aritmética, explicando o conceito por trás dos códigos Huffman, e explicando a codificação aritmética que é a utilizada neste trabalho.

O algoritmo de Huffman constrói uma lista de todos os símbolos do alfabeto em ordem decrescente, segundo suas probabilidades de ocorrência. Logo, constrói uma árvore com um símbolo em cada folha, de baixo pra cima. Isso é feito selecionando os dois símbolos de menor probabilidade, sendo apagados da lista e substituídos com um símbolo auxiliar que representa os dois símbolos originais. Quando a lista se reduz a apenas um símbolo auxiliar, a árvore está completa [6]. Atribuindo valores 1 e 0 a cada folha da árvore, se obtém o código resultante. O explicado pode ser mais facilmente entendido no procedimento mostrado na Figura 2.10.

No exemplo da figura 2.10, a entropia do sinal é dada por:

$$H = - \sum_{n=1}^n P_i \log_2 P_i = -0,4 \log_2 0,4 - 0,3 \log_2 0,3 - 0,2 \log_2 0,2 - 0,05 \log_2 0,05 - 0,05 \log_2 0,05 \quad (2.30)$$

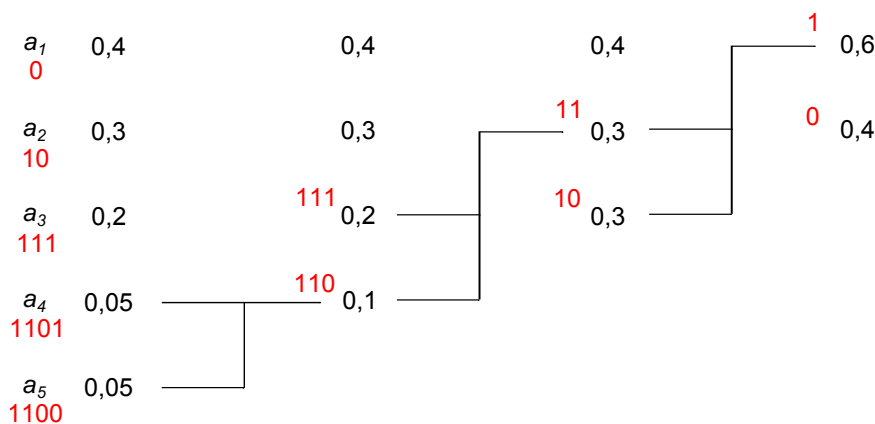


Figura 2.10: Codificação Huffman. Adaptado de [6].

$$H = 1,94643 \quad (2.31)$$

Como falamos previamente, esse valor representa o menor número de bits necessários para representar qualquer símbolo do alfabeto, em média. Fazendo o cálculo do valor em bits médio para cada símbolo no nosso código Huffman obtemos:

$$0,4 \times 1 + 0,3 \times 2 + 0,2 \times 3 + 0,05 \times 4 + 0,05 \times 4 = 0,4 + 0,6 + 0,6 + 0,2 + 0,2 = 2 \text{ bits/símbolo} \quad (2.32)$$

Podemos ver que o código Huffman conseguiu um resultado próximo à entropia, porém não conseguiu chegar a esse nível de compressão. O único caso que produz um código de tamanho variável ideal (o tamanho médio de cada símbolo é igual à entropia) é quando os símbolos têm probabilidades de ocorrência que são potências negativas de 2 ($1/2$, $1/4$, $1/8$, etc.). Isso se deve ao fato de que o método de Huffman dá um número inteiro de bits a cada símbolo do alfabeto [6]. Superando esse obstáculo, existe o método de codificação aritmética que explicaremos a seguir. Cabe mencionar que o código de Huffman é um dos métodos de compressão estatísticos mais estudados pela sua simplicidade e bons resultados, e portanto, a teoria envolvida no algoritmo se encontra bem desenvolvida. Existem códigos Huffman ternários [6], teoria sobre a escolha do código Huffman ótimo [6] (para um dado grupo de símbolos, não existe apenas um código Huffman, depende de como se atribuem os bits), teoria sobre árvores de código Huffman [6], códigos Huffman canônicos [6] e códigos Huffman adaptativos [6]. Não entraremos em detalhe nos tópicos mencionados devido ao fato de que já foi desenvolvido o necessário para entender o funcionamento e princípio da codificação Huffman.

Explicaremos agora a codificação aritmética. A codificação aritmética atribui um código à sequência de dados de entrada completo (normalmente um código longo) invés de atribuir uma palavra código diferente para cada símbolo, como no código Huffman. O algoritmo começa analisando os dados símbolo por símbolo, e utiliza a probabilidade de ocorrência de cada símbolo para ir diminuindo o intervalo de probabilidade que a sequência de símbolos que se quer enviar tem. Ou seja que, em outras palavras, utiliza a função de probabilidade acumulada para a sequência de símbolos que quer ser codificada. À medida que o intervalo diminui (aumenta a precisão do intervalo), a quantidade de decimais aumenta, e, para poder representá-los, o número de bits

necessários aumenta. A compressão é obtida devido ao fato de que símbolos com probabilidade de ocorrência alta diminuem menos o intervalo que os de baixa probabilidade, ou seja, símbolos de alta probabilidade contribuem com menos bits ao código de saída [6]. Para simplificar o entendimento do funcionamento do algoritmo, o explicaremos através de um exemplo. Imaginemos três símbolos a_1 , a_2 e a_3 , com probabilidades $P_1 = 0,4$, $P_2 = 0,5$ e $P_3 = 0,1$. O intervalo $[0, 1)$ é dividido em intervalos do tamanho da probabilidade de cada símbolo, ou seja, de 0 a 0,4 o primeiro intervalo, de 0,4 a 0,9 o segundo intervalo e de 0,9 a 1 o terceiro intervalo. Pensemos na sequência de símbolos $a_2a_2a_2a_3$ a ser codificada. O primeiro símbolo a_2 reduz o intervalo a $[0,4; 0,9)$. O segundo símbolo a_2 reduz o intervalo atual a um intervalo proporcional a $[0,4;0,9)$ do seu valor, ou seja o intervalo atual $[0,4;0,9)$ muda para $[0,6;0,85)$. O terceiro símbolo a_2 produz $[0,7;0,825)$ e o quarto símbolo a_3 gera $[0,8125;0,8250)$. O código final que esse método gera é qualquer número dentro desse intervalo. O que não fica evidente à primeira vista é onde está a compressão, ou se o algoritmo comprime próximo ao nível ideal. Para mostrar isso, vejamos um exemplo um pouco mais complexo. Imaginemos que desejamos enviar a mensagem SWISS-MISS, e as probabilidades dos símbolos são $S = 0,5$, $W = 0,1$, $I = 0,2$, $M = 0,1$ e $- = 0,1$. Então o processo de codificação aritmética dessa mensagem pode ser vista na Figura 2.11.

símbolo		cálculo dos intervalos
S	L	$0.0 + (1.0 - 0.0) \times 0.5 = 0.5$
	H	$0.0 + (1.0 - 0.0) \times 1.0 = 1.0$
W	L	$0.5 + (1.0 - 0.5) \times 0.4 = 0.70$
	H	$0.5 + (1.0 - 0.5) \times 0.5 = 0.75$
I	L	$0.7 + (0.75 - 0.70) \times 0.2 = 0.71$
	H	$0.7 + (0.75 - 0.70) \times 0.4 = 0.72$
S	L	$0.71 + (0.72 - 0.71) \times 0.5 = 0.715$
	H	$0.71 + (0.72 - 0.71) \times 1.0 = 0.72$
S	L	$0.715 + (0.72 - 0.715) \times 0.5 = 0.7175$
	H	$0.715 + (0.72 - 0.715) \times 1.0 = 0.72$
-	L	$0.7175 + (0.72 - 0.7175) \times 0.0 = 0.7175$
	H	$0.7175 + (0.72 - 0.7175) \times 0.1 = 0.71775$
M	L	$0.7175 + (0.71775 - 0.7175) \times 0.1 = 0.717525$
	H	$0.7175 + (0.71775 - 0.7175) \times 0.2 = 0.717550$
I	L	$0.717525 + (0.71755 - 0.717525) \times 0.2 = 0.717530$
	H	$0.717525 + (0.71755 - 0.717525) \times 0.4 = 0.717535$
S	L	$0.717530 + (0.717535 - 0.717530) \times 0.5 = 0.7175325$
	H	$0.717530 + (0.717535 - 0.717530) \times 1.0 = 0.717535$
S	L	$0.7175325 + (0.717535 - 0.7175325) \times 0.5 = 0.71753375$
	H	$0.7175325 + (0.717535 - 0.7175325) \times 1.0 = 0.717535$

Figura 2.11: Codificação Aritmética. Adaptado de [6].

Podemos ver que o intervalo final que representa essa mensagem fica delimitado pelos valores 0,71753375 e 0,717535. Podemos escolher um número que esteja dentro desse intervalo, por exemplo o representado pelo código 10110111101100000100, que seria nossa saída comprimida. É um código de 20 bits de comprimento. Para calcular quanto seria o mínimo necessário em bits para representar essa mensagem calculamos a probabilidade dessa mensagem, que é $P = 0,5 \times 0,1 \times 0,2 \times 0,5 \times 0,5 \times 0,1 \times 0,1 \times 0,2 \times 0,5 \times 0,5 = 1,25 \times 10^{-6}$. A auto-informação

dessa sequência esta dada por $-\log_2 P = 19,6096$ bits. Podemos ver então que 20 bits é o mínimo necessário na prática para poder codificar essa sequência, que foi a mesma quantidade de bits que o algoritmo de a codificação aritmética obteve [6].

2.7 MÉTODOS EXISTENTES DE COMPRESSÃO DE ECG

Nesta seção, faremos uma breve descrição dos métodos que se encontram na literatura para a compressão de sinais ECG, mencionando alguns métodos clássicos, e falando dos mais modernos. Dividiremos os métodos atuais nos grupos métodos com transformadas de wavelets, métodos por dicionário, métodos 2D, métodos com compressed sensing, métodos sem perda e outros. Não serão mostrados os desempenhos dos algoritmos explicados nessa seção devido à falta de um resultado comum entre eles para poder comparar os desempenhos. Alguns deles mostram desempenhos na média para todo o banco de dados MIT-BIH, alguns mostram resultados para sinais isolados, alguns mostram resultados para outros sinais que não pertencem ao banco de dados MIT. No Capítulo 4 o método proposto no presente trabalho será comparado com vários métodos mencionados aqui, com os quais foi possível comparar devido a uma forma de avaliar igual em todos eles.

Como mencionamos na Seção 1.2, existem os métodos clássicos AZTEC [10], FAN/SAPA [11], CORTES [12] e Turning Point (TP) [13], nos quais se opera sobre o sinal no domínio do tempo. No caso do AZTEC (Amplitude-Zone-Time-Epoch-Coding), o conceito do algoritmo consiste em amostrar o sinal ECG a cada 2 milissegundos, medir o nível de voltagem nessa amostra, e ao medir a próxima amostra, se calcula um diferença. Se essa diferença não supera um limiar escolhido pelo usuário, pode-se representar esse período do sinal com um valor constante no ponto meio entre as duas voltagens. Se a diferença supera o limiar escolhido, esse período se representa com outro valor constante, no ponto meio das novas duas voltagens amostradas. Pode-se observar o ECG obtido através do algoritmo AZTEC na Figura 2.12. O algoritmo TP analisa a tendência dos pontos amostrados e guarda só um valor de cada dois valores amostrados (que melhor represente o sinal). Mantém os pontos dos picos e os vales nos quais o sinal muda seu comportamento (são os chamados *turning points*). O TP consegue manter os atributos do sinal diminuindo pela metade a taxa de amostragem (compressão de 2:1). O algoritmo CORTES (Coordinate-Reduction-Time-Encoding System) utiliza uma combinação de AZTEC e TP, em que se aplica TP às partes de alta frequência clinicamente importantes do sinal, e AZTEC é utilizado nas regiões iso-elétricas do sinal ECG para atingir redução dos dados grande com pouca distorção do sinal. No caso do FAN/SAPA (Scan-along polygonal approximation) são dois algoritmos desenvolvidos em diferentes épocas (FAN é anterior) que têm o mesmo funcionamento. A idéia neles é representar o ECG através de uma série de segmentos de linhas retas, e pode se escolher uma tolerância ϵ escolhida pelo usuário que serve para ter um limite no desvio entre o sinal original e as linhas retas que o representam.

Com relação aos métodos com transformadas wavelets, encontramos vários exemplos. Isso se

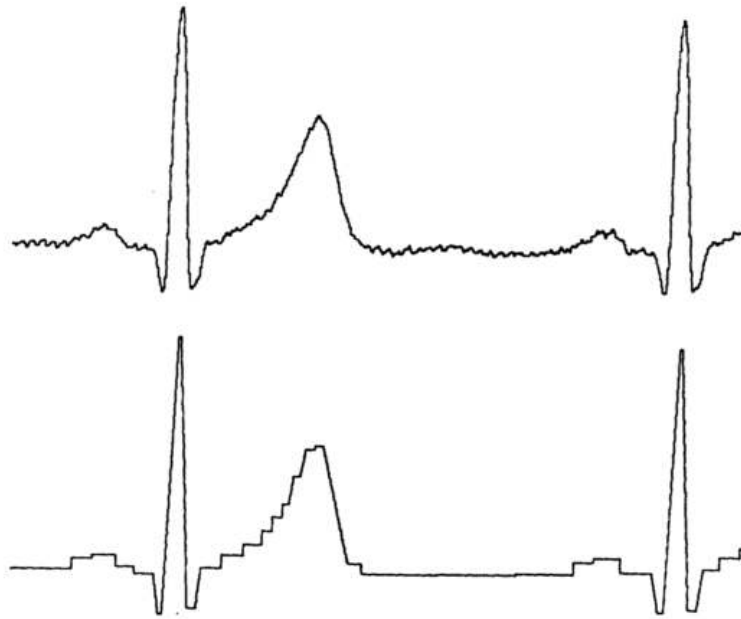


Figura 2.12: Exemplo de um ECG original e reconstruído utilizando o algoritmo AZTEC. Fonte: [10].

deve à recente popularidade da transformada wavelet devido ao seu poder de transformação e concentração de energia, e porque os métodos de compressão ECG desse tipo obtêm resultados muito satisfatórios sem uma elevada complexidade. Devido ao fato dos métodos serem de compressão por transformação, eles têm o diagrama básico mostrado em 2.6.1 composto por transformação, quantização (que onde ocorre a perda de informação) e codificação de entropia. Na compressão por métodos de transformação apenas os coeficientes que contêm a maior parte da energia são de interesse para armazenar. Por isso, uma melhor compressão é obtida se toda a energia do sinal pode ser concentrada em poucos coeficientes.

No método de Agulhari [46], se procura achar a melhor função wavelet para comprimir o sinal através de um procedimento de otimização. Ou seja, não se utilizam famílias wavelets existentes, mas se formula um problema de otimização para achar uma família wavelet que otimize a compressão (concentre a energia no menor número de coeficientes possível). Esse processo se realiza para cada sinal que se quer comprimir. Utiliza uma quantização adaptativa na qual os coeficientes de cada sub-espaco são quantizados usando um número de bits diferentes, e no processo de codificação se baseia no método de run-length-encoding (RLE).

No caso de Arvinti [21], se utiliza a DWT e se estuda o desempenho da família Daubechies. Após a transformação, se utiliza uma limiarização dura onde os valores abaixo de um limiar escolhido pelo usuário são zerados.

Em Chan [22], também se usa a DWT utilizando um esquema de banco de filtros, e utilizam a família Bi-ortogonal 5.3. Após isso, se calcula a resolução em bits dos coeficientes transformados para utilizar a quantização denominada preservação do campo de bits, em que se um coeficiente tem 8 bits, por exemplo, e se se querem utilizar 6 bits para representar esse valor, se utilizam os 6

bits mais significativos. Após esse processo ser realizado, é utilizado um RLE.

Em Chen [23], é aplicada uma DWT ao sinal, se realiza uma quantização escalar de zona-morta, se dividem os coeficientes quantizados em 4 grupos e, por último, se realiza uma codificação aritmética adaptativa. A quantização escalar de zona-morta consiste em zerar os coeficientes que estejam abaixo de um limiar, e os coeficientes com magnitudes maiores que esse limiar serão quantizadas ainda utilizando um quantizador midtread uniforme (como o explicado na Seção 2.6.2). Após esse passo, se dividem os coeficientes quantizados em 4 tipos de sequências de símbolos para assim ter sequências mais curtas e poder realizar uma codificação de entropia mais eficiente. Se utiliza um codificador aritmético adaptativo.

Chompusri [24] se baseia na segmentação de pulsos cardíacos. Se realiza um processo para a separação dos complexos QRS. Essa segmentação é feita porque o método de compressão utiliza o resíduo entre o ECG original e um ECG de referência, mas para poder fazer a diferença entre ambos se precisa ter janelas contendo o QRS dentro, motivo pelo qual um método eficiente de segmentação de pulsos é necessário. Após ter sido calculada a diferença entre o sinal original e o de referência, se faz uma DWT utilizando db2 do resultado, e se realiza uma limiarização. O resultado é codificado utilizando codificação Huffman.

Ebrahimzadeh [25] utiliza DWT para transformar, particularmente as famílias Daubechies, Symlets e Biortogonal, com 5 níveis de decomposição. Se realiza uma limiarização de 3 níveis, em que o primeiro limiar é a metade do valor do coeficiente transformado máximo, e todos os coeficientes abaixo desse valor são zerados. Após isso, se utiliza um limiar da metade do valor do anterior, e por último se utiliza outro limiar da metade do valor do último. Se realiza um EZW (embedded zero-tree wavelet encoding) no qual se explora a relação dos coeficientes wavelet de cada nível de decomposição entre si, e por último se realiza uma codificação Huffman.

Hung [26] estuda o controle da qualidade do sinal reconstruído. Utiliza o esquema tradicional de quantizar os coeficientes wavelet, mas utiliza o denominado programa de evolução (EP), que é uma ferramenta comumente utilizada para achar soluções ótimas para sistemas não-lineares multi-variável. Assim, consegue selecionar um PRD a ser atingido, e o algoritmo varia os valores de quantização através do EP para achar a solução.

No trabalho de Kumar [47], para a decomposição se utiliza uma análise multi-resolução (MRA) usando uma pirâmide gaussiana, e ao resultado dessa decomposição é aplicada uma DWT. Se realiza uma limiarização do resultado e por última se usa RLE.

Swarnkar [48] realiza uma DWT do sinal utilizando as mesmas transformadas utilizadas no presente trabalho. É aplicado um predição linear (*linear predictive coding* - LPC) aos coeficientes transformados, na qual se tenta achar o próximo valor do sinal utilizando valores anteriores do sinal. Se minimiza a diferença entre o sinal transformado e a predição linear através de um processo de otimização. A essa diferença é aplicado uma limiarização de vários tipos, se realiza uma quantização e por último um RLE.

Rajankar [49] utiliza a família wavelet Bi-ortogonal4.4 com 5 níveis de decomposição para

transformar. Se realiza uma limiarização dos coeficientes, utilizando o conceito de teste de hipótese, na qual se testam vários valores do limiar para um nível de significância do coeficiente que se quer zerar. Após isso, RLE é aplicado, e codificação Huffman.

Hilton [50] realiza uma transformada wavelet do sinal, para realizar um EZW dos coeficientes. O EZW prioriza a transmissão da localização e o sinal dos coeficientes wavelets significativos, codifica as localizações dos coeficientes não significantes explorando a similaridade da transformada wavelet em cada nível de decomposição e uma aproximação da magnitude dos coeficientes significativos através de transmissão ordenada dos planos de bits dos coeficientes. Para terminar, realiza uma codificação aritmética adaptativa.

O algoritmo Al-shrouf [51] é similar ao de Swarnkar [48]. É baseado na compressão dos resíduos linearmente preditos dos coeficientes wavelet. No algoritmo, se realiza uma DWT, e os coeficientes wavelet resultantes são linearmente preditos. Para o sinal resultante é realizado um processo de limiarização e logo é comprimido usando RLE e codificação Huffman. O erro correspondendo à diferença entre os coeficientes wavelet e os coeficientes preditos é minimizado para obter o melhor predictor.

Tohumoglu [52] também utiliza EZW, mas uma versão modificada do mesmo. No caso desse algoritmo, é um EZW multi-iterativo, no qual se explora a relação entre coeficientes de diferentes níveis de decomposição, mas também se utilizam múltiplos limiares para o processo de limiarização. Por último, um codificador de Huffman é aplicado.

Em Boukhenoufa [53] é utilizado o método SPIHT (Set Partitioning in Hierarchical Trees) que é um método similar ao EZW, porém ordena cada nível de decomposição wavelet em uma árvore onde existirá uma relação pai-filho entre coeficientes. Nesse trabalho também se utiliza o VKTP (Vector K-Tree Partitioning), que é um método de codificação sem perdas para codificar entropia.

No trabalho de Hossain [54], se utiliza uma transformação através da DWT. É utilizada uma técnica de limiarização baseada na eficiência de empacotamento de energia (EPE) na qual se analisa cada nível de decomposição wavelet de acordo com o nível de concentração de energia que possui cada um deles e, de acordo com isso, se limiariza de forma diferente cada um deles.

Nos métodos 2D, costuma-se utilizar um esquema similar ao de métodos por transformada, só que nesse caso se organizam os dados, podendo eles serem transformados, ou o próprio sinal ECG, de uma forma tal que o resultado seja uma matriz de duas dimensões. Após ter feito essa organização dos dados, se realiza o processamento sobre eles, utilizando algum método como quantização ou limiarização, normalmente explorando a relação entre cada fila da matriz, ou entre cada coluna, e codificando o resultado.

Em Chou [17], se realiza um pre-processamento do sinal ECG 1D. Se detectam os complexos QRS e se segmenta o sinal ECG, tomando 130 amostras prévias ao complexo QRS e redistribuindo cada parte segmentada em filas que são parte de uma matriz 2D. Posteriormente, se realiza uma classificação de períodos em que se classificam os segmentos de pulso cardíaco da matriz

de acordo com seus períodos em ordem ascendente ou descendente. Como passo seguinte, se normalizam os comprimentos de cada fila mediante dois métodos. Por último, se comprime o resultado utilizando um codificador JPEG2000.

Rezazadeh [55] realiza um pre-processamento similar a Chou [17] no sinal ECG para conseguir a matriz 2D que representa os pulsos cardíacos. Após ter conseguido a matriz, se aplica uma transformada wavelet a essa matriz, se explora a compressão de energia por sub-banda, tendo em conta o EPE para poder quantizar os coeficientes. Por último, se utiliza o algoritmo SPIHT (será explicado em um dos métodos 2D) para transmitir o resultado.

Wang X. [15] propõe um método no qual se realiza a detecção dos complexos QRS do ECG, e se constrói a matriz 2D a partir dessa detecção, com normalização dos comprimentos. É aplicada um transformada wavelet 2D ao resultado e, por último, se realiza uma quantização vetorial (VQ), como a explicada na Seção 2.6.2, às sub-bandas resultantes.

Lu [56] propõe pela primeira vez o algoritmo SPIHT para compressão de sinais ECG, que até esse momento era utilizado para compressão de imagens estáticas. Quando se realiza uma decomposição wavelet, se geram diferentes níveis de composição, onde o nível com menos componentes representa a baixa frequência, e cada nível de decomposição posterior possui o dobro de coeficientes, ou seja que existe uma relação pai-filho na qual para um coeficiente do nível de decomposição i existem dois coeficientes do nível de decomposição $i + 1$ que correspondem ao coeficiente do nível i . Levando esse princípio em consideração, o algoritmo realiza a transformada wavelet sobre o sinal, usando a bi-ortogonal 9.7, e re-ordena os níveis de decomposição como pode se ver na Figura 2.13, em que pode-se ver a relação pai-filho entre cada nível de decomposição. Após isso se escolhe um limiar como sendo a metade do valor em bits do coeficiente de maior amplitude. No exemplo é 59, então o limiar escolhido é 32 (2^5), e se zeram os coeficientes menores a esse limiar seguindo uma ordem através da árvore como mostrada na figura, recalculando o limiar para coeficientes menores que ele.

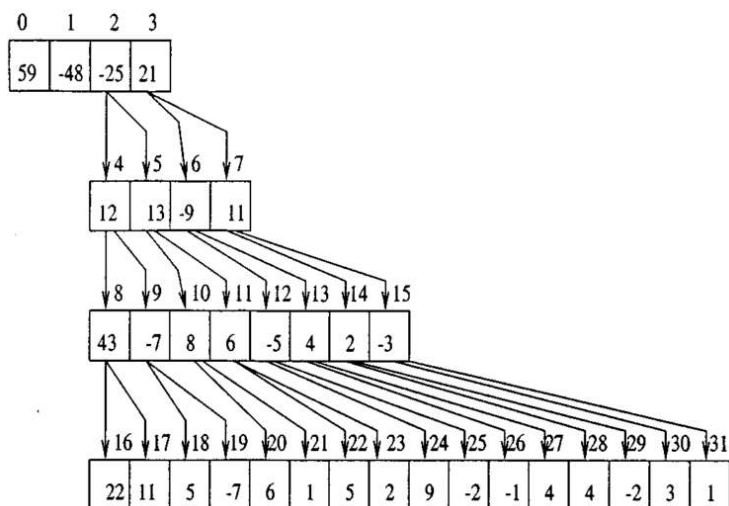


Figura 2.13: Exemplo de decomposição wavelet para 32 coeficientes com relação pai-filho para cada nível de decomposição. Fonte: [56].

Bilgin [57] foi o primeiro a propor o codec JPEG2000 como uma ferramenta para comprimir sinais de ECG. O algoritmo consiste na detecção dos pulsos cardíacos, re-ordenamento dos pulsos numa matriz 2D, e normalização dos comprimentos de cada pulso (fila). Após isso, é aplicado o codec JPEG2000.

Sahraeian [58] utiliza o pre-processamento típico para obter uma representação 2D do sinal de ECG. Após isso, é aplicado o algoritmo SPIHT à matriz. O resultado é decodificado se obtendo um sinal ECG reconstruído, e esse resultado é subtraído do sinal ECG original. Tendo o resíduo, se aplica uma quantização vetorial a ele. Por último é aplicada uma codificação aritmética ao resultado.

Outros métodos aplicados a compressão, em várias áreas, não apenas para sinais biomédicos, são os métodos por dicionário. Nesses métodos, se forma um dicionário composto por átomos que tentaram representar da melhor forma possível o sinal que se quer transformar. O dicionário busca possuir as melhores bases possíveis para que, quando se queira transformar o sinal, ele possa ser representado pelo menor número possível de coeficientes e, portanto, ter uma concentração da energia grande.

Brito [59] realiza uma segmentação do sinal ECG em segmentos do mesmo comprimento e normalizados. Cada segmento tem uma forma de U, pois se toma cada segmento entre dois picos R do complexo QRS. Para representar um segmento S, se procura segmentos representativos que formam parte de um dicionário que foi criado a partir de aproximações utilizando curvas cúbicas de Bezier. Cada segmento do dicionário é um componente D. Uma vez que se achou o D para qual a função de correlação com S é máxima, manda-se o índice de D para saber onde achá-lo no dicionário.

Em Lee [60], se treina um dicionário sobre-completo utilizando o algoritmo de aprendizado de dicionário K-SVD. Os átomos do dicionário são versões de segmentos QRS do sinal de ECG. Uma vez treinado o dicionário, tem que se obter a representação esparsa do sinal (transformá-lo), utilizando alguns átomos do dicionário como funções de base, e as soluções para uma representação esparsa usando o dicionário sobre-completo podem se achar usando MP, OMP ou LASSO. No caso de Lee, foi usado o OMP. Para terminar de gerar o código final, mandam-se as posições dos átomos do dicionário.

Wang C. [61] realiza um processo muito similar ao realizado em Lee. Utiliza K-SVD para o aprendizado do dicionário utilizando ECGs da base de dados MIT-BIH. Após isso, a combinação de átomos do dicionário que melhor representa o sinal que se quer comprimir é achada usando OMP. Logo se obtém uma transformada do sinal ECG que possui menos coeficientes que uma transformada convencional. Se enviam os coeficientes e o dicionário utilizado, que, para sinais ECGs de suficiente duração, é desprezível no nível de compressão final.

Nos métodos com *compressed sensing*, se realiza a análise de um ponto de vista da aquisição do sinal ECG, e se observa a compressão desde o ponto de vista da amostragem dele. São poucos os trabalhos estudando esse tipo de compressão, pois, na atualidade, é feito mais com fins de pesquisa e estudo que pelos resultados obtidos. *Compressed sensing* (CS) reduz os recursos ne-

cessários na aquisição do sinal. A natureza sub-nyquist do processo de amostragem em CS produz menos medições digitais que Nyquist, e tira a necessidade de compressão do sinal após a amostragem, conseguindo simultaneamente amostrar e comprimir o sinal. No trabalho de Craven [62], se estudam diferentes trabalhos que aplicam CS contra técnicas tradicionais de compressão de sinais ECG, como técnicas por transformação, e se conclui que as técnicas já existentes superam amplamente a performance do CS para a compressão de sinais ECG. Em Mamaghianian [63], também se estuda a performance de CS para compressão de sinais ECG, mas desde um ponto de vista de consumo de energia, especificamente para sensores corporais sem-fio de ECG. Conclui mostrando que, apesar de CS mostrar um desempenho de compressão inferior aos métodos com DWT, a complexidade do processo de compressão é menor, e o tempo de execução do CPU permite que CS tenha um desempenho melhor que DWT em termos de eficiência energética.

Técnicas de compressão sem perdas também foram estudadas, mas não em profundidade, considerando que na compressão de ECGs é mais importante conseguir uma compressão elevada, podendo aceitar um pouco de perda de informação. Em Cho [64], se estuda um algoritmo sem perdas para a transmissão em tempo real de sinais ECG. São diferentes os processos de compressão de transformada, por exemplo, nos quais se realiza uma aquisição completa do sinal ECG, e após isso é realizado o algoritmo de compressão para posterior armazenamento. No caso de Cho [64], se propõe um algoritmo para poder enviar o ECG em tempo real. Seu algoritmo é baseado no V.42 bis que é uma variação do algoritmo Lempel-Ziv-Welch (LZW) de compressão por dicionário sem perdas.

No que se refere à compressão de sinais, não é necessário apenas usar métodos propostos, ou técnicas amplamente conhecidas. Existem casos de métodos para compressão de sinais ECG no que se utilizam outros métodos com técnicas diferentes. No algoritmo de Mukhopadhyay [65], é utilizado um sistema híbrido utilizando codificação ASCII. Se realiza uma detecção dos picos R e se usam 60 ms prévios e posteriores a esse pico para comprimir sem perdas utilizando codificação ASCII. Para as outras amostras do ECG, é realizada também uma codificação ASCII, porém não se utilizam todas as amostras, sendo escolhidas apenas algumas amostras úteis para serem codificadas.

3 METODOLOGIA

No presente capítulo são detalhadas as implementações realizadas para compressão de sinais de ECG. Também são feitas comparações sistemáticas das diferentes opções de famílias wavelets para realizar a transformação, além de comparações dos parâmetros dos algoritmos desenvolvidos. São discutidas ainda as figuras de mérito utilizadas e as formas de análise dos resultados obtidos.

3.1 DESENVOLVIMENTO DO SISTEMA DE COMPRESSÃO DE SINAIS DE ECG

Na Figura 3.1, podemos observar um diagrama geral do sistema, em que estão incluídas as etapas de métodos de transformação, o algoritmo proposto para quantização e o estágio de codificação de entropia. Segundo podemos ver no diagrama, o algoritmo desenvolvido respeita o diagrama básico de um codificador por transformada (ver Figura 2.5 na Seção 2.6.1). No codificador, se realiza uma transformada wavelet sobre o sinal ECG no domínio do tempo. Após isso, o sinal é dividido em sub-bandas, e em seguida se realizará uma quantização escalar com diferentes valores de quantização para cada sub-banda, junto com uma quantização que segue o princípio de correção gamma em que os coeficientes de maior tamanho serão quantizados em menor medida que os coeficientes de menor tamanho. Após ter realizado esses passos, o algoritmo realiza um run-length encoding (RLE) da sequência resultante e por último se realiza uma codificação aritmética para obter o código final, eliminando qualquer redundância que possa estar presente na sequência. O processo de recuperação do sinal é o processo inverso ao mencionado. Se recupera o código fazendo uma decodificação aritmética, se recupera o sinal transformado quantizado aplicando um decodificador RLE, se aplica uma quantização inversa (os valores exatos originais não poderão ser recuperados devido ao fato de que no processo de quantização na codificação houve informação que foi perdida e não pode ser recuperada) e por último se faz uma transformada wavelet inversa para recuperar o sinal. Nas seguintes seções, explicaremos cada estágio do diagrama por separado.

3.1.1 Transformada de wavelets

Para o melhor entendimento do explicado nesta seção, fazer referência ao explicado na Seção 2.4. Nesta instância, se realiza uma decomposição wavelets do sinal em N níveis. No trabalho, foram utilizadas diferentes famílias wavelets (Daubechies, Symlets, Ortogonal, Biortogonal e Coiflets) na decomposição. A WT representará o sinal em um domínio transformado, em que podem-se observar diferentes resoluções de frequência nos instantes que elas ocorrem. Na Figura 3.2 pode-se ver um sinal ECG no domínio do tempo, e sua respectiva DWT (com a família Daubechies 4) na Figura 3.3, em que podemos ver os diferentes níveis de decomposição, que nesse caso

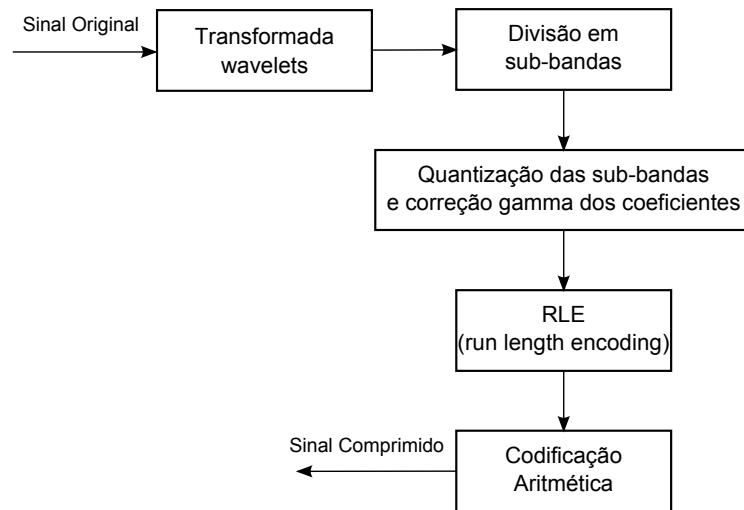


Figura 3.1: Diagrama de blocos do sistema implementado.

são 5. Cada nível representa uma faixa de frequência diferente, e podemos observar o sinal completo em cada nível, ou seja, podemos observar quando ocorre uma faixa de frequência particular através da duração completa do sinal. A diferença entre cada nível é o intervalo de duração dele, então, podemos observar que para as altas frequências, que seria o primeiro nível da direita para a esquerda, indicado por w_4 (o primeiro passa-alta no processo de banco de filtros, ver na Figura 3.4), temos uma boa resolução no tempo, devido ao fato de que o intervalo de duração para as altas frequências é o maior, significando que é possível indicar em que instante ocorreu um componente de alta frequência com a maior precisão no tempo. Para o caso das baixas frequências, que seriam os outros níveis à esquerda até chegar ao eixo ordenado $(w_3, w_2, w_1, w_0, v_0)$, vemos que o intervalo de duração de cada nível é cada vez menor, significando que temos menos resolução no tempo, porém, mais precisão na frequência pois cada nível subsequente é o resultado de mais uma filtragem passa-baixa/passa-alta e, portanto, está mais delimitado em frequência. O processo de filtragem para 5 níveis pode ser observado na Figura 3.4.

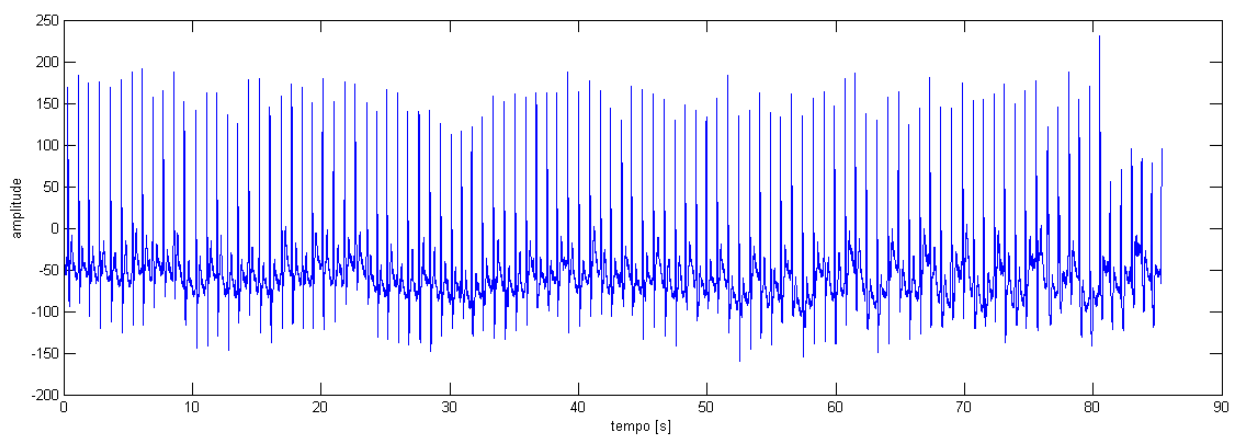


Figura 3.2: Sinal 102 da base de dados MIT-BIH no domínio do tempo. Fonte: [5].

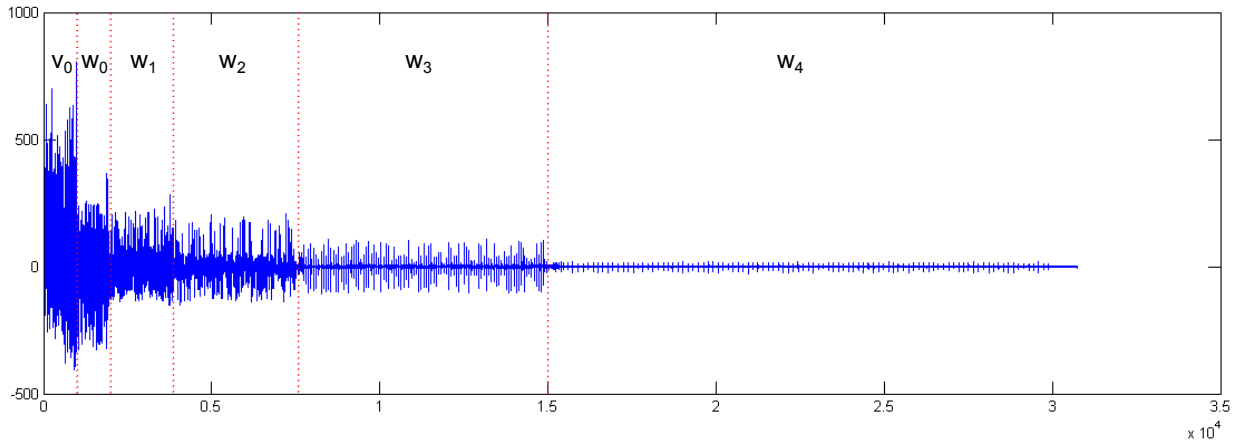


Figura 3.3: Decomposição wavelets do sinal 102 em 5 níveis com a família db4 com cada nível delimitado.

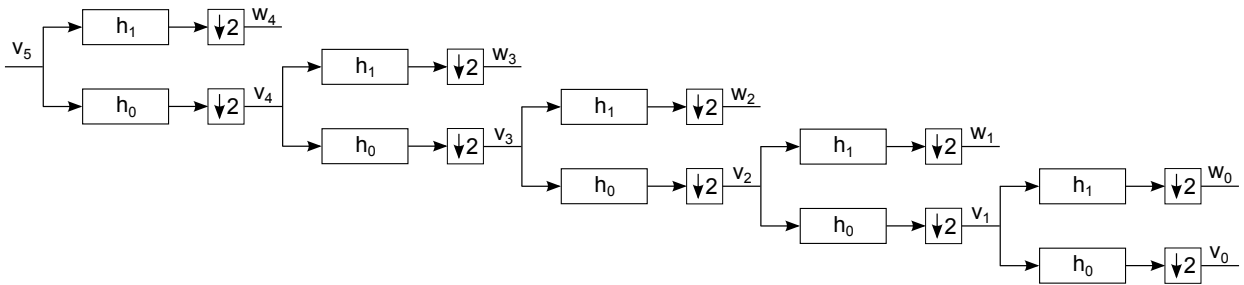


Figura 3.4: Decomposição wavelets do sinal 102 em 5 níveis representado por banco de filtros.

Podemos observar que a amplitude em w_4 é muito menor que em v_0 . Isso se deve a que a família *db4* é uma boa escolha para definir o sinal de ECG e, portanto, concentra a maior quantidade de energia nas baixas frequências e em poucos coeficientes. Isso vai permitir poder perder informação nos coeficientes em w_4 sem isso significar uma perda grande na fidelidade do sinal reconstruído após a compressão. Isso é um efeito desejado, pois poderão ser quantizados fortemente os coeficientes de alta frequência, o que resultará em uma boa compressão, sem gerar distorção alta no sinal recuperado. A idéia principal nesse estágio é obter uma representação o mais esparsa possível do sinal, utilizando uma família de wavelets adequada, para que a energia dele esteja concentrada no menor número possível de coeficientes, podendo assim quantizar os coeficientes que menos representam o sinal (os de alta frequência), sem isso gerar uma grande perda de fidelidade no sinal recuperado.

3.1.2 Divisão em sub-bandas

Neste passo, divide-se o sinal no domínio transformado em M sub-bandas de igual largura. Essa divisão em sub-bandas será útil posteriormente para realizar a quantização. Na Figura 3.5, pode-se ver uma divisão da transformada do sinal 102 em 8 sub-bandas a modo de exemplo. Essa divisão é útil pois permite que seja possível aplicar um nível diferente de quantização para cada

sub-banda, que, como temos visto anteriormente, será de grande utilidade, porque é do nosso interesse aplicar mais quantização às componentes de alta-frequência, e menos quantização às componentes de baixa frequência. No trabalho, após testar diferentes valores de M , se definiu um valor de 32 sub-bandas para a decomposição, pois se observa uma melhora no desempenho a medida que se aumentam as sub-bandas (a sintonia de quantização para cada sub-banda pode ser mais fina) porém a partir das 32 sub-bandas, esse desempenho não aumenta (é assintótico), motivo pelo qual se decidiu definir a divisão das sub-bandas em 32.

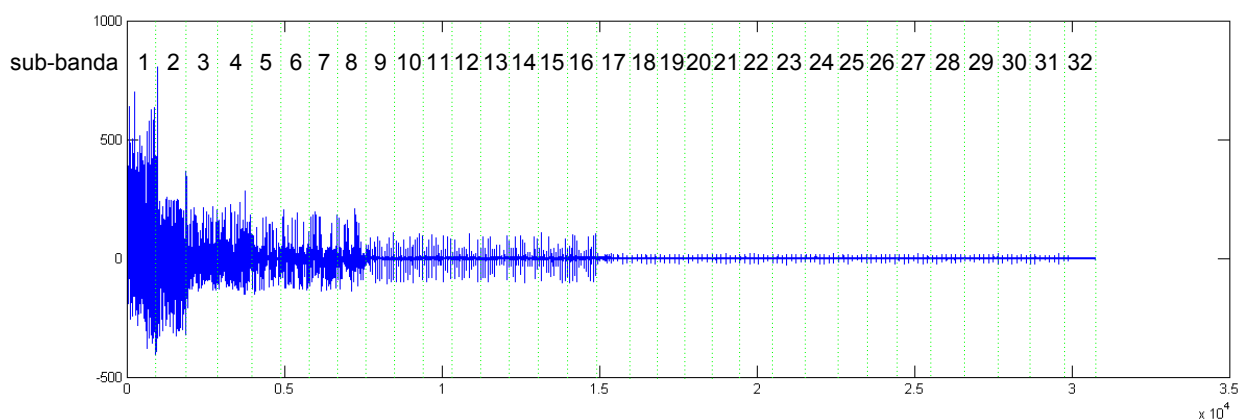


Figura 3.5: Divisão em 8 sub-bandas da transformada wavelet db4 do sinal 102.

3.1.3 Quantização das sub-bandas e correção gamma dos coeficientes

Na quantização por sub-bandas, é aplicado um nível de quantização diferente para cada sub-banda. Lembrando que a intensidade de quantização vem dada pelo fator Q da equação 2.27 na Seção 2.6.2, a maior intensidade de quantização é dada para a última sub-banda (altas-frequências) e as intensidades de quantização vão diminuindo até chegar a uma quantização nula (não tem quantização) que é dada à primeira sub-banda (frequência-baixa). Isso é devido ao fato de que, segundo a decomposição de wavelets utilizada, os coeficientes transformados representam diferentes frequências do sinal, onde a baixa-frequência é representada pelos coeficientes transformados próximos ao eixo das ordenadas (primeiras sub-bandas) e a alta-frequência é representada pelos coeficientes nas últimas sub-bandas. Se a representação wavelet for adequada, ou seja, quando a wavelet escolhida representa fielmente o ECG, os componentes transformados de baixa-frequência conterão a maior parte da energia do sinal e, portanto, serão menos quantizados, o que resulta em menor perda de informação. Como os componentes transformados de alta-frequência possuem pouca informação a respeito do sinal original, podemos quantizar eles mais fortemente, sem perder muita informação, adquirindo mais compressão.

Como temos visto, se a função de base escolhida (wavelet) é uma boa representação do sinal, o processo de transformação dará coeficientes de amplitude grande onde existe correlação com a frequência da wavelet, e dará coeficientes de pouca amplitude onde há pouca correlação. Ou seja, intuitivamente, os coeficientes de grande amplitude possuem mais informação representativa do

sinal, e os coeficientes de pequena amplitude possuem menos informação representativa do sinal, o que significa que, no que se refere à compressão, é desejável quantizar menos os coeficientes de grande amplitude (porque eles representam mais o sinal) e quantizar mais os coeficientes de pequena amplitude (porque eles representam menos o sinal), independentemente da sub-banda na qual eles se encontrem. Para isso é aplicado uma redistribuição dos valores dos coeficientes, similar ao processo de correção gamma utilizado em imagens.

A correção gamma foi utilizada inicialmente para corrigir imagens explorando o fato de que os humanos percebem cor e luz de forma não-linear. Possuímos mais sensibilidade para diferenças relativas entre tons escuros que para tons claros. Se a correção gamma não se fizesse, se usariam muitos valores para representar tons claros que os humanos não podem diferenciar, e poucos valores para tons escuros os quais podemos discernir com mais facilidade e, portanto, perceberíamos uma diferença na qualidade da imagem para os tons claros e escuros (essa correção varia para cada tipo de display). Se a correção gamma é realizada, essa diferença de qualidade entre tons claros e escuros não é perceptível [66, 67] (atribuem-se mais valores para tons escuros e menos para claros). Utilizaremos esse princípio de re-ordenamento não linear dos dados para obter compressão no nosso algoritmo. O processo de correção gamma pode se observar na Figura 3.6, em que v_{in} é o valor inicial do coeficiente e v_{out} é o valor do coeficiente após passar pela correção gamma. Pode-se observar que, dependendo do valor de λ , a redistribuição dos coeficientes é diferente. Se $\lambda < 1$, se expandem os valores pequenos e se comprimem os valores grandes. Se $\lambda > 1$, se comprimem os valores pequenos e se expandem os valores grandes. Isso pode ser demonstrado pensando em pequenas variações de x numa função, ou, em outras palavras, $f(x + \Delta)$. Essa expressão pode ser definida usando uma expansão em série de Taylor de primeira ordem, tal que:

$$f(x + \Delta) \approx f(x) + \Delta f'(x). \quad (3.1)$$

Para a correção gamma, $f(x) = x^\lambda$ e $f'(x) = \lambda x^{\lambda-1}$. Logo,

$$(x + \Delta)^\lambda \approx x^\lambda + \Delta \lambda x^{\lambda-1}, \quad (3.2)$$

em que temos os seguintes casos:

1. $\lambda x^{\lambda-1} = 1$, não há compressão nem expansão.
2. $\lambda x^{\lambda-1} > 1$, há expansão porque pequenas variações Δ resultarão em variações maiores $\Delta \rightarrow \Delta \lambda x^{\lambda-1}$.
3. $\lambda x^{\lambda-1} < 1$, há compressão porque pequenas variações Δ resultarão em variações menores $\Delta \rightarrow \Delta \lambda x^{\lambda-1}$.

As curvas na Figura 3.7 representam a função $\lambda x^{\lambda-1}$ para um valor de λ fixo. Podemos ver em que intervalos a correção gamma expande ou comprime os valores para um λ dado, ou seja, saberemos os intervalos exatos das curvas da Figura 3.6, em que se comprimem ou expandem os coeficientes. Podemos aproveitar o princípio por trás dessa idéia para quantizar os coeficientes wavelet, usando λ para determinar se os valores pequenos ou os grandes serão mais quantizados.

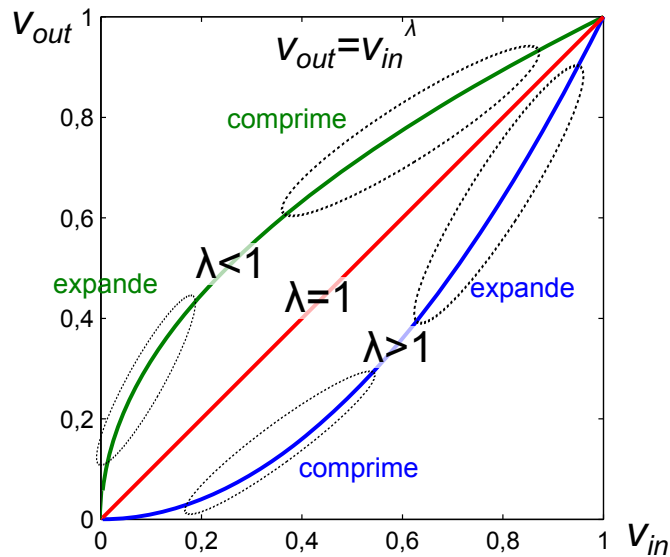


Figura 3.6: Correção gamma.

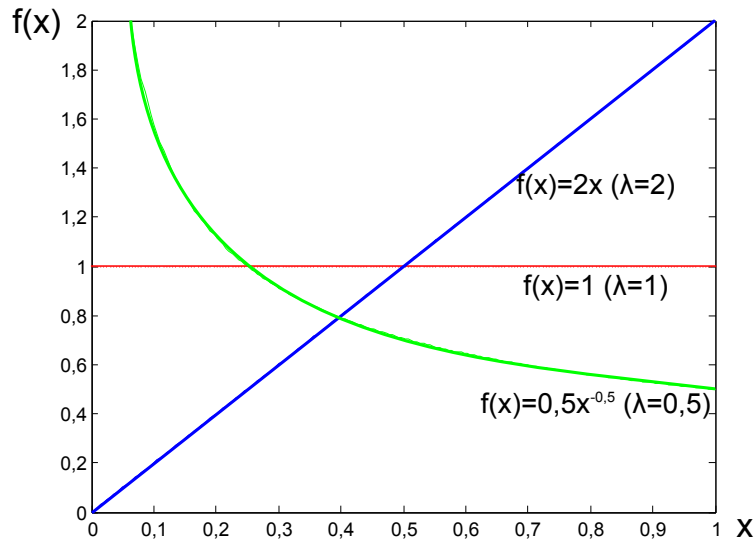


Figura 3.7: Compressão e expansão na correção gamma. A curva verde representa $\lambda = 0,5$ e vemos que $\lambda x^{\lambda-1} > 1$ (expande) no intervalo $x = (0; 0,24)$ e que $\lambda x^{\lambda-1} < 1$ (comprime) no intervalo $x = (0,24; 1)$. A curva azul representa $\lambda = 2$ e vemos que $\lambda x^{\lambda-1} < 1$ (comprime) no intervalo $x = (0; 0,5)$ e que $\lambda x^{\lambda-1} > 1$ (expande) no intervalo $x = (0,5; 1)$. A curva vermelha representa $\lambda = 1$ e vemos que $\lambda x^{\lambda-1} = 1$ (não comprime nem expande) no intervalo $(0;1)$.

Como explicamos anteriormente, é possível realizar uma quantização por sub-bandas, pois é uma estratégia muito utilizada na literatura [20, 46, 21, 22, 23, 24, 25]. A idéia do algoritmo é incorporar a quantização por sub-bandas, e explorar o princípio mostrado na correção gamma. Cada sub-banda é multiplicada por um valor proporcional à amplitude média da banda, e também é utilizada uma correção gamma nessa média por um fator λ .

Os dois processos explicados, de quantização por sub-bandas e de se utilizar uma redistribuição dos coeficientes seguindo o princípio da correção gamma, podem ser representados pela

equação 3.3.

$$C_q = \left[\frac{\text{sgn}(C)|C|^{\lambda_1^*}|C|_{max}}{|C|_{max}^{\lambda_1^*}} \frac{|\bar{C}|_i^{\lambda_2}}{|\bar{C}|_{max}^{\lambda_2}} \right], \quad (3.3)$$

em que C representa os valores dos coeficientes transformados, C_q representa os valores dos coeficientes transformados após a quantização e $\lceil \cdot \rceil$ é o operador que converte o valor de entrada para o inteiro mais próximo. $\text{sgn}(x)$ é a função que retorna o sinal de x , tal que

$$\text{sgn}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}.$$

O termo

$$\frac{\text{sgn}(C)|C|^{\lambda_1^*}|C|_{max}}{|C|_{max}^{\lambda_1^*}} \quad (3.4)$$

é a correção gamma. Representa a redistribuição dos valores de C controlada por λ_1^* , em que o efeito que queremos conseguir é que o valor máximo de C seja menos afetado pela quantização e, à medida em que os valores de C sejam menores, eles recebam mais intensidade de quantização. A quantidade de quantização que esses coeficientes recebem é controlada pelo fator λ_1^* . Como vimos anteriormente na correção gamma, sabemos que ela é dada pela equação:

$$v_{out} = v_{in}^\lambda, \quad (3.5)$$

em que v_{out} é o valor de saída e v_{in} é o valor de entrada. Essa equação é válida para valores de entrada $0 \leq v_{in} \leq 1$, mas podemos fazer o seguinte desenvolvimento para qualquer valor de v_{in} :

$$v_{out} = v_{in}^\lambda, \quad 0 \leq v_{in} \leq 1 \quad (3.6)$$

$$v_{out} = \text{sgn}(v_{in})|v_{in}|^\lambda, \quad -1 \leq v_{in} \leq 1 \quad (3.7)$$

$$v_{out} = \text{sgn}(v_{in}) \left(\frac{|v_{in}|}{M} \right)^\lambda, \quad -M \leq v_{in} \leq M. \quad (3.8)$$

Podemos ver que a equação 3.8 é igual à equação 3.4, em que $v_{in} = C$ e $M = |C|_{max}^\lambda$, com a exceção do fator multiplicativo $|C|_{max}$ da equação 3.4 que está aí para, após ser efetuada a correção gamma, desnormalizar os valores e trazê-los novamente à sua escala original (a correção gamma é feita com valores que variam entre 0 e 1, então é necessário levá-los à sua escala original após a correção). Ou seja, pelo explicado, podemos ver que o algoritmo empregado cumpre o princípio mencionado na correção gamma. Já o termo

$$\frac{|\bar{C}|_i^{\lambda_2}}{|\bar{C}|_{max}^{\lambda_2}} \quad (3.9)$$

controla a quantização por sub-bandas. Podemos observar na Figura 3.8 o efeito de quantização obtido pela quantização por sub-bandas. O numerador $|\bar{C}|_i^{\lambda_2}$ representa a média de cada sub-banda i e o denominador $|\bar{C}|_{max}^{\lambda_2}$ representa a média máxima dessas sub-bandas. O fator λ_2 controla a quantidade de quantização que cada sub-banda recebe, considerando que a sub-banda

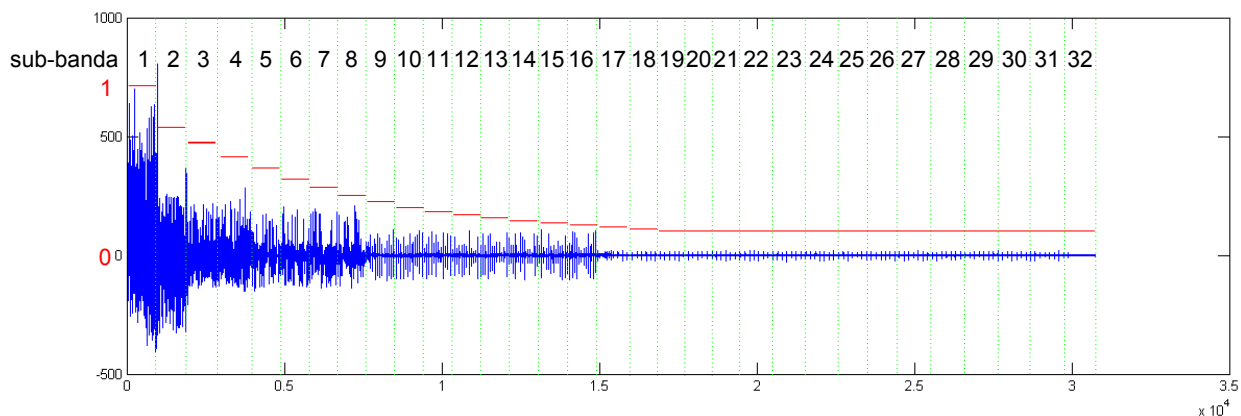


Figura 3.8: Divisão em 8 sub-bandas da transformada wavelet db4 do sinal 102 com diferentes níveis de quantização para cada sub-banda.

1, que é onde maior concentração de energia teremos na transformada e, portanto, a maior média, é multiplicada por um fator de quantização igual a 1 e portanto não recebe quantização por sub-bandas. A quantização por sub-bandas por si só consegue bons resultados, de modo que se consegue uma boa compressão sem perder muita informação valiosa. Porém, só leva em conta a energia média que cada sub-banda tem para realizar a quantização. Por isso, foi empregada também a correção gamma, para ter em conta a amplitude de cada coeficiente, independentemente da sub-banda na qual se encontra, e realizar uma quantização forte nos coeficientes pequenos (contém pouca informação) e uma quantização leve nos coeficientes grandes (contém muita informação). Levando os dois critérios em consideração, consegue-se uma melhor quantização dos coeficientes, com uma sintonia mais fina, resultando em altos níveis de compressão com pouca perda de informação. Note que podemos escrever a equação de quantização como

$$C_q = \left[\frac{\text{sgn}(C)|C|^{\lambda_1^*}}{|C|_{max}^{\lambda_1^*-1}} \frac{|\bar{C}|_i^{\lambda_2}}{|\bar{C}|_{max}^{\lambda_2}} \right], \quad (3.10)$$

e, se definimos $\lambda_1^* = 1 + \lambda_1$, então a equação vira

$$C_q = \left[\frac{\text{sgn}(C)|C|^{1+\lambda_1}}{|C|_{max}^{\lambda_1}} \frac{|\bar{C}|_i^{\lambda_2}}{|\bar{C}|_{max}^{\lambda_2}} \right]. \quad (3.11)$$

Essa re-definição da equação foi feita porque, no algoritmo implementado no trabalho, testam-se diferentes valores para uma variável λ que se define como $\lambda = \lambda_1 + \lambda_2$. Vemos que existe uma relação entre λ_1 e λ_2 , ou seja, existe uma relação entre a correção gamma e a quantização por sub-bandas, em que, à medida que uma aumenta, a outra diminui, e vice-versa. Isso é útil para o algoritmo, pois podemos encontrar o ponto ótimo de funcionamento sabendo qual a proporção de quantização por sub-bandas que foi utilizada e qual a proporção de correção gamma que foi utilizada. Fazer que $\lambda = \lambda_1 + \lambda_2$ é importante porque, quando tratamos com sinais biomédicos, é de suma importância manter uma boa qualidade no sinal reconstruído, motivo pelo qual devemos ter um controle sobre a quantização. Em outras palavras, se aumento a correção não-linear, não seria bom aumentar também a quantização por sub-bandas porque aumentaria muito o erro de

reconstrução. Também é válida a afirmação para o caso contrário, se se aumenta a quantização por sub-bandas, não é bom aumentar a correção não linear desde um ponto de vista de qualidade no sinal reconstruído. Por esses motivos, $\lambda = \lambda_1 + \lambda_2$.

3.1.4 RLE (*run length encoding*)

Após a quantização, teremos várias sequências de coeficientes que se tornam nulos. No caso de wavelets, o primeiro nível de decomposição faz com que a metade dos coeficientes transformados representem alta frequência, como pode ser observado na Figura 3.3 e, se for uma boa representação, serão de baixa amplitude. Por esse motivo, uma vez feito o reordenamento não linear e a quantização por sub-bandas dos coeficientes de alta-frequência, todos eles serão zero, motivo pelo qual fazer um RLE nessa instância conseguiria um grande nível de compressão. Podemos observar o resultado da quantização na Figura 3.9, em que pode-se ver, como foi explicado, que temos agora uma grande quantidade de coeficientes com valor zero. Por esse motivo, uma

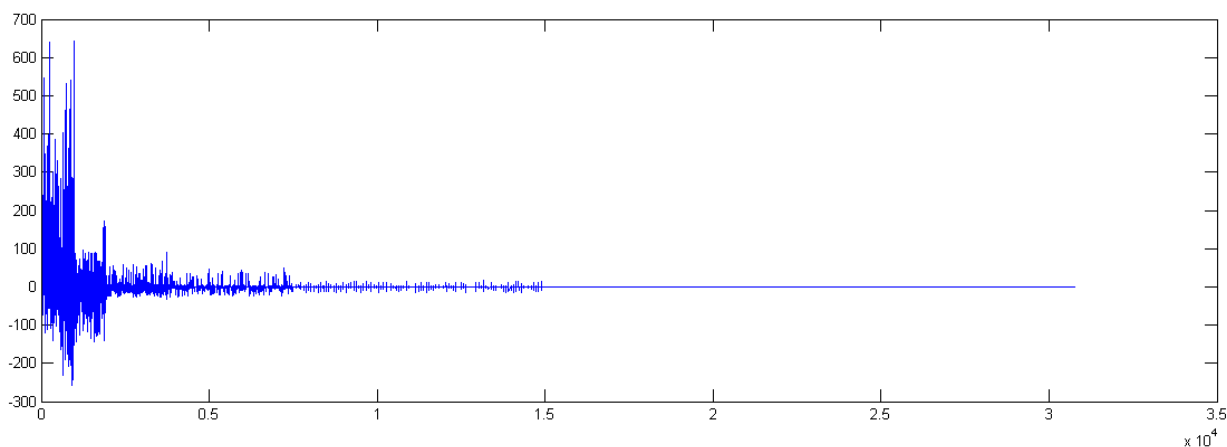


Figura 3.9: Transformada de wavelets db4 do sinal 102 após o processo de quantização.

estratégia para explorar esse fato seria útil. O algoritmo RLE representa uma sequência de números repetidos de maneira mais eficiente. No nosso caso, o número repetido será o zero, que é o número mais comumente utilizado. O RLE empregado é mais facilmente explicado através de um exemplo. Para uma sequência [100 200 340 15 0 0 0 1 0 0 0 0] a saída seria [100 200 340 15 0 3 1 0 5], ou seja que os valores que não são zero ficam inalterados e as sequências de zero são representadas por um zero seguido por um número que representa a quantidade de vezes de ocorrência de zeros.

3.1.5 Codificação aritmética

O que resta por fazer é gerar uma sequência de bits, ou código, que represente a sequência após o RLE. Com esse objetivo, é aplicado um codificador aritmético, que é o codificador de entropia que na teoria mais se aproxima à entropia do sinal, para terminar de remover qualquer tipo de redundância que possa estar presente ainda. O tipo de codificador aritmético utilizado é o

do tipo binário, ou seja que da como resultado um código binário representando a sequência que se quis codificar.

3.1.6 Processo de reconstrução do sinal

Para recuperar o sinal a partir do código comprimido, se realiza o processo inverso ao implementado. Podemos observar esse processo na Figura 3.10.

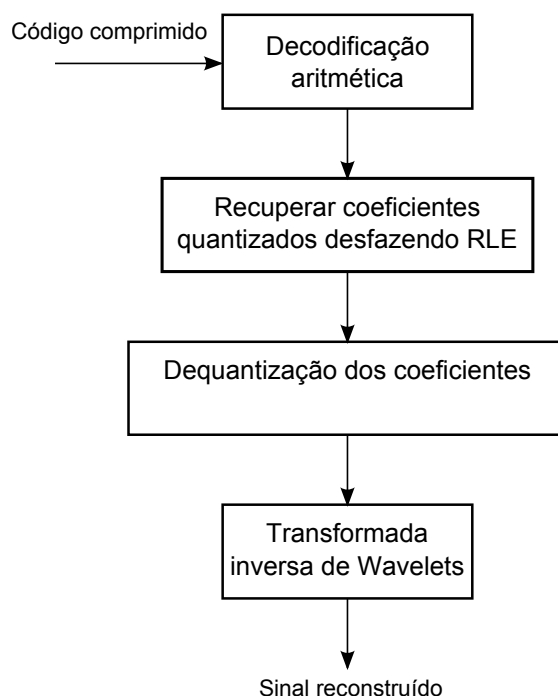


Figura 3.10: Diagrama de blocos do sistema de reconstrução.

Para recuperar o sinal deve-se desfazer a codificação aritmética, recuperar os coeficientes quantizados desfazendo o RLE, fazer as operações correspondentes à dequantização dos coeficientes e finalmente realizar a transformada inversa de wavelets. A equação de dequantização é dada por

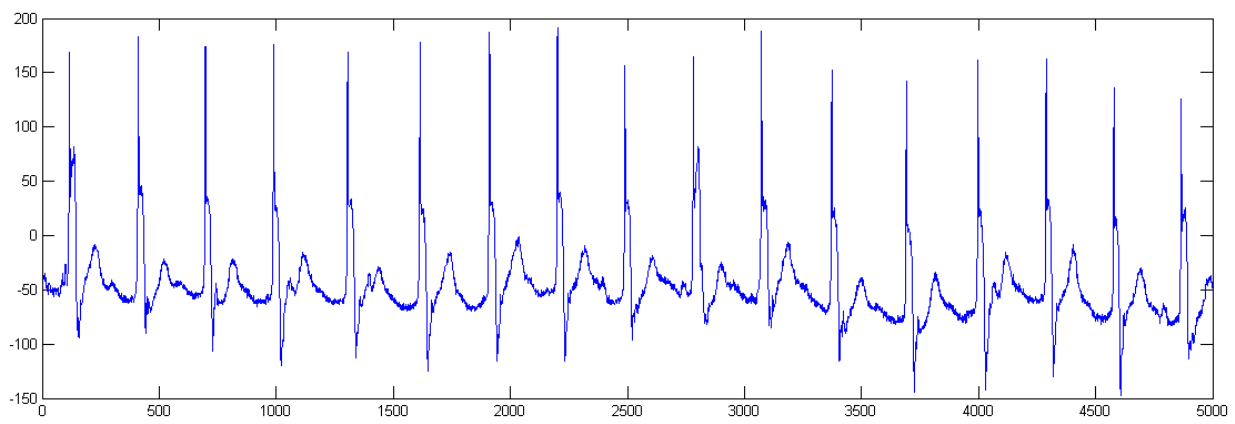
$$C_{reconstruido} = \text{sgn}(C_{rec}) \left[\frac{|C_{rec}| |C_{rec}|_{max}^{\lambda_1}}{\alpha_i} \right]^{\frac{1}{1+\lambda_1}}, \quad (3.12)$$

levando em consideração que expressamos o termo de quantização por sub-bandas da seguinte forma:

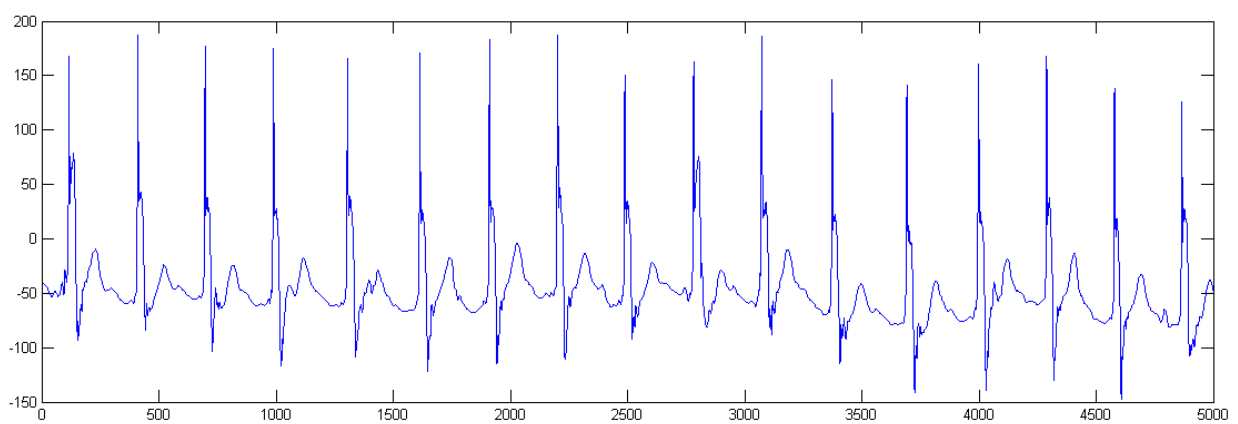
$$\frac{|C_{rec}|_i^{\lambda_2}}{|C_{rec}|_{max}^{\lambda_2}} = \alpha_i, \quad (3.13)$$

em que C_{rec} representa os valores dos coeficientes recuperados após desfazer a codificação aritmética e desfazer o RLE.

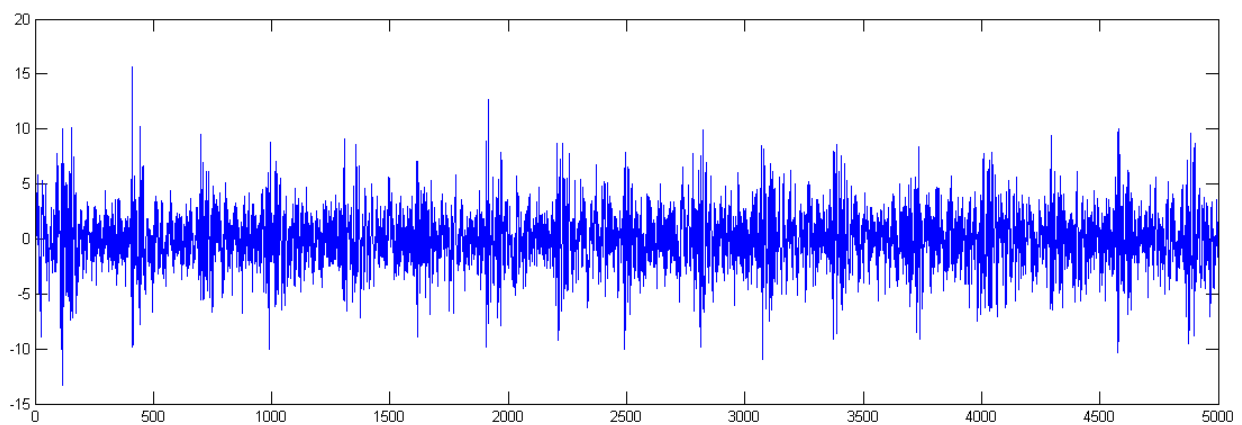
Na Figura 3.11 (a) podemos ver o sinal 102 original, na Figura 3.11 (b) podemos ver o sinal 102 reconstruído a partir do código comprimido através do processo de reconstrução descrito e na Figura 3.11 (c) podemos ver o resíduo entre os dois sinais que é de uma ordem muito menor à amplitude dos sinais. As métricas CR e PRD serão definidas em seções posteriores.



(a)



(b)



(c)

Figura 3.11: (a) Sinal 102 original. (b) Sinal 102 reconstruído a partir do código comprimido usando db4, 5 níveis de decomposição, 32 sub-bandas, $\lambda = 0,82$, $\lambda_1 = 0,6560$, $\lambda_2 = 0,1640$, obtendo uma compressão de CR= 12,80 : 1 e erro de reconstrução PRD= 4,2%.

3.2 PROCEDIMENTOS EXPERIMENTAIS PARA VALIDAÇÃO DO SISTEMA DESENVOLVIDO

3.2.1 Banco de Dados

Para realizar os diferentes testes de desempenho do algoritmo foram utilizados sinais de ECG da base de dados de arritmias do MIT-BIH [5]. A base de dados é composta por 48 ECGs ambulatoriais de dois canais, de meia hora de duração cada. Essa base de dados é a mais utilizada nos trabalhos de compressão de ECG para comparar os diferentes resultados dos algoritmos, pela facilidade no seu acesso. Os sinais de ECG que formam parte do banco de dados foram digitalizados com uma amostragem de 360 amostras por segundo por canal e 11 bits de resolução. O banco de dados possui sinais diversos, sendo que 23 desses sinais foram escolhidos aleatoriamente de um conjunto de 4000 sinais, e os 25 restantes foram escolhidos do mesmo conjunto para incluir arritmias menos comuns mas de importância clínica [5]. Portanto, os resultados obtidos a partir dos testes feitos sobre esse banco podem ser considerados conclusivos, pois esses sinais são muito diversos e representam a maioria dos ECGs que se encontrarão na prática.

3.2.2 Métricas de desempenho

Antes de explicar os testes realizados com o algoritmo utilizando o banco de dados, é necessário entender as métricas que serão utilizadas nos cálculos. Elas são o fator de compressão (CR) e a raiz da diferença média percentual (PRD). O CR é definido como:

$$CR = \frac{\text{tamanho}_{\text{sinal original}}}{\text{tamanho}_{\text{sinal comprimido}}}, \quad (3.14)$$

em que $\text{tamanho}_{\text{sinal original}} = 11 \text{ bits} \times \text{número de amostras}$ (os sinais da base de dados MIT-BIH são amostrados com uma frequência de 360 Hz e uma resolução de 11 bits). O $\text{tamanho}_{\text{sinal comprimido}}$ é calculado pelo comprimento do código após a codificação aritmética [28]. O CR expressado dessa forma dá uma idéia de quantas vezes se conseguiu comprimir o sinal. Por exemplo, se o $CR=10$, significa que teve uma taxa de compressão de 10 : 1, o que significa que se reduziu o tamanho do sinal em 10 vezes. Ou seja, se o sinal original tinha um tamanho de 1 Mb, o sinal comprimido tem um tamanho de 100 kb.

O PRD é calculado por:

$$PRD = \sqrt{\frac{\sum_{n=1}^N (x_{orig}(n) - x_{recon}(n))^2}{\sum_{n=1}^N x_{orig}(n)^2}} \times 100, \quad (3.15)$$

em que $x_{orig}(n)$ é a n -ésima amostra do sinal original e $x_{recon}(n)$ é a n -ésima amostra do sinal recuperado a partir do código comprimido. Usaremos também uma versão modificada do PRD que é independente do valor médio do sinal original que denominaremos PRD1 [28]. Medindo a distorção do sinal recuperado, o PRD1 não tem o erro do nível de base do sinal ECG original. O

PRD1 se define como

$$PRD1 = \sqrt{\frac{\sum_{n=1}^N (x_{orig}(n) - x_{recon}(n))^2}{\sum_{n=1}^N (x_{orig}(n) - \bar{x}_{orig}(n))^2}} \times 100, \quad (3.16)$$

em que $\bar{x}_{orig}(n)$ é o valor médio de $x_{orig}(n)$.

Definiremos também um indicador que denominaremos fator de mérito (FM), que terá um valor mais alto a medida que o CR for mais alto, e o erro de reconstrução for menor (menor PRD1).

$$FM = \frac{CR}{PRD1}. \quad (3.17)$$

É um indicador que foi definido para avaliar o resultado do algoritmo tendo em conta a qualidade do sinal reconstruído além do CR. Se o CR não foi tão alto quanto outro caso, mas teve menor erro de reconstrução, o algoritmo pode ser considerado melhor apesar de ter menor CR, se o FR for mais alto. Como vimos anteriormente, à medida que a compressão aumenta, o erro de reconstrução aumenta também. Por esse motivo, um algoritmo de compressão com bom desempenho tentará comprimir mais mantendo o erro de reconstrução baixo, e, portanto, procurará aumentar o FM.

3.2.3 Testes realizados

O algoritmo empregado utiliza o primeiro canal de cada sinal usado da base de dados MIT-BIH, e realiza o processo descrito na Seção 3.1 utilizando diferentes famílias wavelets para o processo de transformação, testando diferentes valores de λ_1 e λ_2 na quantização para cada família wavelet utilizada. As famílias wavelet utilizadas foram a Daubechies, nas suas variações db1 até db25, a Symlets, nas suas variações sym1 até sym15, coiflets, nas suas variações coif1 até coif5, Ortogonais e Bi-ortogonais em diversas variações e a Meyer discreta. Em todos os casos, foram utilizados 5 níveis de decomposição para a transformação wavelet. Para cada transformação realizada, foram testados diferentes valores de λ . Lembremos que é definido como $\lambda = \lambda_1 + \lambda_2$, que testa valores no intervalo $0,5 \leq \lambda \leq 1,3$ (usaram-se 11 valores para encurtar o tempo de execução, ou seja, 0,5, 0,58, 0,66, 0,74, 0,82, 0,90, 0,98, 1,06, 1,14, 1,22 e 1,30). Esse intervalo foi escolhido empiricamente, pois valores de λ fora desse intervalo não têm um impacto perceptível sobre a compressão e o erro de reconstrução. Além de λ variar entre 0,5 e 1,3, se $\lambda = 1$, por exemplo, então se testaram onze combinações de λ_1 e λ_2 para esse valor, ou seja $\lambda_1 = 0$ e $\lambda_2 = 1$, $\lambda_1 = 0,1$ e $\lambda_2 = 0,9$, e assim até $\lambda_1 = 1$ e $\lambda_2 = 0$.

Do banco de dados de 48 sinais, foram escolhidos 7 sinais para poder calcular λ_1 e λ_2 ótimos para a compressão (melhor FM médio), e os 41 sinais restantes foram utilizados para realizar simulações para avaliar os resultados. Para calcular λ_1 e λ_2 ótimos foi feito o seguinte procedimento:

1. Para cada família wavelet foram testados diferentes combinações de λ_1 e λ_2 fazendo λ variar entre $0,5 \leq \lambda \leq 1,3$, dando como resultado uma matriz 11×11 (11 valores de λ_1

e 11 valores de λ_2). Também como resultado se obtém uma matriz 11×11 com taxas de compressão CR para cada combinação de λ_1 e λ_2 e uma matriz 11×11 com erros de reconstrução PRD para cada combinação de λ_1 e λ_2 . A divisão da matriz CR pela matriz PRD, da como resultado uma matriz 11×11 que representa os FM para diferentes combinações de λ_1 e λ_2 para essa família wavelet.

2. Essa mesma simulação se executou sobre os 7 primeiros sinais do banco de dados (sinais 100, 101, 102, 103, 104, 105 e 106).
3. Se calculou a matriz FM média, somando as 7 matrizes FM e dividindo por 7.
4. Tendo a matriz FM média, localizou-se o valor máximo de FM, que representa o melhor desempenho obtido pelo algoritmo, e guardou-se a posição da matriz. Sabendo a posição da matriz, sabem-se também os valores de λ_1 e λ_2 ótimos devido a que cada ponto da matriz corresponde a uma combinação única λ_1, λ_2 .
5. Até esse passo, foram calculados λ_1 e λ_2 ótimos para uma família de wavelets. O procedimento é realizado novamente para a seguinte família e assim sucessivamente até ter feito a simulação para todas as famílias e todos os valores de λ .

Após o algoritmo acabar, sabemos agora λ_1 e λ_2 ótimos para cada família wavelet. Esse processo foi realizado para agilizar a execução do algoritmo na compressão de outros ECGs. Quando agora se queira comprimir um sinal de ECG que não seja um dos sete primeiros da base de dados, não se testarão todas as combinações de λ_1 e λ_2 . Se utilizarão apenas os valores de λ_1 e λ_2 ótimos, fazendo que o algoritmo se execute de forma mais rápida, sabendo que o resultado obtido pode ser considerado satisfatório. Terminaremos de provar isso nas simulações seguintes.

Nessa etapa temos calculado um valor de λ_1 e λ_2 ótimos para cada família wavelet, e sabemos o FM médio correspondente a essa combinação de λ_1 e λ_2 com essa família wavelet. Para testar o algoritmo e ver que o desempenho seja de acordo ao esperado, ou seja que o FM se mantenha para os diferentes sinais, executamos o algoritmo para o resto do banco de dados, sobre os 41 sinais restantes não utilizados no cálculo de λ_1 e λ_2 ótimos. Obtemos assim os resultados de CR e PRD para cada sinal, usando λ_1 e λ_2 calculados, para cada família wavelet. Ou seja, para a família db1 temos 41 resultados de CR e 41 resultados de PRD, para a família db2 temos 41 resultados de CR e 41 resultados de PRD, e assim para todas as famílias wavelets.

3.3 MÉTODOS PARA A ANÁLISE DOS RESULTADOS

Com a informação que possuímos, estamos em posição de apresentar os resultados do algoritmo, e avaliar seu desempenho. Os resultados serão mostrados com um intervalo da forma $\mu \pm \varepsilon$, em que ε é chamado de intervalo de confiabilidade, ou erro, e pode variar dependendo da confiabilidade que for dada ao resultado. Para poder expressar os resultados dessa forma, devemos ter em conta o seguinte. Tomemos como exemplo os resultados das taxas de compressão e, assumindo que eles têm uma distribuição normal, a distribuição de probabilidade deles será da forma mostrada na Figura 3.12. Podemos calcular a média μ_c deles como a soma deles dividido pelo

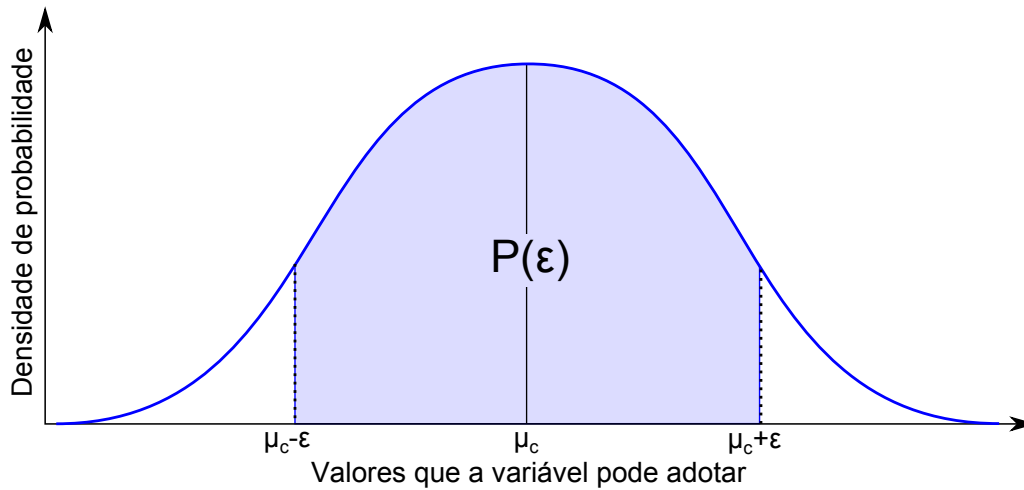


Figura 3.12: Função de distribuição de probabilidade normal.

número de sinais. A variância é definida como

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (CR_i - \mu_c)^2, \quad (3.18)$$

em que CR_i é a taxa de compressão de cada sinal. Observando a Figura 3.12, vemos que $P(\varepsilon)$ define a probabilidade de um CR cair dentro da região sombreada. Podemos calcular $P(\varepsilon)$ como

$$P(\varepsilon) = 2 \int_{\mu_c}^{\mu_c + \varepsilon} f dp(x) dx, \quad (3.19)$$

em que $f dp(x)$ é a função densidade de probabilidade. Se ela tem uma distribuição normal então pode ser escrita como

$$P(\varepsilon) = 2 \int_{\mu_c}^{\mu_c + \varepsilon} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x - \mu_c)^2}{2\sigma^2}} dx, \quad (3.20)$$

e, se definimos uma variável y como

$$y = \frac{x - \mu_c}{\sqrt{2}\sigma}, \quad (3.21)$$

podemos re-escrever a probabilidade como

$$P(\varepsilon) = 2 \int_0^{\frac{\varepsilon}{\sqrt{2}\sigma}} \frac{1}{\sigma \sqrt{2\pi}} e^{-y^2} dy \sqrt{2}\sigma, \quad (3.22)$$

que simplificando é

$$P(\varepsilon) = \frac{2}{\sqrt{\pi}} \int_0^{\frac{\varepsilon}{\sqrt{2}\sigma}} e^{-y^2} dy. \quad (3.23)$$

Por definição de teoria da probabilidade também sabemos que a função erro, que é definida como a probabilidade de uma variável aleatória com distribuição normal de média 0 e variância 1/2 cair no intervalo $[-x, x]$ [68], pode ser expressada como

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (3.24)$$

Podemos ver então que a $P(\varepsilon)$ pode ser definida como

$$P(\varepsilon) = \text{erf} \left(\frac{\varepsilon}{\sqrt{2}\sigma} \right). \quad (3.25)$$

Finalmente podemos calcular ε como

$$\varepsilon = \sqrt{2} \text{erf}^{-1}(P(\varepsilon)) \sigma \quad (3.26)$$

e podemos chamar $P(\varepsilon)$ de confiabilidade do resultado.

Fazer uma comparação estatística dos resultados permitirá tirar conclusões que permitem aceitar uma família como sendo melhor que outra de forma mais conclusiva que apenas mostrando as médias e o intervalo de confiabilidade. Podemos pensar no exemplo dos 41 resultados de CR para uma família wavelet serem similares, tendo apenas um resultado muito acima da média. Ao comparar os resultados dessa família com outra, por causa desse resultado isolado muito acima da média, poderia-se concluir que a primeira família é melhor que a segunda utilizando só as médias para comparar, e isso não necessariamente é verdade (o resultado da primeira família foi maior por causa do resultado isolado muito acima da média). Para evitar isso, se realiza uma comparação estatística onde se dividiram as famílias de wavelets em 5 grupos. Todas as Daubechies em um grupo, todas as Symlets em outro grupo, todas as Coiflets em outro grupo, todas as Bi-ortogonais em outro grupo, todas as Bi-ortogonais inversas em outro grupo e a Meyer Discreta em outro grupo, para decidir assim qual a melhor wavelet dentro de cada grupo, desde um ponto de vista estatístico. Uma vez feito isso, se comparará a melhor wavelet de cada grupo uma com outra.

Para poder realizar as comparações estatísticas usaremos o teste de U de Mann-Whitney, também conhecido como o teste de *Wilcoxon rank sum* [69] (o teste de U de Mann-Whitney pode usar conjuntos de diferente tamanho, ao contrario de *Wilcoxon rank sum* que utiliza conjuntos do mesmo tamanho). Esse teste serve para comparar dois conjuntos de dados através de uma hipótese, que pode ser aceita ou rejeitada com uma probabilidade de valor p (normalmente um valor de $p < 0,05$ para a hipótese alternativa permite aceitar a hipótese nula). As hipóteses possíveis são que a mediana do conjunto A seja maior que a do B , que a mediana de B seja maior que a do A , ou que as medianas sejam iguais. No nosso caso, cada conjunto será o resultado de compressão de uma família wavelet para os 41 sinais do banco de dados, e se fará o teste de *Wilcoxon rank sum* tomando como conjunto os valores de FM para uma família wavelet, para aceitar uma família como sendo melhor que outra. Se aceitará uma família como sendo melhor que outra se o teste realizado devolver um valor de $p < 0,05$ (95% de confiança). Caso não seja possível obter resultados com $p < 0,05$, se avaliaram outras alternativas.

Por último, os resultados do algoritmo proposto serão comparados com a literatura. Na maioria dos trabalhos da literatura se utilizam os resultados do algoritmo de compressão para o sinal 117. Por tanto, será apresentado um quadro comparando resultados para esse sinal particular. Idealmente, seria mais útil comparar os resultados na média com outros algoritmos. Porém, na literatura esse tipo de resultado raramente se encontra, e, além disso, não forma parte do escopo

do presente trabalho implementar os diferentes algoritmos da literatura para poder calcular os resultados na média.

4 RESULTADOS E DISCUSSÕES

Fazendo referência à Seção 3.2.3, podemos ver os diferentes testes que foram realizados. Para o cálculo de λ_1 e λ_2 ótimos, devemos ter em conta que, no algoritmo implementado, se utilizam duas variáveis para o cálculo deles. Uma variável é λ que varia entre $0,5 \leq \lambda \leq 1,3$, que foi escolhida de forma empírica, levando em consideração que o algoritmo de quantização não apresentava mudanças perceptíveis no desempenho fora desse intervalo. A outra variável é K , que varia entre $0 \leq K \leq 1$ (temos também K^* que se define como $K^* = 1 - K$). K e K^* são variáveis que dão peso à quantização por sub-bandas e à redistribuição não-linear, ou seja, dão peso a λ_1 e λ_2 . λ_1 é definido como $\lambda_1 = \lambda K$ e λ_2 é definido como $\lambda_2 = \lambda K^*$. Na Tabela 4.1 podemos observar os valores de λ e K ótimos, que correspondem com λ_1 e λ_2 ótimos, calculados a partir dos 7 primeiros sinais do bancos de dados (sinais 100, 101, 102, 103, 104, 105 e 106) como explicado em 3.2.3. Os resultados mostrados na Tabela 4.1 podem ser observados de forma gráfica na Figura 4.1.

Tabela 4.1: Tabela com λ e K ótimos para cada família wavelet calculados a partir das médias de 7 sinais do banco de dados, com seus respectivos FM médio.

Família Wavelet	λ	K	FM
Daubechies 1	0,50	0,80	1,38
Daubechies 2	0,66	0,90	1,86
Daubechies 3	0,66	0,90	2,05
Daubechies 4	0,74	0,80	2,11
Daubechies 5	0,66	0,90	2,14
Daubechies 6	0,74	0,80	2,12
Daubechies 7	0,66	0,90	2,10
Daubechies 8	0,66	0,90	2,07
Daubechies 9	0,66	0,90	2,04
Daubechies 10	0,58	1,00	2,00
Daubechies 11	0,66	0,90	1,98
Daubechies 12	0,58	1,00	1,96
Daubechies 13	0,58	1,00	1,93
Daubechies 14	0,58	1,00	1,91
Daubechies 15	0,58	0,90	1,89
Daubechies 16	0,58	1,00	1,87
Daubechies 17	0,58	0,90	1,84
Daubechies 18	0,50	1,00	1,83
Daubechies 19	0,58	0,90	1,81
Daubechies 20	0,50	0,90	1,79

Daubechies 21	0,50	0,90	1,78
Daubechies 22	0,50	0,90	1,77
Daubechies 23	0,50	0,90	1,76
Daubechies 24	0,50	0,80	1,74
Daubechies 25	0,50	0,90	1,74
Symlets 1	0,50	0,80	1,38
Symlets 2	0,66	0,90	1,86
Symlets 3	0,66	0,90	2,05
Symlets 4	0,74	0,80	2,17
Symlets 5	0,66	0,90	2,20
Symlets 6	0,66	0,90	2,22
Symlets 7	0,74	0,80	2,22
Symlets 8	0,66	0,90	2,21
Symlets 9	0,66	0,90	2,19
Symlets 10	0,66	0,90	2,20
Symlets 11	0,66	0,90	2,17
Symlets 12	0,66	0,80	2,18
Symlets 13	0,66	0,90	2,14
Symlets 14	0,66	0,90	2,16
Symlets 15	0,66	0,90	2,13
Coiflets 1	0,66	0,90	1,88
Coiflets 2	0,74	0,80	2,17
Coiflets 3	0,66	0,90	2,23
Coiflets 4	0,66	0,90	2,20
Coiflets 5	0,66	0,90	2,17
Bi-ortogonal 1.1	0,50	0,80	1,38
Bi-ortogonal 1.3	0,50	0,70	1,33
Bi-ortogonal 1.5	0,50	0,80	1,30
Bi-ortogonal 2.2	0,66	1,00	2,09
Bi-ortogonal 2.4	0,58	1,00	2,08
Bi-ortogonal 2.6	0,66	1,00	2,06
Bi-ortogonal 2.8	0,66	1,00	2,05
Bi-ortogonal 3.1	0,82	1,00	1,69
Bi-ortogonal 3.3	0,74	1,00	2,07
Bi-ortogonal 3.5	0,74	1,00	2,16
Bi-ortogonal 3.7	0,74	1,00	2,18
Bi-ortogonal 3.9	0,74	1,00	2,16
Bi-ortogonal 4.4	0,74	0,80	2,32
Bi-ortogonal 5.5	0,74	0,70	2,26

Bi-ortogonal 6.8	0,66	0,90	2,30
Bi-ortogonal Inv 1.1	0,50	0,80	1,38
Bi-ortogonal Inv 1.3	0,66	0,90	2,08
Bi-ortogonal Inv 1.5	0,74	0,80	2,21
Bi-ortogonal Inv 2.2	0,58	0,70	1,43
Bi-ortogonal Inv 2.4	0,66	0,80	1,85
Bi-ortogonal Inv 2.6	0,74	0,60	1,97
Bi-ortogonal Inv 2.8	0,66	0,80	2,02
Bi-ortogonal Inv 3.1	0,50	0,00	0,85
Bi-ortogonal Inv 3.3	0,74	0,40	1,16
Bi-ortogonal Inv 3.5	0,74	0,50	1,45
Bi-ortogonal Inv 3.7	0,74	0,60	1,58
Bi-ortogonal Inv 3.9	0,74	0,50	1,63
Bi-ortogonal Inv 4.4	0,66	0,90	1,95
Bi-ortogonal Inv 5.5	0,66	1,00	2,02
Bi-ortogonal Inv 6.8	0,66	0,90	2,13
Meyer Discreta	0,58	1,00	1,96

Nas curvas da Figura 4.2, podemos ver o comportamento de cada grupo de famílias com relação ao seu FM. Temos uma curva para todas as famílias Daubechies utilizadas (db1-db25), em que podemos ver como varia o FM para cada uma das famílias. Podemos ver que o FM aumenta até chegar a um máximo para a família db5. Temos uma curva para todas as famílias Symlets utilizadas (sym1-sym15), em que podemos ver que o FM aumenta até chegar a um máximo para a família sym7. Temos uma curva para todas as famílias Coiflets utilizadas (coif1-coif5), em que podemos ver que o FM aumenta até chegar a um máximo para a família coif3. Temos uma curva para todas as famílias Bi-ortogonais utilizadas, em que podemos ver que o comportamento do FM não está claramente definido e tem um máximo para a família bior6.8, muito próximo ao valor da família bior 4.4. Temos uma curva para todas as famílias Bi-ortogonais Inversas utilizadas, em que podemos ver que o comportamento do FM não está claramente definido e tem um máximo para a família rbio1.5. Por último temos um ponto para a família Meyer Discreta dmey.

Para cada família wavelet com seus respectivos λ_1 e λ_2 ótimos calculados, foi executado o algoritmo para os demais sinais do banco de dados, obtendo-se 41 resultados de compressão (41 sinais restantes no banco de dados). Para esses resultados, foram calculadas as médias e o intervalo de confiabilidade (para uma confiabilidade de 95%) segundo o explicado na Seção 3.3. Esses resultados podem ser observados na Tabela 4.2.

Tabela 4.2: Tabela com λ e K ótimos calculados a partir de 7 sinais, aplicados para cada família wavelet para os 41 sinais restantes do banco de dados, com os valores médios de PRD, PRD1, CR e FM, e seus respectivos intervalos de confiabilidade calculados com 95% de confiabilidade.

Família Wavelet	λ	K	μ_{PRD}	ε_{PRD}	μ_{PRD1}	ε_{PRD1}	μ_{CR}	ε_{CR}	μ_{FM}	ε_{FM}
Daubechies 1	0,50	0,80	2,56	2,18	3,68	2,66	5,30	2,10	1,56	0,82
Daubechies 2	0,66	0,90	3,28	2,80	4,71	3,34	9,37	3,77	2,17	1,29
Daubechies 3	0,66	0,90	3,15	2,77	4,51	3,30	10,17	4,08	2,48	1,52
Daubechies 4	0,74	0,80	3,29	2,85	4,72	3,42	11,08	4,54	2,57	1,55
Daubechies 5	0,66	0,90	3,02	2,73	4,32	3,27	10,24	4,10	2,62	1,63
Daubechies 6	0,74	0,80	3,24	2,85	4,63	3,40	10,94	4,33	2,59	1,58
Daubechies 7	0,66	0,90	2,98	2,72	4,26	3,23	10,03	3,94	2,60	1,60
Daubechies 8	0,66	0,90	2,95	2,70	4,22	3,24	9,85	3,91	2,58	1,60
Daubechies 9	0,66	0,90	2,95	2,69	4,22	3,21	9,73	3,78	2,55	1,57
Daubechies 10	0,58	1,00	2,67	2,50	3,81	2,93	8,71	3,12	2,55	1,60
Daubechies 11	0,66	0,90	2,92	2,69	4,18	3,21	9,43	3,65	2,50	1,52
Daubechies 12	0,58	1,00	2,65	2,48	3,77	2,95	8,47	3,10	2,50	1,57
Daubechies 13	0,58	1,00	2,65	2,48	3,77	2,93	8,39	2,98	2,48	1,55
Daubechies 14	0,58	1,00	2,64	2,48	3,75	2,92	8,27	2,93	2,46	1,53
Daubechies 15	0,58	0,90	2,54	2,42	3,61	2,87	7,92	2,76	2,46	1,53
Daubechies 16	0,58	1,00	2,61	2,49	3,72	2,97	8,08	2,97	2,43	1,52
Daubechies 17	0,58	0,90	2,53	2,39	3,60	2,82	7,75	2,71	2,41	1,50
Daubechies 18	0,50	1,00	2,21	2,11	3,14	2,48	6,87	2,12	2,47	1,60
Daubechies 19	0,58	0,90	2,51	2,42	3,57	2,86	7,60	2,69	2,38	1,48
Daubechies 20	0,50	0,90	2,12	2,02	3,01	2,38	6,56	2,07	2,46	1,61
Daubechies 21	0,50	0,90	2,12	2,01	3,01	2,37	6,52	2,03	2,45	1,60
Daubechies 22	0,50	0,90	2,12	2,02	3,00	2,37	6,47	1,95	2,43	1,58
Daubechies 23	0,50	0,90	2,11	2,02	3,00	2,39	6,41	1,99	2,42	1,58
Daubechies 24	0,50	0,80	2,04	1,94	2,89	2,28	6,21	1,92	2,43	1,61
Daubechies 25	0,50	0,90	2,10	2,00	2,99	2,36	6,33	1,96	2,39	1,56
Symlets 1	0,50	0,80	2,56	2,18	3,68	2,66	5,30	2,10	1,56	0,82
Symlets 2	0,66	0,90	3,28	2,80	4,71	3,34	9,37	3,77	2,17	1,29
Symlets 3	0,66	0,90	3,15	2,77	4,51	3,30	10,17	4,08	2,48	1,52
Symlets 4	0,74	0,80	3,31	2,86	4,74	3,41	11,36	4,55	2,62	1,58
Symlets 5	0,66	0,90	3,04	2,72	4,35	3,24	10,59	4,11	2,69	1,69
Symlets 6	0,66	0,90	3,02	2,72	4,33	3,23	10,65	4,23	2,72	1,70
Symlets 7	0,74	0,80	3,23	2,83	4,62	3,40	11,28	4,54	2,68	1,66
Symlets 8	0,66	0,90	3,00	2,73	4,29	3,24	10,56	4,09	2,73	1,70
Symlets 9	0,66	0,90	2,99	2,72	4,27	3,21	10,50	4,06	2,72	1,71
Symlets 10	0,66	0,90	2,98	2,72	4,26	3,23	10,46	4,06	2,72	1,68
Symlets 11	0,66	0,90	2,97	2,70	4,24	3,18	10,29	3,98	2,69	1,69
Symlets 12	0,66	0,80	2,87	2,64	4,10	3,13	9,97	3,89	2,69	1,67
Symlets 13	0,66	0,90	2,99	2,72	4,26	3,20	10,33	3,95	2,68	1,67

Symlets 14	0,66	0,90	2,97	2,71	4,24	3,22	10,25	4,03	2,68	1,66
Symlets 15	0,66	0,90	2,96	2,71	4,23	3,18	10,11	3,90	2,64	1,64
Coiflets 1	0,66	0,90	3,27	2,81	4,69	3,33	9,46	3,82	2,20	1,30
Coiflets 2	0,74	0,80	3,30	2,87	4,74	3,42	11,39	4,59	2,64	1,60
Coiflets 3	0,66	0,90	3,01	2,71	4,31	3,25	10,58	4,21	2,71	1,69
Coiflets 4	0,66	0,90	2,98	2,70	4,27	3,23	10,46	4,10	2,71	1,70
Coiflets 5	0,66	0,90	2,98	2,70	4,26	3,22	10,37	4,03	2,69	1,68
Bi-ortogonal 1.1	0,50	0,80	2,56	2,18	3,68	2,66	5,30	2,10	1,56	0,82
Bi-ortogonal 1.3	0,50	0,70	2,50	2,11	3,59	2,55	5,02	1,84	1,52	0,80
Bi-ortogonal 1.5	0,50	0,80	2,62	2,20	3,76	2,65	5,13	1,84	1,48	0,77
Bi-ortogonal 2.2	0,66	1,00	3,38	2,90	4,85	3,44	11,12	4,32	2,52	1,58
Bi-ortogonal 2.4	0,58	1,00	2,91	2,60	4,15	3,02	9,42	3,66	2,51	1,60
Bi-ortogonal 2.6	0,66	1,00	3,24	2,81	4,63	3,31	10,59	4,32	2,51	1,56
Bi-ortogonal 2.8	0,66	1,00	3,22	2,79	4,62	3,30	10,45	4,29	2,48	1,55
Bi-ortogonal 3.1	0,82	1,00	5,52	4,54	7,96	5,68	14,65	4,50	2,04	1,29
Bi-ortogonal 3.3	0,74	1,00	3,87	3,25	5,57	3,98	12,67	4,76	2,52	1,69
Bi-ortogonal 3.5	0,74	1,00	3,66	3,09	5,24	3,69	12,60	4,89	2,65	1,76
Bi-ortogonal 3.7	0,74	1,00	3,57	3,02	5,12	3,61	12,48	4,80	2,69	1,78
Bi-ortogonal 3.9	0,74	1,00	3,54	3,01	5,07	3,57	12,33	4,74	2,68	1,75
Bi-ortogonal 4.4	0,74	0,80	3,32	2,87	4,77	3,43	12,15	4,77	2,80	1,73
Bi-ortogonal 5.5	0,74	0,70	3,31	2,89	4,75	3,50	11,82	4,79	2,73	1,68
Bi-ortogonal 6.8	0,66	0,90	3,00	2,71	4,29	3,22	10,93	4,29	2,82	1,77
Bi-ortogonal Inv 1.1	0,50	0,80	2,56	2,18	3,68	2,66	5,30	2,10	1,56	0,82
Bi-ortogonal Inv 1.3	0,66	0,90	3,19	2,80	4,58	3,40	10,47	4,34	2,51	1,55
Bi-ortogonal Inv 1.5	0,74	0,80	3,37	2,93	4,84	3,54	11,80	4,72	2,68	1,65
Bi-ortogonal Inv 2.2	0,58	0,70	2,96	2,56	4,27	3,14	6,44	2,57	1,66	1,00
Bi-ortogonal Inv 2.4	0,66	0,80	3,22	2,81	4,64	3,42	9,33	3,86	2,21	1,37
Bi-ortogonal Inv 2.6	0,74	0,60	3,25	2,84	4,68	3,46	10,05	4,23	2,36	1,44
Bi-ortogonal Inv 2.8	0,66	0,80	3,11	2,78	4,48	3,40	9,79	4,04	2,42	1,52
Bi-ortogonal Inv 3.1	0,50	0,00	1,64	1,56	2,39	2,12	2,19	0,53	1,05	0,64
Bi-ortogonal Inv 3.3	0,74	0,40	3,85	3,20	5,57	4,02	6,67	2,94	1,31	0,79
Bi-ortogonal Inv 3.5	0,74	0,50	3,66	3,15	5,29	3,92	8,18	3,57	1,70	1,05
Bi-ortogonal Inv 3.7	0,74	0,60	3,67	3,13	5,30	3,92	8,96	3,81	1,86	1,15
Bi-ortogonal Inv 3.9	0,74	0,50	3,48	3,05	5,02	3,80	8,71	3,72	1,91	1,18
Bi-ortogonal Inv 4.4	0,66	0,90	3,15	2,77	4,51	3,31	9,68	3,84	2,36	1,45
Bi-ortogonal Inv 5.5	0,66	1,00	3,14	2,79	4,49	3,28	10,15	4,00	2,49	1,54
Bi-ortogonal Inv 6.8	0,66	0,90	3,02	2,73	4,33	3,27	10,18	4,08	2,60	1,62
Meyer Discreta	0,58	1,00	2,67	2,49	3,80	2,94	8,51	3,04	2,50	1,56

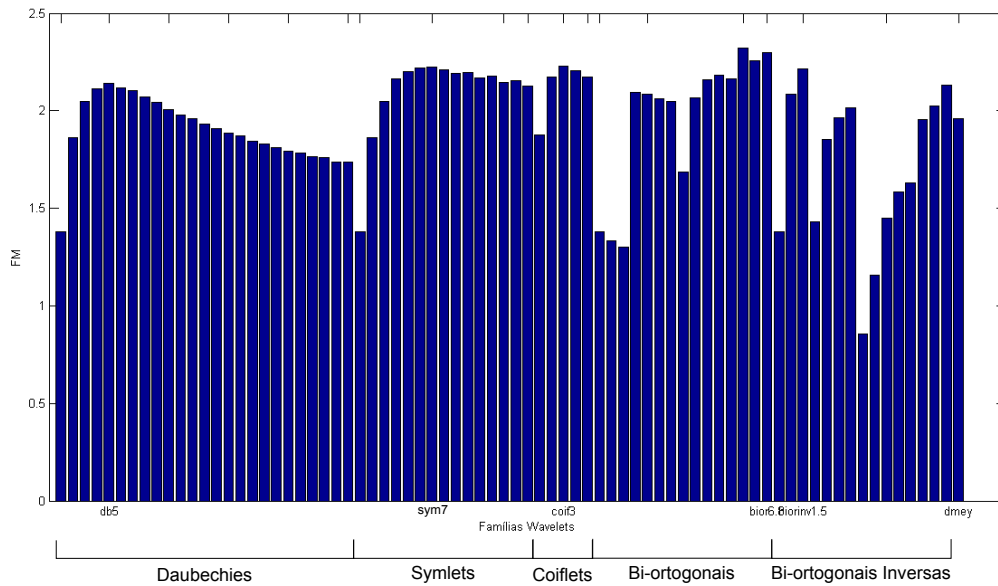


Figura 4.1: Gráfico de barras mostrando o desempenho de cada família de wavelets respeito ao seu FM .

Para poder tirar conclusões sobre qual família é a melhor, ou seja, qual família tem o valor de FM mais elevado, utilizamos a média e o erro ε . Entretanto, para uma confiabilidade grande (95%), a magnitude do erro é de uma ordem de grandeza comparável à média. Por essa razão, outro tipo de teste é necessário. Aplicaremos aqui o teste de *Wilcoxon rank sum* segundo o explicado na Seção 3.3 para ver quais famílias são as melhores, comparando seus FM . Primeiro é comparado cada grupo de famílias entre si (Daubechies com Daubechies, Symlets com Symlets, Coiflets com Coiflets, etc.) para ver qual família é a melhor dentro de um grupo. Posteriormente, as famílias vencedoras no teste anterior são comparadas entre si.

Na Figura 4.3, podemos ver os resultados do teste de *Wilcoxon rank sum* para cada grupo de famílias wavelets, ou seja, é calculada a probabilidade de uma família ser melhor que outra (ter FM maior). Em teoria, o teste de *Wilcoxon rank sum* aceita ou rejeita uma hipótese se o valor de probabilidade da hipótese alternativa é menor que 0,05 (ou seja, 95% de confiança). Por exemplo, comparando db1 com db2 (membro 1 com membro 2), se a hipótese nula é de que db2 é maior do que db1, a hipótese alternativa seria que db1 é maior do que db2, e se o resultado do teste for $p < 0,05$, significa que a hipótese alternativa tem baixa probabilidade de ocorrência. Portanto, posso aceitar a hipótese nula, que é a de que db2 é maior do que db1 (membro 2 maior que membro 1) e, portanto, melhor, porque estamos comparando FM .

O resultado desse teste é mostrado na Figura 4.3, pressupondo que os membros 1 no teste são o eixo das ordenadas, os membros 2 são o eixo abscissas e os valor de p (que o teste dá como resultado) são mostrados com um código de cores, em que vermelho escuro é $p = 1$ e azul escuro é $p = 0$. Os valores na diagonal são ignorados porque é uma comparação de uma família com ela mesma (db1 com db1, db2 com db2, etc.). Como pode se observar no gráfico dos testes de *Wilcoxon rank sum* (Figura 4.3 - (a, c, e, g, i)), os valores de p não são conclusivos para dizer se uma família é melhor que outra, porque o valor p não é sempre menor que 0,05. Isso é devido

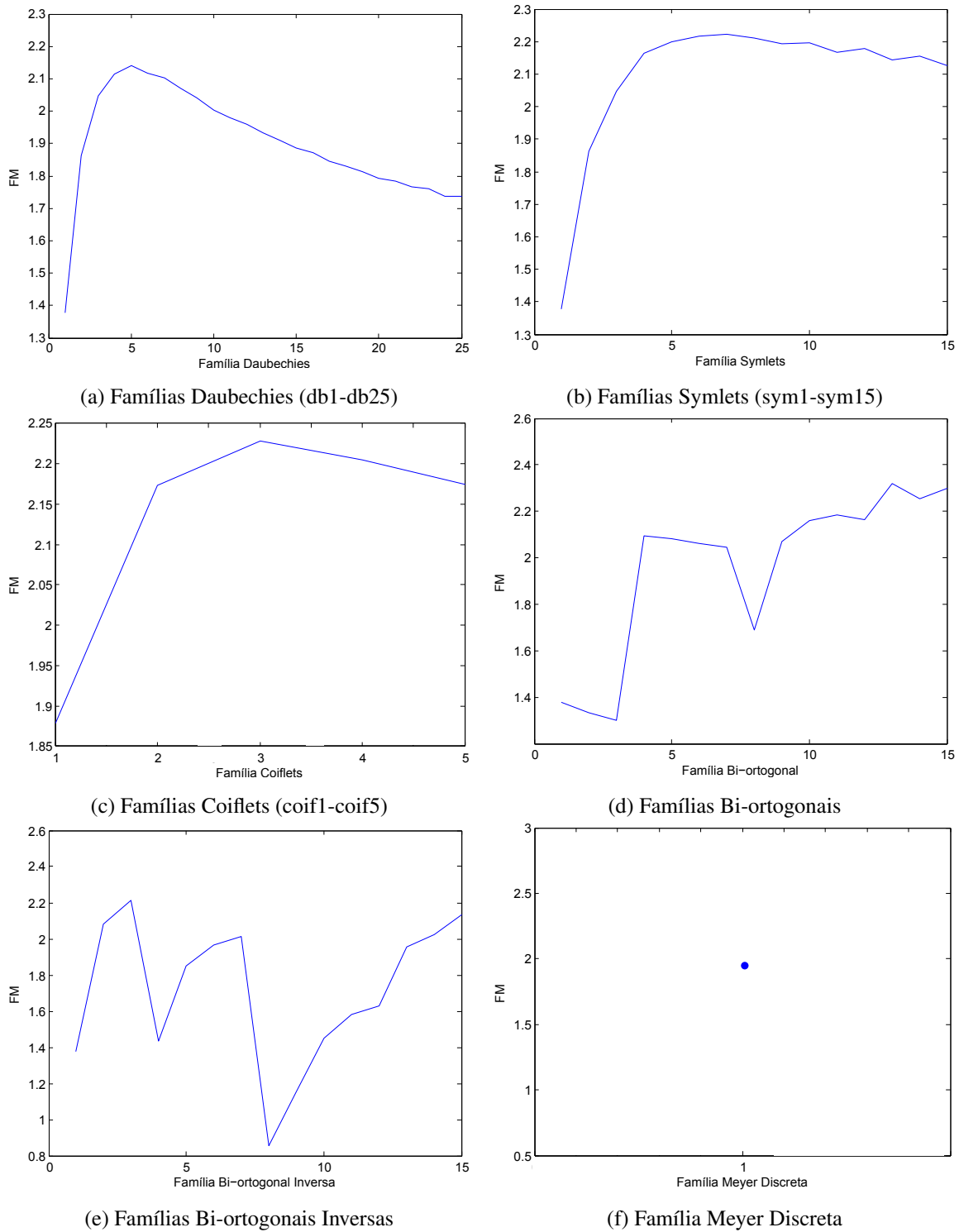


Figura 4.2: Curva do FM médio (para os 7 primeiros sinais do banco de dados) obtido para cada família de wavelets utilizando os λ_1 e λ_2 ótimos de cada uma delas.

à similaridade de performance de uma família com outra dentro de um grupo, como pode ser observado na Figura 4.2, em que várias delas apresentam FM parecidos; por exemplo, dentro do grupo Daubechies, as famílias db5 até a db25, ou no grupo Symlets, as famílias sym5 até a sym15. Logo, relaxamos o critério de comparação, tomando $p < 0,5$, pois mesmo assim podemos dizer

que o membro 2 no teste tem mais probabilidade de ser maior que o membro 1, ou seja, ter um FM maior. Os resultados desse teste são mostrados nos gráficos das Figuras 4.3 - (b, d, f, h e j) onde o branco representa $p < 0.5$ e o preto representa os valores de p restantes. Podemos ver que para cada caso existe uma família na vertical (uma coluna) que é melhor que todas as outras famílias ($p < 0,5$), que foi sinalizada por uma linha vermelha. No caso das famílias Daubechies, a família com melhor desempenho no FM para os 41 sinais, utilizando o teste de *Wilcoxon rank sum*, é a família db5. No caso das famílias Symlets é a sym8. No caso das Coiflets é a coif3. No caso das Bi-ortogonais é a bior6.8 (família bi-ortogonal na coluna 15). No caso das Bi-ortogonais Inversas é a biorinv1.5 (família bi-ortogonal inversa na coluna 3). A Meyer Discreta não foi comparada, pois só existe uma variação dela.

Com os resultados desse teste, resta comparar as melhores famílias wavelets entre si, e avaliar, através do teste de *Wilcoxon rank sum*, se alguma delas possui um desempenho melhor que as outras. Podemos observar através desse teste, na Figura 4.4, que a família com melhor desempenho é a Bi-ortogonal na variação 6.8.

Um dado a se destacar é o fato de que, utilizando o teste de *Wilcoxon rank sum* sobre os resultados obtidos para a compressão dos 41 sinais restantes, obteve-se que as famílias wavelets com melhor desempenho foram a db5, a sym8, a coif3, a bior6.8 e a biorinv1.5. Lembrando os testes realizados no começo sobre os 7 sinais utilizados para calcular λ_1 e λ_2 ótimos, tínhamos concluído através da Figura 4.2 que as famílias com o melhor desempenho eram db5, sym7, coif3, bior6.8, biorinv1.5. Podemos ver que as famílias wavelets com melhor desempenho obtidas com *Wilcoxon rank sum* para 41 sinais, são as mesmas que as obtidas no cálculo de λ_1 e λ_2 ótimos para 7 sinais, diferentes desses 41 sinais usados. A única exceção é na família Symlet. No teste inicial a melhor foi sym7 e no teste *Wilcoxon rank sum* foi a sym8, porém, elas tem um desempenho muito próximo e não consideramos isso como uma discrepância nos resultados. A similaridade entre os resultados demonstra robustez no algoritmo implementado, tendo em vista que, uma vez calculados os parâmetros λ_1 e λ_2 ótimos, podem-se esperar resultados semelhantes para qualquer sinal de ECG que se queira comprimir.

Por último, compararemos os resultados do algoritmo desenvolvido com os resultados da literatura. Em particular, serão mostrados os resultados de compressão para o sinal 117, pois é um dos mais utilizados na literatura.

Na Figura 4.5 (a), podemos observar o sinal 117 original, na Figura 4.5 (b) podemos observar o sinal 117 reconstruído utilizando a família Bi-ortogonal 6.8 para transformar e $\lambda = 0,66$ e $K = 0,90$ (segundo a Tabela 4.1) para o processo de quantização. Na Figura 4.5 (c), podemos observar o resíduo entre o sinal 117 original e o sinal 117 reconstruído. Ambos sinais (o sinal 117 original e o reconstruído) são quase idênticos, com a exceção de que no sinal reconstruído parecem existir menos componentes de alta-frequência, que é um resultado lógico devido ao fato de que vários coeficientes de alta frequência foram zerados durante o processo de compressão.

Na Figura 4.6, podemos ver uma curva do PRD1 em função do CR usando a família de wavelets Bi-ortogonal 6.8 para comprimir o sinal 117. Cada linha vermelha mostra um valor

fixo de λ enquanto varia K e cada linha verde mostra um valor fixo de K enquanto varia λ . Conceitualmente, o ponto de funcionamento da compressão para o sinal 117 com a família Bi-ortogonal 6.8 pode estar em qualquer ponto dentro da região delimitada pelas linhas verdes e vermelhas, dependendo dos fatores λ e K que se utilizem. O ponto onde obtemos o FM ótimo é no ponto $CR = 11,40$ e $PRD1 = 5,13\%$ ($PRD = 1,38\%$), que significa $FM = 2,22$. Para ver as curvas de $PRD1$ em função do CR usando a família Bi-ortogonal 6.8 para todos os sinais do banco de dados, vide Anexo I.

Na Tabela 4.3 se mostra a performance do algoritmo para todos os sinais do banco de dados, incluindo os 7 usados para achar λ_1 e λ_2 ótimos em azul, e os 41 sinais restantes, utilizando a família biortogonal 6.8.

Tabela 4.3: Resultados de compressão para todo o banco de dados MIT-BIH, utilizando λ e K ótimos para a família Bi-ortogonal 6.8.

SINAL	λ	K	PRD	PRD1	CR	FM (CR/PRD1)
100	0,66	0,90	2,78	5,89	11,81	2,01
101	0,66	0,90	3,30	5,29	10,20	1,93
102	0,66	0,90	3,49	6,01	11,37	1,89
103	0,66	0,90	2,74	3,34	11,06	3,31
104	0,66	0,90	3,54	4,65	9,45	2,03
105	0,66	0,90	3,45	4,19	10,93	2,61
106	0,66	0,90	3,86	4,45	10,26	2,31
107	0,66	0,90	1,92	1,99	9,56	4,81
108	0,66	0,90	4,26	7,50	10,43	1,39
109	0,66	0,90	2,60	3,06	10,63	3,47
111	0,66	0,90	4,89	5,92	12,63	2,13
112	0,66	0,90	1,24	4,97	10,33	2,08
113	0,66	0,90	2,95	3,16	11,74	3,72
114	0,66	0,90	6,27	9,54	12,91	1,35
115	0,66	0,90	1,88	3,47	12,18	3,51
116	0,66	0,90	1,30	2,42	7,93	3,28
117	0,66	0,90	1,38	5,13	11,39	2,22
118	0,66	0,90	1,27	3,22	8,53	2,65
119	0,66	0,90	1,55	2,82	12,28	4,35
121	0,66	0,90	1,48	4,73	17,52	3,70
122	0,66	0,90	1,68	4,18	10,44	2,50
123	0,66	0,90	1,41	4,16	10,71	2,57
124	0,66	0,90	1,39	3,36	15,02	4,48
200	0,66	0,90	3,65	3,90	10,15	2,60
201	0,66	0,90	3,77	5,01	11,47	2,29
202	0,66	0,90	4,34	5,24	15,63	2,98

203	0,66	0,90	3,19	3,36	7,85	2,34
205	0,66	0,90	2,24	4,91	9,56	1,95
207	0,66	0,90	3,39	3,63	15,05	4,15
208	0,66	0,90	3,38	3,45	9,42	2,73
209	0,66	0,90	4,37	5,26	8,91	1,69
210	0,66	0,90	4,07	4,60	12,81	2,79
212	0,66	0,90	3,98	4,36	8,67	1,99
213	0,66	0,90	1,95	2,33	7,43	3,20
214	0,66	0,90	3,16	3,52	10,89	3,10
215	0,66	0,90	5,40	5,92	9,65	1,63
217	0,66	0,90	2,47	2,55	10,84	4,25
219	0,66	0,90	1,70	2,91	11,37	3,90
220	0,66	0,90	1,75	3,51	9,93	2,83
221	0,66	0,90	4,13	4,89	12,75	2,61
222	0,66	0,90	6,49	8,34	12,46	1,49
223	0,66	0,90	1,72	2,93	11,11	3,79
228	0,66	0,90	4,63	5,22	12,07	2,31
230	0,66	0,90	2,99	3,29	10,04	3,05
231	0,66	0,90	2,94	3,44	10,18	2,96
232	0,66	0,90	4,05	7,53	8,23	1,09
233	0,66	0,90	2,70	2,80	8,14	2,90
234	0,66	0,90	3,11	3,45	9,22	2,67

Na Tabela 4.4, podemos observar CR, PRD (não PRD1) e FM para o sinal 117 do banco de dados MIT-BIH, comparando com diferentes métodos achados na literatura. Foi utilizada a métrica PRD, e não PRD1, devido a que na literatura se utiliza essa métrica para comparar resultados com os trabalhos já desenvolvidos. Para mostrar os resultados do método proposto foi escolhida a wavelet Bi-ortogonal 6.8 com seus respectivos parâmetros ótimos λ e K , tendo em vista os resultados achados nos testes onde provamos que é a melhor wavelet a se utilizar, no que se refere ao maior valor de FM . Outro dado a se ter em conta é que valores de PRD (não PRD1) que variam entre 0% e 2% são considerados muito bons e entre 2% e 9% são considerados bons, no que se refere à qualidade do sinal reconstruído [70]. Comparando os resultados do método proposto com outros métodos, podemos observar que o proposto tem um desempenho melhor que as técnicas do estado-da-arte na maioria dos casos, comparando os valores de FM. Cada um dos métodos foi explicado brevemente na Seção 2.7. Para o caso do algoritmo que utiliza JPEG2000 [57], o valor do FM tem um valor mais elevado que o algoritmo proposto, porém, com um $CR = 10 : 1$, que é menor ao alcançado com o algoritmo proposto. Como valores de PRD entre 0% e 2% são considerados muito bons, podemos considerar nosso algoritmo comparável ao JPEG2000, ou até melhor, pois conseguimos maior CR e a informação clínica relevante recuperada é de qualidade muito boa (a diferença é pequena para o PRD= 1,38 para o algoritmo proposto contra

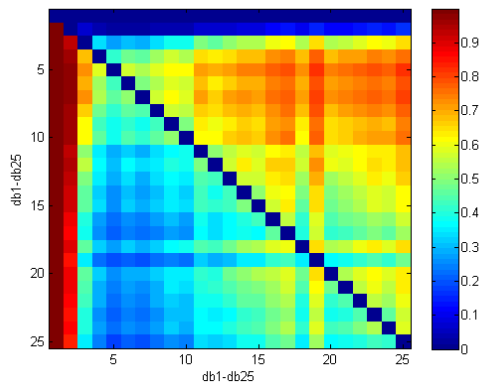
PRD= 1,03 do JPEG). Também cabe destacar que o algoritmo JPEG2000 é um algoritmo que possui mais passos de execução que o proposto, onde se realiza um pre-processamento extenso do sinal (segmentação do sinal, detecção de pulsos, formação da matriz 2D, normalização dos comprimentos dos segmentos) e o processo de transformação e quantização requer mais processamento. O caso do algoritmo proposto, uma vez calculados λ_1 e λ_2 ótimos, é um processo direto onde transformamos com uma família wavelet conhecida, se quantiza com parâmetros conhecidos e se codifica o resultado. Nosso algoritmo então é de implementação mais simples, e mais eficiente. No caso de Chen [23], o algoritmo apresenta um desempenho levemente melhor, ainda comparável com o algoritmo proposto, mas novamente, o algoritmo de Chen possui um processo que leva mais passos na execução, em que se dividem os coeficientes quantizados em 4 grupos de símbolos, e posteriormente se realiza uma codificação aritmética condicional, que codifica de formas diferentes dependendo do grupo de símbolos em questão. O algoritmo proposto, por outro lado, segundo temos explicado, é um método direto fazendo dele um algoritmo mais simples de implementar.

Tabela 4.4: Valores de comparação para CR, PRD e FM do sinal 117 entre o método proposto e métodos de compressão de sinais ECG com wavelets do estado-da-arte.

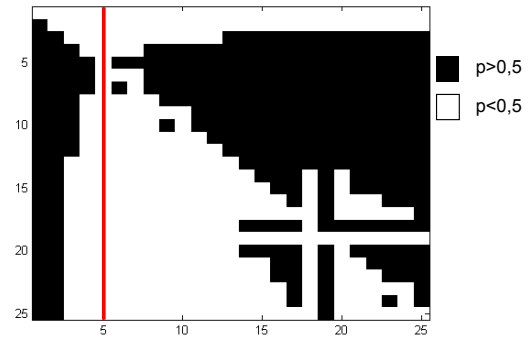
Algorithm	PRD%	CR	FM
JPEG 2000 [57]	1,03	10,00	9,71
Sec&SPIHT [55]	1,01	8,00	7,92
SPIHT&VQ [58]	1,45	8,00	5,52
SPIHT [56]	1,18	8,00	6,78
Hilton [50]	2,60	8,00	3,08
WT&Huffman [51]	5,30	11,60	2,18
Agulhari [46]	4,00	8,36	2,09
Chen [23]	1,08	12,00	11,11
Tohumoglu and Sezgin [52]	5,83	14,90	2,56
Boukhennoufa [53]	2,43	14,30	5,88
Hossain and Amin [54]	2,50	15,10	6,04
Abo-Zahhad [71]	2,80	15,60	5,57
Método proposto	1,38	11,40	8,26

Resumindo o apresentado nesta seção, inicialmente foram calculados os parâmetros de quantização λ_1 e λ_2 ótimos para cada família wavelet, utilizando 7 sinais do banco de dados. Após isso, foi executado o algoritmo para os 41 sinais restantes do banco de dados, já com λ_1 e λ_2 ótimos calculados para cada família wavelet, e os resultados foram expressados como uma média com o respectivo intervalo de confiabilidade desse resultado. Se quis escolher a família wavelet com melhor desempenho, porém, os resultados expressados dessa forma não permitem tirar conclusões. Por esse motivo foi realizado o teste de *Wilcoxon rank sum* para avaliar o FM de cada família para os 41 sinais utilizados. Realizando esse teste, e relaxando a condição de aceitação, conseguiu-se determinar que a família wavelet com melhor desempenho é a bi-ortogonal 6.8. Finalmente se

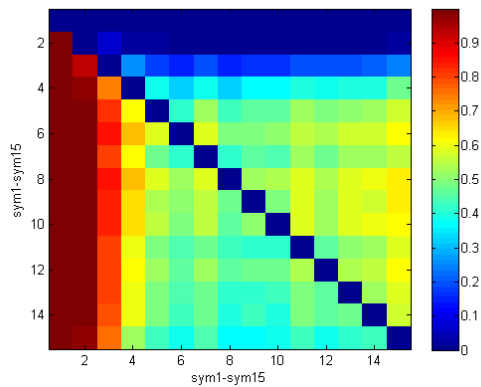
comparou o desempenho do algoritmo proposto com outros métodos do estado-da-arte, usando a melhor família wavelet determinada, com seus respectivos λ_1 e λ_2 ótimos. Podemos dizer que o algoritmo se mostrou robusto para diferentes tipos de sinais, pois λ_1 e λ_2 ótimos foram calculados utilizando 7 sinais, e as famílias com melhor desempenho foram a Daubechies 5, a symlets 7, a coiflets 3, a bi-ortogonal 6.8 e a bi-ortogonal inversa 1.5. Utilizando λ_1 e λ_2 ótimos pro resto do banco de dados, e avaliando o desempenho das famílias wavelets através do teste de *Wilcoxon rank sum*, chegou-se nos mesmos resultados para as melhores famílias wavelets, o que mostra que o algoritmo desenvolvido mantém sua performance para diferentes tipos de sinais ECG. Por esse motivo, pode-se dizer que a família wavelet bi-ortogonal 6.8, efetivamente, tem o melhor desempenho para a compressão de sinais de ECG.



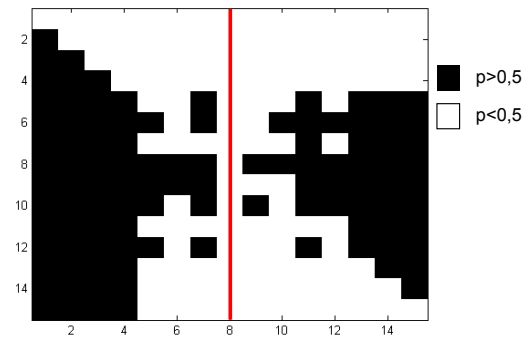
(a) Valores de p fornecidos pelo teste de Wilcoxon na comparação de desempenho de todas as wavelets da família Daubechies com condição de aceitação $p < 0,05$.



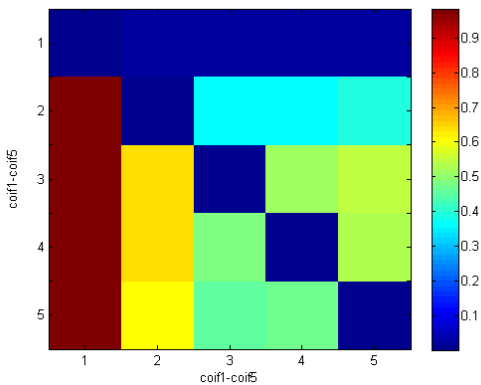
(b) Teste Wilcoxon para Daubechies com $p < 0,5$. A linha vermelha indica a família que vence para todas as outras no teste de Wilcoxon, com a condição de $p < 0,5$.



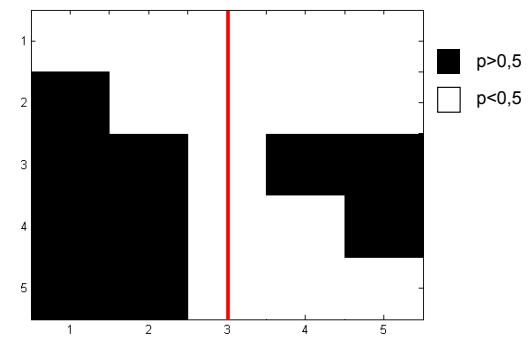
(c) Valores de p fornecidos pelo teste de Wilcoxon na comparação de desempenho de todas as wavelets da família Symlets com condição de aceitação $p < 0,05$.



(d) Teste Wilcoxon para Symlets com $p < 0,5$. A linha vermelha indica a família que vence para todas as outras no teste de Wilcoxon, com a condição de $p < 0,5$.

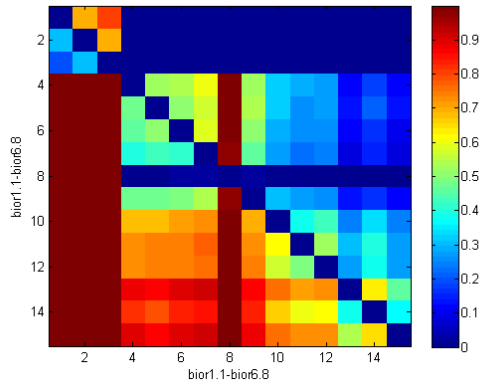


(e) Valores de p fornecidos pelo teste de Wilcoxon na comparação de desempenho de todas as wavelets da família Coiflets com condição de aceitação $p < 0,05$.

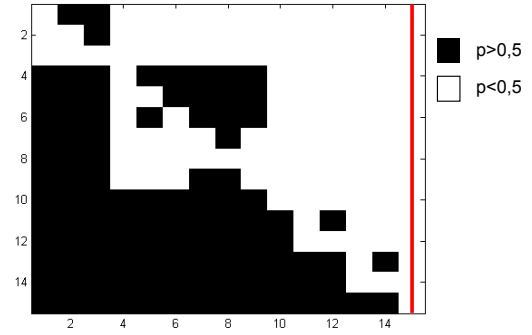


(f) Teste Wilcoxon para Coiflets com $p < 0,5$. A linha vermelha indica a família que vence para todas as outras no teste de Wilcoxon, com a condição de $p < 0,5$.

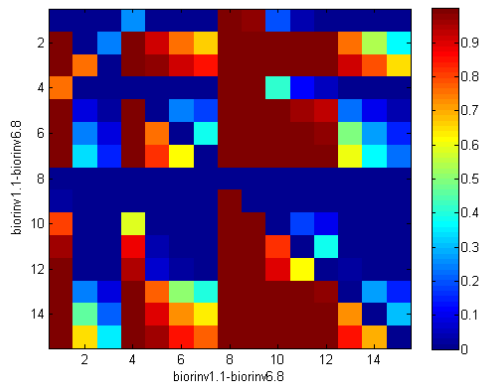
Figura 4.3: Testes de Wilcoxon para cada grupo de famílias wavelets.



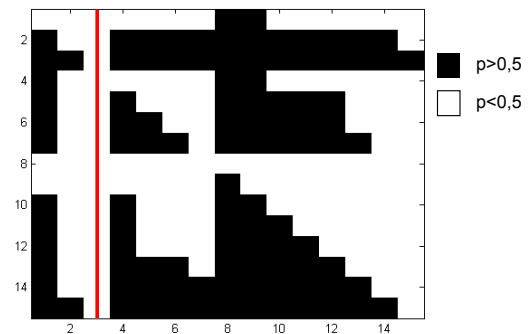
(g) Valores de p fornecidos pelo teste de Wilcoxon na comparação de desempenho de todas as wavelets da família Bi-ortogonal com condição de aceitação $p < 0,05$.



(h) Teste Wilcoxon para Bi-ortogonais com $p < 0,5$. A linha vermelha indica a família que vence para todas as outras no teste de Wilcoxon, com a condição de $p < 0,5$.

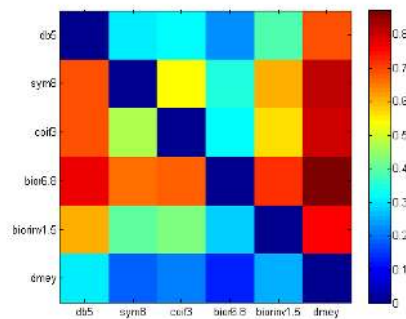


(i) Valores de p fornecidos pelo teste de Wilcoxon na comparação de desempenho de todas as wavelets da família Bi-ortogonal inversa com condição de aceitação $p < 0,05$.

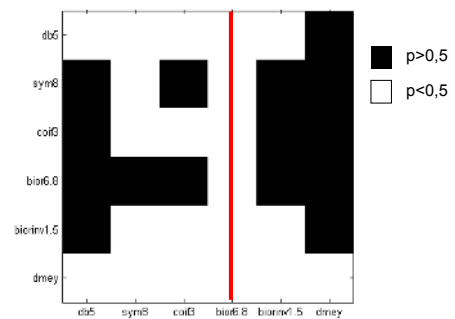


(j) Teste Wilcoxon para Bi-ortogonais Inversas com $p < 0,5$. A linha vermelha indica a família que vence para todas as outras no teste de Wilcoxon, com a condição de $p < 0,5$.

Continuação da Figura 4.3 para os testes de Wilcoxon para cada grupo de famílias wavelets.

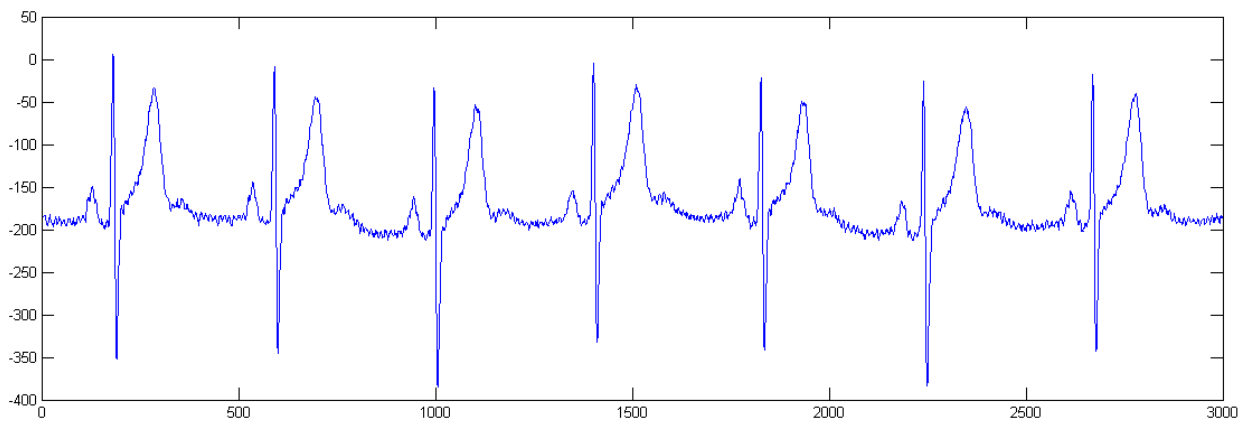


(a) Valores de p fornecidos pelo teste de Wilcoxon na comparação de desempenho das famílias db5, sym8, coif3, bior6.8, biorinv1.5 e dmey com condição de aceitação $p < 0,05$

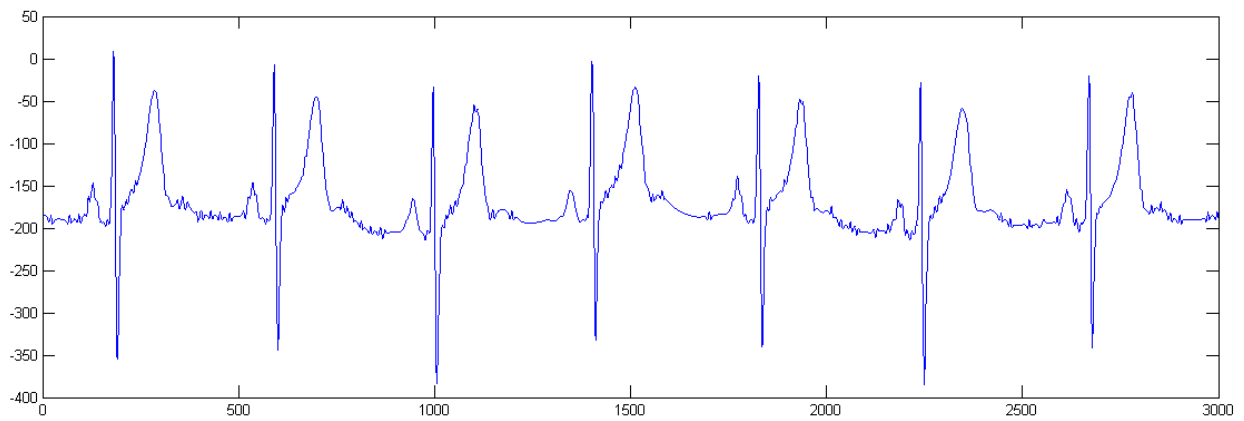


(b) Teste Wilcoxon para as famílias db5, sym8, coif3, bior6.8, biorinv1.5 e dmey com $p < 0,5$. A linha vermelha indica a família que vence para todas as outras no teste de Wilcoxon, com a condição de $p < 0,5$.

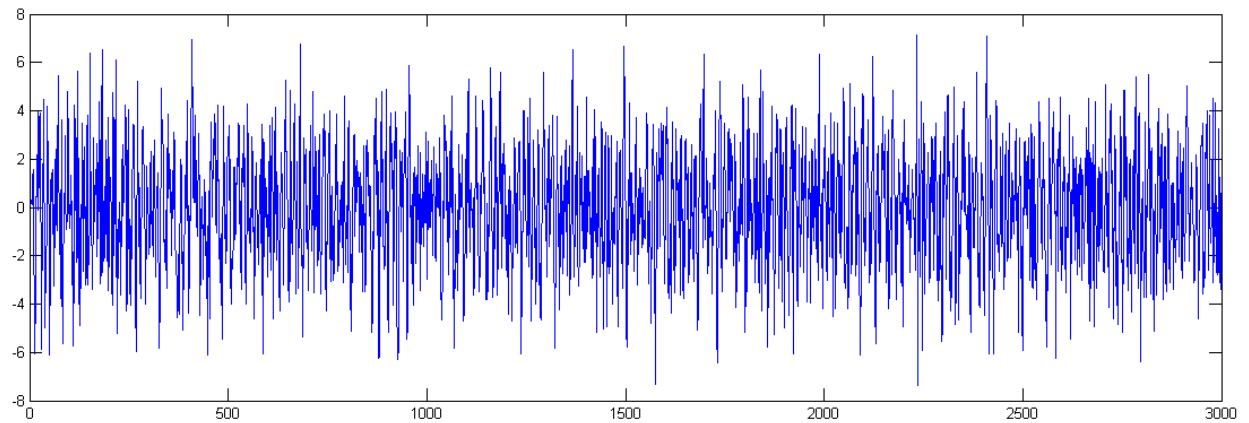
Figura 4.4: Testes de Wilcoxon para as famílias wavelets com melhor desempenho.



(a)



(b)



(c)

Figura 4.5: (a) Sinal 117 original. (b) Sinal 117 reconstruído usando bior6.8, 5 níveis de decomposição, 32 sub-bandas, $\lambda = 0,66$, $K = 0,90$, $\lambda_1 = 0,594$, $\lambda_2 = 0,066$, obtendo uma compressão de CR= 11,40 : 1 e erro de reconstrução PRD= 1,38%. (c) Resíduo entre o sinal 117 original e o sinal 117 reconstruído.

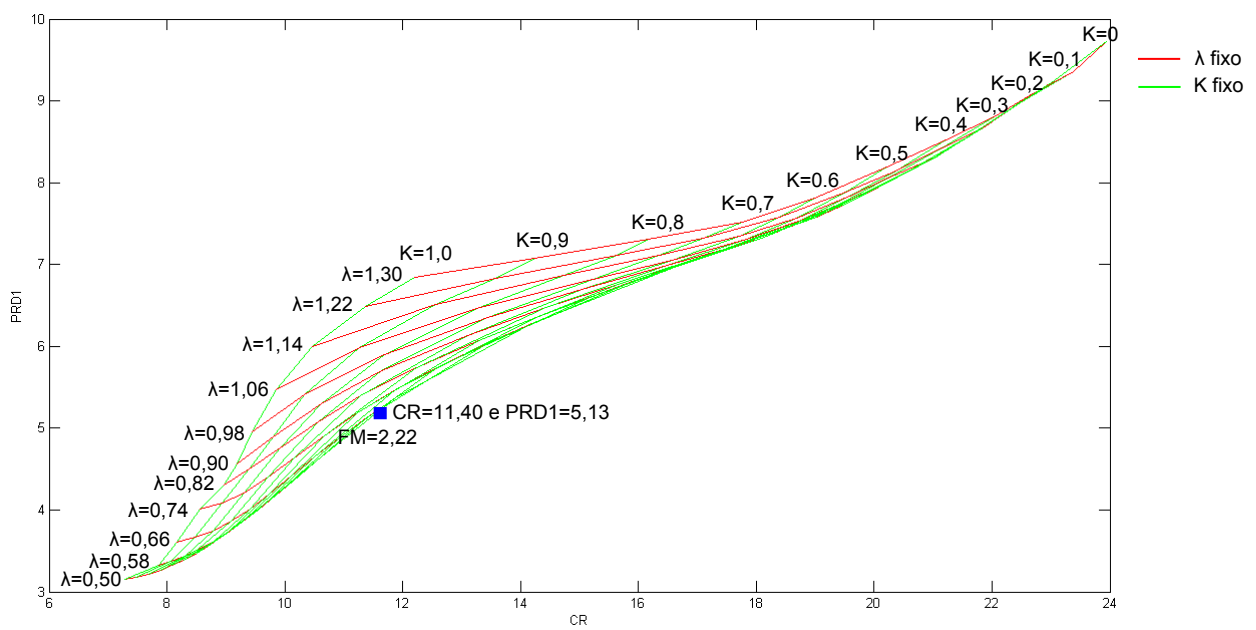


Figura 4.6: Curva de PRD1 em função de CR para o sinal 117 utilizando a transformada Bi-ortogonal 6.8.

5 CONCLUSÃO

Devido à evolução no uso dos sinais ECG, e suas diferentes aplicações, a compressão dele tem se tornado um foco de pesquisa no campo de compressão de sinais biomédicos. Com esse fim, muitos algoritmos e métodos foram desenvolvidos para conseguir melhores níveis de compressão, mas continua sendo um tópico de interesse no qual ainda tem-se vários aspectos que podem ser melhorados. Neste trabalho, desenvolvemos a teoria de compressão por transformadas, explicamos a teoria de transformadas de wavelets, técnicas de quantização e codificação de entropia. Foi proposto um método novo para compressão de sinais de ECG, e foi avaliado seu desempenho e comparado com outros trabalhos do estado-da-arte.

No Capítulo 1, demos uma descrição do que é um sinal de ECG, aplicações dele e as tecnologias desenvolvidas fazendo uso dele. Foi demonstrada a importância da compressão desse tipo de sinal e o motivo da implementação de um método transformado de compressão. Também foram mencionados os objetivos a se cumprir, e quais são as contribuições deste trabalho.

No Capítulo 2, explicou-se com mais profundidade como um sinal de ECG é gerado, a teoria por trás de uma transformada linear, transformada de wavelets contínua, discreta e banco de filtros multi-taxa. Além disso, tratou-se dos conceitos envolvidos na compressão de dados, compressão por transformada, quantização e compressão por entropia. Finalmente, mostraram-se os métodos existentes para compressão de sinais de ECG.

Já no Capítulo 3 expuseram-se o próprio algoritmo proposto e os diferentes estágios nos quais ele consiste. Também explicaram-se os diferentes testes realizados e justificou-se o motivo da realização deles. Fechando esse capítulo, explicou-se como seriam analisados os testes feitos, e que conclusões pretendiam-se tirar a partir deles.

Precedendo esta conclusão, no Capítulo 4, mostraram-se os resultados dos testes realizados, explicando o porquê de cada teste subsequente. Demonstrou-se, outrossim, que o algoritmo proposto se comporta de acordo com o esperado e que se mostra robusto para diferentes sinais de ECG.

Os objetivos propostos foram cumpridos. Conseguiu-se achar as melhores famílias wavelets para compressão de sinais ECG e calcularam-se os parâmetros ótimos para a quantização. Particularmente, concluímos que a família Bi-ortogonal 6.8 mostrou o melhor desempenho na compressão, com parâmetros calculados de valores $\lambda = 0,66$ e $K = 0,90$. Também conseguiu-se propor um método simples para a implementação em relação a outros métodos da literatura.

Os resultados obtidos se mostram satisfatórios, em muitos casos com melhor desempenho que trabalhos do estado-da-arte. O desempenho do algoritmo proposto foi de acordo ao esperado, e existem pontos que podem ser melhorados. A transformação foi feita com famílias wavelets existentes, porém, poderiam se parametrizar os parâmetros dos filtros passa-alta e passa-baixa da decomposição wavelet por banco de filtros, para achar uma transformada wavelet única na qual o

empacotamento da energia seja mais concentrado para os coeficientes transformados. Também, em posteriores pesquisas, poderia ser explorada a relação pai-filho entre os coeficientes wavelet de cada nível de decomposição para obter um maior nível de compressão. O método de quantização proposto se mostrou adequado para a compressão, especialmente o re-ordenamento não linear das magnitudes dos coeficientes, que não tinha sido utilizado em outros trabalhos da literatura. Uma possível melhora para a quantização seria utilizar um sistema híbrido no qual se utilizasse uma limiarização, num primeiro estágio, para zerar coeficientes que contém pouca informação, seguindo um raciocínio similar ao SPIHT (explicado na Seção 2.7). Posteriormente, aplicar uma quantização como a aplicada neste trabalho aos coeficientes restantes, alcançando assim, segundo se esperaria, maiores níveis de compressão sem muita perda de informação.

As métricas utilizadas para avaliar o algoritmo foram o CR e o PRD, que são as mais utilizadas na literatura. Analisando os resultados, observamos que o algoritmo tem um ponto ótimo de funcionamento, que é onde maior FM se obtém, e corresponde a λ e K ótimos. Porém, pode-se conseguir compressão nas regiões delimitadas quando variamos os valores de λ e K , e observamos que, a medida que se aumenta o CR, também aumenta o PRD.

Outros pontos a se ter em conta no futuro é a utilização de outros bancos de dados para a avaliação/validação do algoritmo. Utilizando sinais de ECG de outros bancos de dados, o algoritmo terá que afrontar sinais amostrados com frequências diferentes, resoluções diferentes e comportamentos diferentes. Essas diferenças serviram para testar mais ainda a robustez do algoritmo proposto, que, utilizando a base de dados MIT-BIH, tem-se mostrado satisfatório.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J. Bronzino, *The Biomedical Engineering Handbook*. IEEE Press Editorial Board, 2000.
- [2] “What is the electrocardiogram (ECG)?” pp. 2–7, 2013, MIT DESIGN OF MEDICAL DEVICES. Disponível em www.mit.edu. Último acesso: 15/09/2015.
- [3] A. R. Perez-Riera, R. Barbosa-Barros, and A. Baranchuk, “Current Value of the Electrocardiogram in the 21 st Century,” *Austin Journal of Clinical Cardiology*, vol. 1, no. 2, pp. 1–3, 2014.
- [4] G. Whyte and S. Sanjay, *Practical ECG for Exercise Science and Sports Medicine*. Human Kinetics, 2010.
- [5] MIT-BIH Arrhythmia Database. [Online]. Available: <http://www.physionet.org/physiobank/database/mitdb/>
- [6] D. Salomon, *Data Compression*, 4th ed. Springer, 2007.
- [7] K. Sayood, *Introduction to Data Compression*, 3rd ed. (Elsevier), Morgan Kaufmann Publishers, 2006.
- [8] A. Koski, “Lossless ECG encoding,” *Computer Methods and Programs in Biomedicine*, vol. 52, no. 1, pp. 23–33, 1997.
- [9] S. M. Jalaeddine, C. G. Hutchens, R. D. Strattan, and W. a. Coberly, “ECG data compression techniques—a unified approach.” *IEEE transactions on bio-medical engineering*, vol. 37, no. 4, pp. 329–343, 1990.
- [10] J. R. Cox, F. M. Nolle, H. A. Fozzard, and G. C. Oliver, “AZTEC, a Preprocessing Program for Real-Time ECG Rhythm Analysis,” *IEEE Transactions on Biomedical Engineering*, vol. BME-15, no. 2, pp. 128–129, 1968.
- [11] R. C. Barr, S. M. Blanchard, and D. A. Dipersio, “SAPA-2 is the Fan,” *IEEE Transactions on Biomedical Engineering*, vol. 32, no. 5, p. 337, 1985.
- [12] J. P. Abenstein and W. J. Tompkins, “A New Data-Reduction Algorithm for Real-Time ECG Analysis,” *Biomedical Engineering, IEEE Transactions on*, vol. BME-29, no. 1, pp. 43–48, 1982.
- [13] W. C. Mueller, “Arrhythmia detection program for an ambulatory ECG monitor,” *Biomedical Science Instrumentation*, vol. 14, pp. 81–85, 1978.
- [14] S. O. Aase, R. Nygaard, and J. H. Husøy, “A Comparative Study of Some Novel ECG Data Compression Techniques.”

- [15] X. Wang and J. Meng, "A 2-D ECG compression algorithm based on wavelet transform and vector quantization," *Digital Signal Processing*, vol. 18, no. 2, pp. 179–188, 2008.
- [16] I. Mohammad Rezazadeh, M. Hassan Moradi, and A. Motie Nasrabadi, "Implementing of SPIHT and Sub-band Energy Compression (SEC) Method on Two-Dimensional ECG Compression: A Novel Approach," vol. 4, 2005, pp. 3763–3766.
- [17] H. H. Chou, Y. J. Chen, Y. C. Shiau, and T. S. Kuo, "An effective and efficient compression algorithm for ECG signals with irregular periods," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1198–1205, 2006.
- [18] R. Mahajan and D. Bansal, "Hybrid ECG signal compression system: A step towards efficient tele-cardiology," in *International Conference on Optimization, Reliability, and Information Technology (ICROIT)*, Feb 2014, pp. 437–442.
- [19] P. S. Addison, "Wavelet transforms and the ECG: a review," *Physiological measurement*, vol. 26, no. 5, pp. R155–R199, 2005.
- [20] M. Abo-Zahhad, A. F. Al-Ajlouni, S. M. Ahmed, and R. J. Schilling, "A new algorithm for the compression of ECG signals based on mother wavelet parameterization and best-threshold levels selection," *Digital Signal Processing: A Review Journal*, vol. 23, no. 3, pp. 1002–1011, 2013.
- [21] B. Arvinti and M. Costache, "The Performance of the Daubechies Mother Wavelets on ECG Compression," pp. 14–17, 2014.
- [22] H. L. Chan, Y. C. Siao, S. W. Chen, and S. F. Yu, "Wavelet-based ECG compression by bit-field preserving and running length encoding," *Computer Methods and Programs in Biomedicine*, vol. 90, no. 1, pp. 1–8, 2008.
- [23] J. Chen, F. Wang, Y. Zhang, and X. Shi, "ECG compression using uniform scalar dead-zone quantization and conditional entropy coding," *Medical Engineering and Physics*, vol. 30, no. 4, pp. 523–530, 2008.
- [24] Y. Chompusri, K. Dejhan, S. Yimman, and N. Charbkaew, "Modified beat segmentation for dtw based ecg compression," in *TENCON 2014 - 2014 IEEE Region 10 Conference*, Oct 2014, pp. 1–4.
- [25] a. Ebrahimzadeh and M. Azarbad, "ECG compression using wavelet transform and three-level quantization," in *6th International Conference on Digital Content, Multimedia Technology and its Applications (IDC)*, no. 2, 2010, pp. 2–6.
- [26] K. C. Hung, T. C. Wu, H. W. Lee, and T. K. Liu, "EP-based wavelet coefficient quantization for linear distortion ECG data compression," *Medical Engineering and Physics*, vol. 36, no. 7, pp. 809–821, 2014.

- [27] C.-T. Ku, K.-C. Hung, T.-C. Wu, and H.-S. Wang, "Wavelet-based ECG data compression system with linear quality control scheme," *IEEE transactions on bio-medical engineering*, vol. 57, no. 6, pp. 1399–1409, 2010.
- [28] M. Blanco-Velasco, F. Cruz-Roldán, J. I. Godino-Llorente, J. Blanco-Velasco, C. Armiens-Aparicio, and F. López-Ferreras, "On the use of PRD and CR parameters for ECG compression," *Medical Engineering and Physics*, vol. 27, no. 9, pp. 798–802, 2005.
- [29] R. Merletti and P. A. Parker, *Electromyography - Physiology, Engineering and Non-Invasive Applications*. IEEE Press Editorial Board, 2004.
- [30] N. Townsend, "Medical Electronics," 2001.
- [31] R. E. Klabunde, *Cardiovascular Physiology Concepts*, 2nd ed. Lippincott Williams & Wilkins, 2011.
- [32] S. K. Mitra, *Digital Signal Processing - A computer-based approach*, 2nd ed. McGraw-Hill, 2001.
- [33] B. P. Lathi, *Modern Digital and Analog Communication Systems*, 3rd ed. Oxford University Press, 1998.
- [34] L. Hejfel and E. Roth, "What is the adequate sampling interval of the ECG signal for heart rate variability analysis in the time domain?" *Physiological measurement*, vol. 25, no. 6, pp. 1405–1411, 2004.
- [35] R. J. Ellis, B. Zhu, J. Koenig, J. F. Thayer, and Y. Wang, *Physiological Measurement*, vol. 36, no. 9, pp. 1827–1852, 2015.
- [36] C. S. Burrus, R. a. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, 1998.
- [37] J. D. Gaskill and R. N. Bracewell, *Linear Systems, Fourier Transforms and Optics*. John Wiley & Sons, Inc., 1979, vol. 32, no. 1.
- [38] G. Kaiser, *A Friendly Guide to Wavelets*. Birkhauser, 1994.
- [39] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed. Prentice Hall, Pearson, 2009.
- [40] S. G. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, ELSEVIER, 2009.
- [41] A. Mertins, *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications*. John Wiley & Sons, 1999.
- [42] M. Vetterli and J. Kovačević, *Wavelets and subband coding*, Prentice-Hall (original publisher 1995), Ed., 2007.

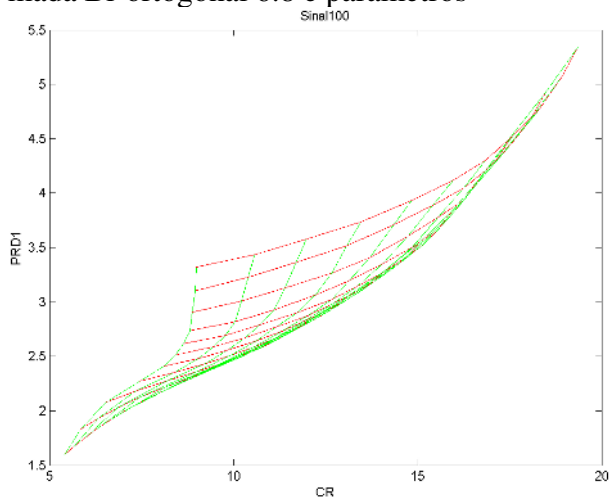
- [43] A. Bultheel and D. Huybrechs, “Wavelets with applications in signal and image processing,” *Course material University of Leuven, Belgium*, p. 196, 2011.
- [44] D. F. Walnut, *An Introduction to Wavelet Analysis*. Birkhauser, 2002.
- [45] R. Polikar. The Wavelet Tutorial. [Online]. Available: <http://users.rowan.edu/~polikar/WAVELETS/WTpart3.html>
- [46] C. M. Agulhari, I. S. Bonatti, and P. L. D. Peres, “An Adaptive Run Length Encoding method for the compression of electrocardiograms,” *Medical Engineering and Physics*, vol. 35, no. 2, pp. 145–153, 2013.
- [47] R. Kumar and S. M. Ieee, “ECG Signal Compression Algorithm based on Joint-Multiresolution Analysis (J-MRA),” in *IEEE SPONSORED 2ND INTERNATIONAL CONFERENCE ON ELECTRONICS AND COMMUNICATION SYSTEM (ICECS 2015)*, 2015, pp. 618–624.
- [48] A. Swarnkar, “Performance of Wavelet Filters for ECG Compression based on Linear Predictive Coding using Different Thresholding Functions,” no. 4, 2014.
- [49] S. Rajankar, “Generalized False Discovery Rate based Adaptive Thresholding Approach for ECG signal Compression,” in *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*, no. Ddc, 2014, pp. 266–271.
- [50] M. L. Hilton, “Wavelet and wavelet packet compression of electrocardiograms,” *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 5, pp. 394–402, 1997.
- [51] A. Al-Shrouf, M. Abo-Zahhad, and S. M. Ahmed, “A novel compression algorithm for electrocardiogram signals based on the linear prediction of the wavelet coefficients,” *Digital Signal Processing*, vol. 13, no. 4, pp. 604–622, 2003.
- [52] G. Tohumoglu and K. E. Sezgin, “Ecg signal compression by multi-iteration ezw coding for different wavelets and thresholds.” *Comp. in Bio. and Med.*, vol. 37, no. 2, pp. 173–182, 2007.
- [53] N. Boukhenoufa, K. Benmahammed, M. Abdi, and F. Djeflal, “Wavelet-based ecg signals compression using spiht technique and vktp coder,” in *3rd International Conference on Signals, Circuits and Systems (SCS)*, 2009.
- [54] M. Hossain and N. Amin, “Ecg compression using sub-band thresholding of the wavelet coefficients.” *Australian Journal of Basic and Applied Sciences*, vol. 5, no. 5, pp. 739–749, 2008.
- [55] I. Rezazadeh, M. Moradi, and A. Nasrabadi, “Implementing of spiht and sub-band energy compression (sec) method on two-dimensional ecg compression: A novel approach,” in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, Jan 2005, pp. 3763–3766.

- [56] Z. Lu, D. Y. Kim, and W. a. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 7, pp. 849–856, 2000.
- [57] A. Bilgin, M. Marcellin, and M. Altbach, "Compression of electrocardiogram signals using jpeg2000," *Consumer Electronics, IEEE Transactions on*, vol. 49, no. 4, pp. 833–840, Nov 2003.
- [58] S. Sahraeian and E. Fatemizadeh, "Wavelet-Based 2-D ECG Data Compression Method Using SPIHT and VQ Coding," in *IEEE International Conference on "Computer as a tool". EUROCON 2007.*, 2007.
- [59] M. Brito, J. Henriques, P. Carvalho, B. Ribeiro, and M. Antunes, "An ECG compression approach based on a segment dictionary and bezier approximations," *European Signal Processing Conference*, pp. 2504–2508, 2007.
- [60] S. Lee, J. Luan, and P. H. Chou, "A new approach to compressing ECG signals with trained overcomplete dictionary," in *4th International Conference on Wireless Mobile Communication and Healthcare: "Transforming healthcare through innovations in mobile and wireless technologies", MOBIHEALTH 2014, Athens, Greece, November 3-5, 2014*, 2014, pp. 83–86.
- [61] C. Wang, J. Liu, and J. Sun, "Compression algorithm for electrocardiograms based on sparse decomposition," *Frontiers of Electrical and Electronic Engineering in China*, vol. 4, no. 1, pp. 10–14, 2009.
- [62] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Compressed Sensing for Bioelectric Signals: A Review." *IEEE journal of biomedical and health informatics*, vol. 2194, no. c, pp. 529–540, 2014.
- [63] H. Mamaghanian, S. Member, N. Khaled, D. Atienza, P. Vandergheynst, and S. Member, "Compressed Sensing for Real-Time Energy-Aware ECG Compression on Wireless Body Sensor Nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 1–10, 2011.
- [64] G.-Y. Cho, S.-J. Lee, and T.-R. Lee, "An optimized compression algorithm for real-time ECG data transmission in wireless network of medical information systems," *Journal of Medical Systems*, vol. 39, no. 1, pp. 1–8, 2014.
- [65] S. K. Mukhopadhyay, M. Mitra, and S. Mitra, "An ECG data compression method via R-Peak detection and ASCII character encoding," *2011 International Conference on Computer, Communication and Electrical Technology, ICCET 2011*, pp. 136–141, 2011.
- [66] W. Kubinger, M. Vincze, and M. Ayromlou, "the Role of Gamma Correction in Colour Image Processing," *Image Processing*, no. 1, pp. 3–6, 1998.
- [67] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

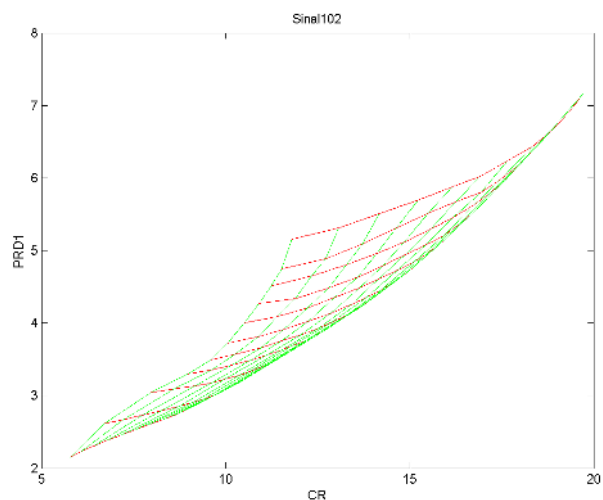
- [68] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 4th ed. McGraw-Hill Europe, 2002.
- [69] J. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference, Fourth Edition: Revised and Expanded*. Taylor & Francis, 2014.
- [70] Y. Zigel, A. Cohen, and A. Katz, “The weighted diagnostic distortion (wdd) measure for ecg signals compression,” *IEEE Trans. on Biomedical Engineering*, pp. 1422–1430, 2000.
- [71] M. M. Abo-zahhad, T. K. Abdel-hamid, and A. M. Mohamed, “Compression of ECG Signals Based on DWT and Exploiting the Correlation between ECG Signal Samples,” *Int. J. Communications, Network and System Sciences*, vol. 7, no. January, pp. 53–70, 2014.

APÊNDICES

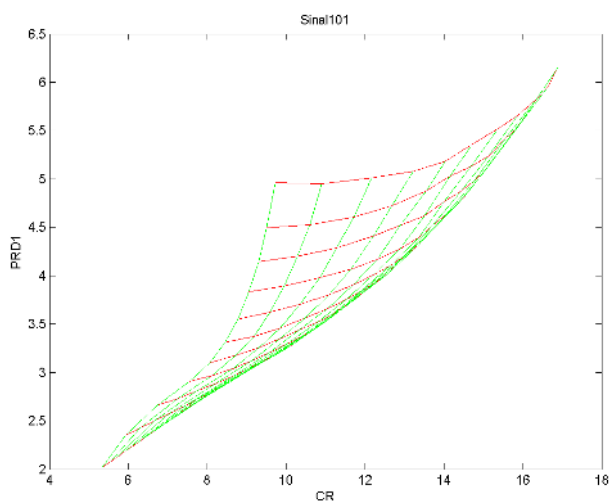
Anexo I: Curvas de desempenho de compressão para todo o banco de dados utilizando a transformada Bi-ortogonal 6.8 e parâmetros



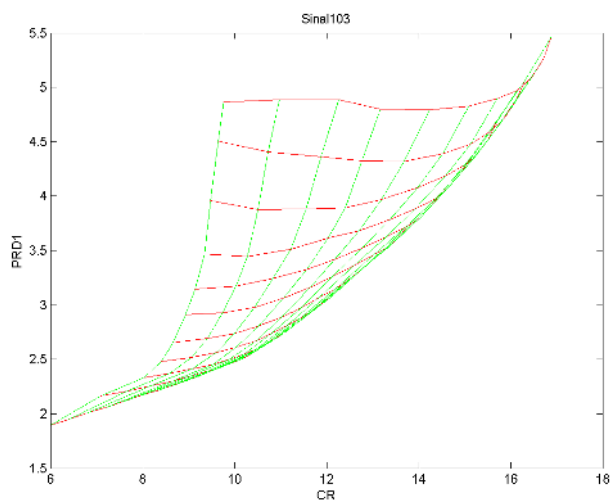
Sinal 100



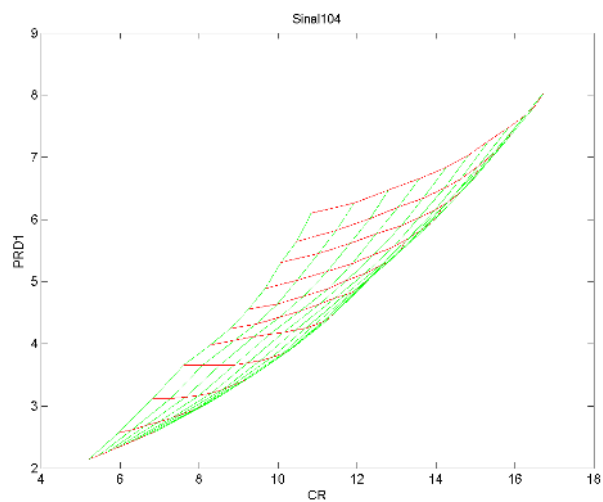
Sinal 102



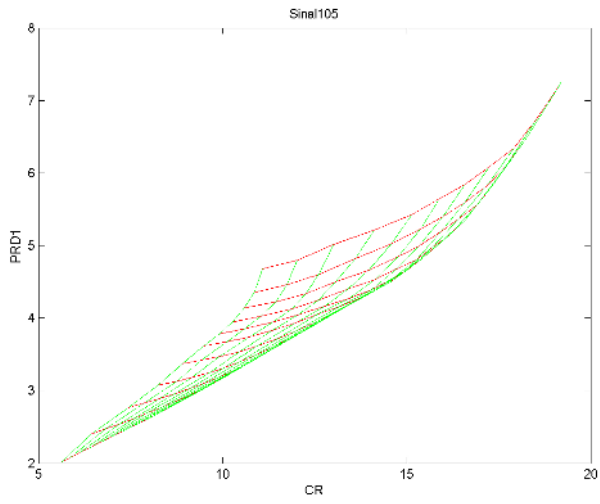
Sinal 101



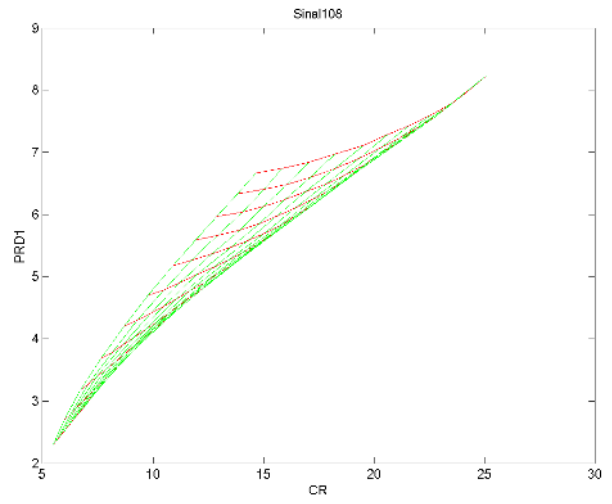
Sinal 103



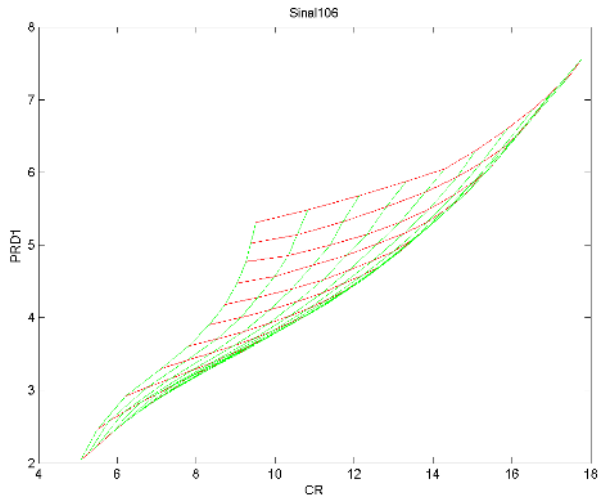
Sinal 104



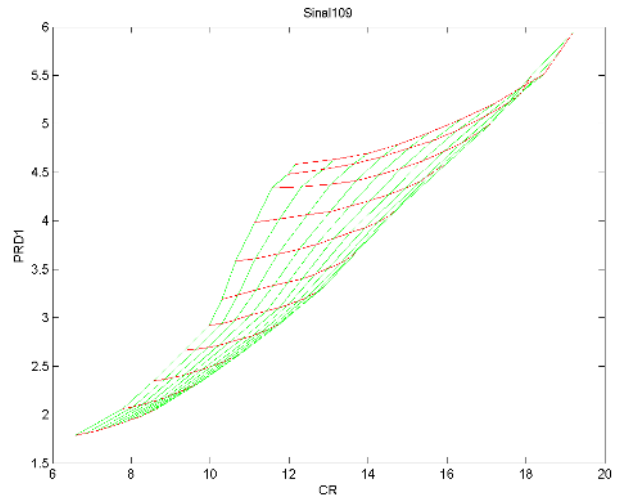
Sinal 105



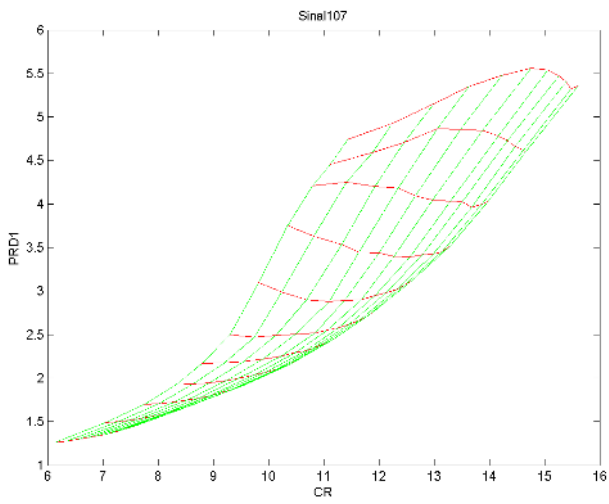
Sinal 108



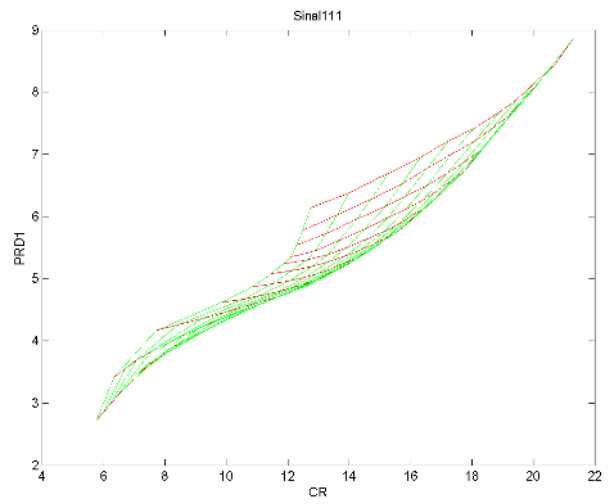
Sinal 106



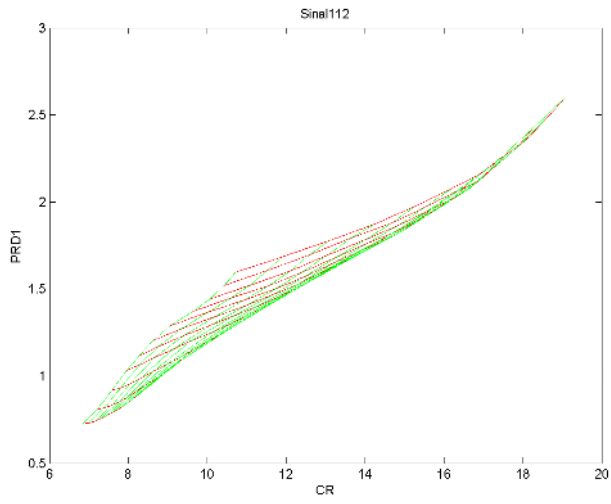
Sinal 109



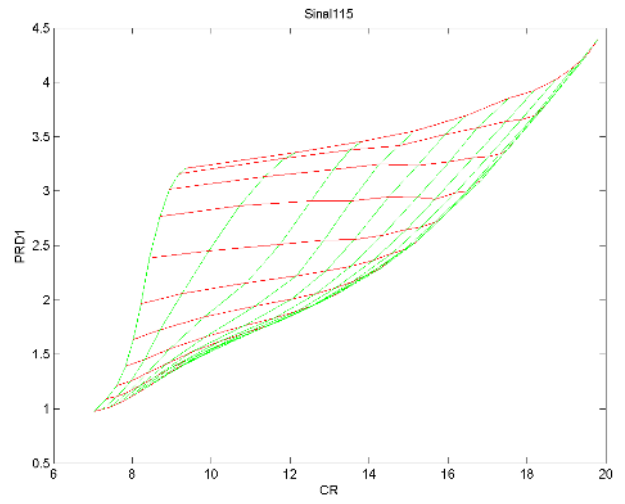
Sinal 107



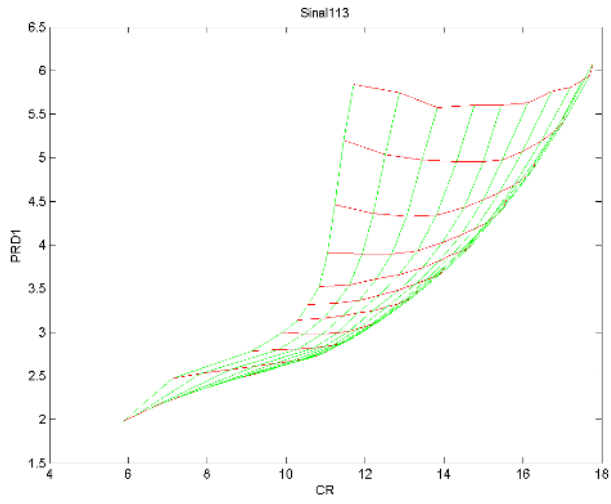
Sinal 111



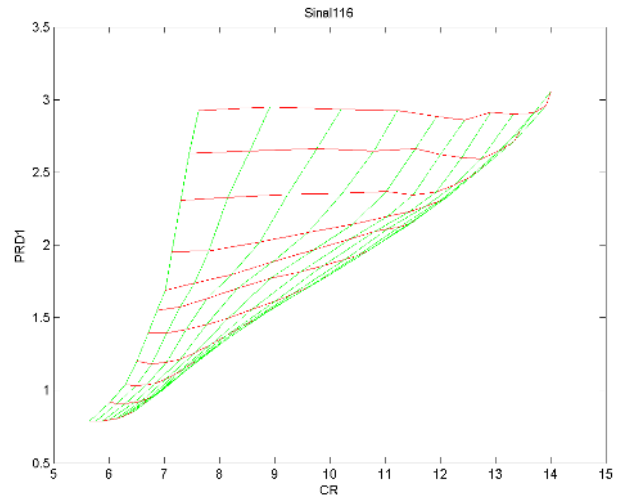
Sinal 112



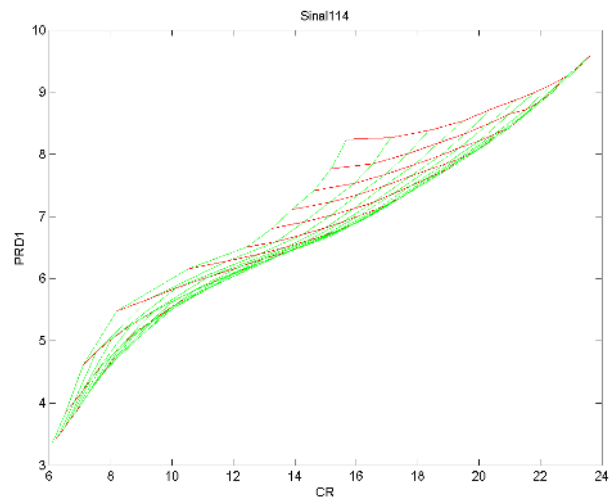
Sinal 115



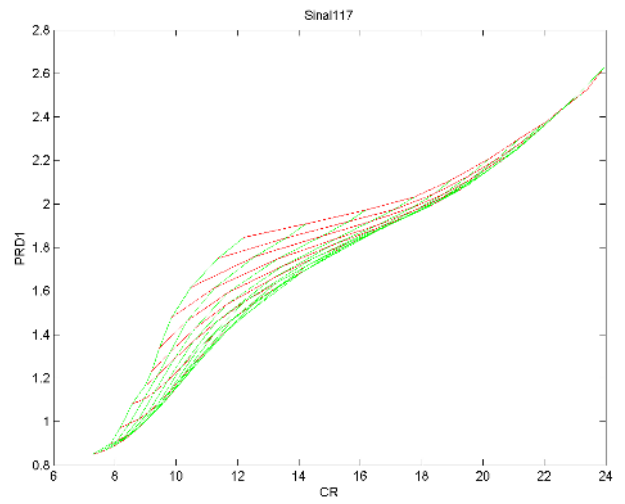
Sinal 113



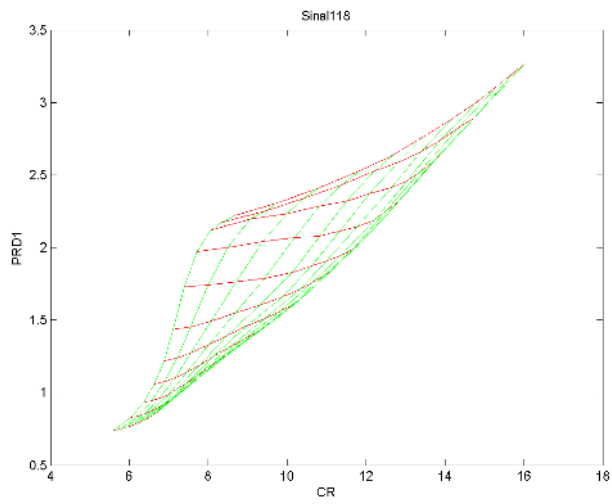
Sinal 116



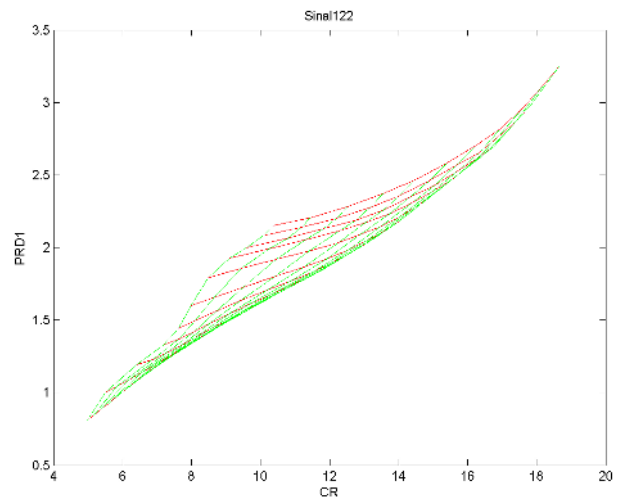
Sinal 114



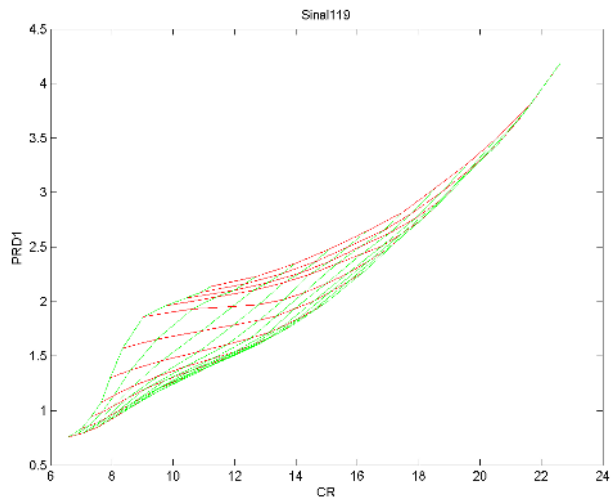
Sinal 117



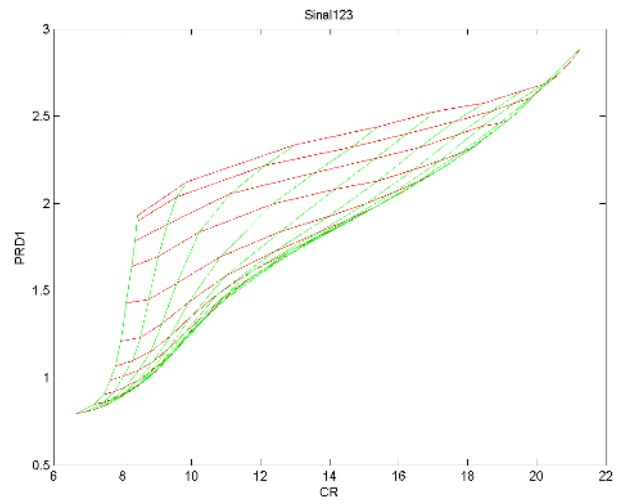
Sinal 118



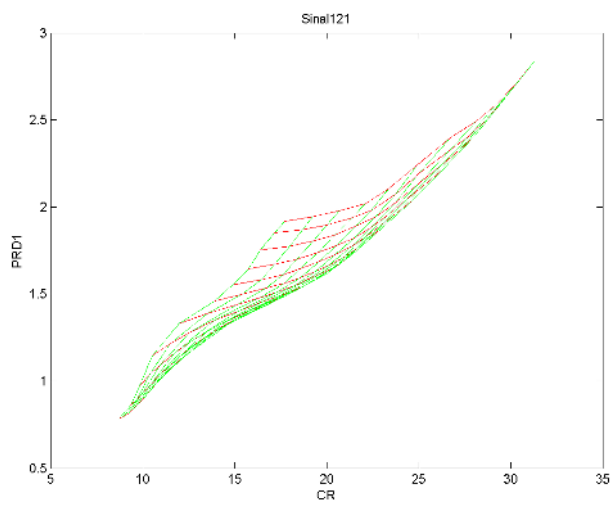
Sinal 122



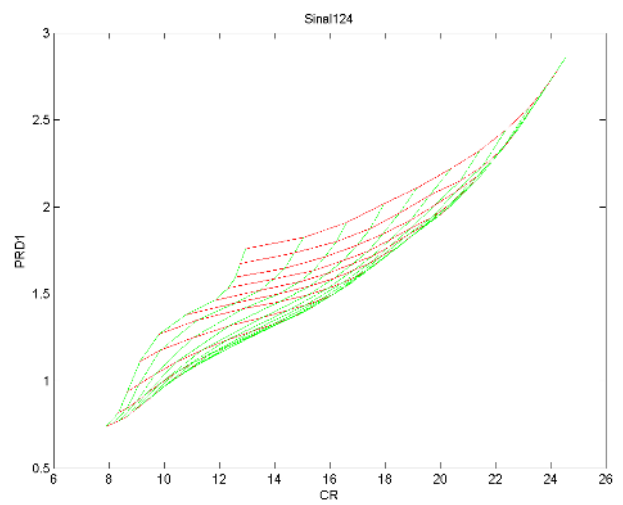
Sinal 119



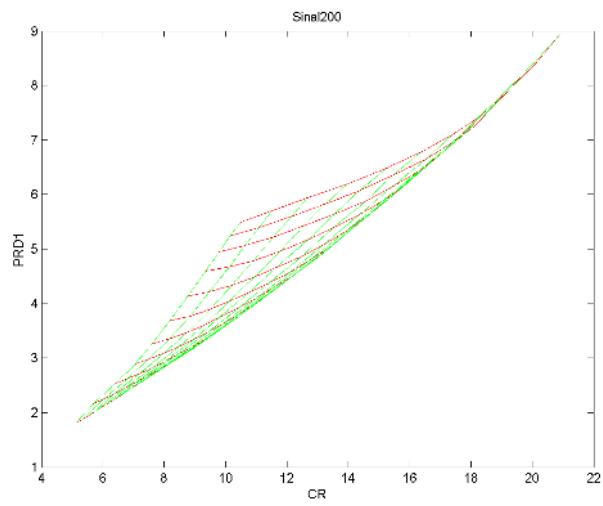
Sinal 123



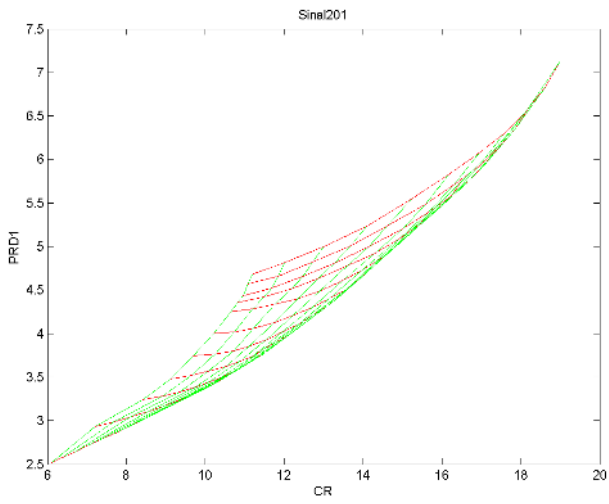
Sinal 121



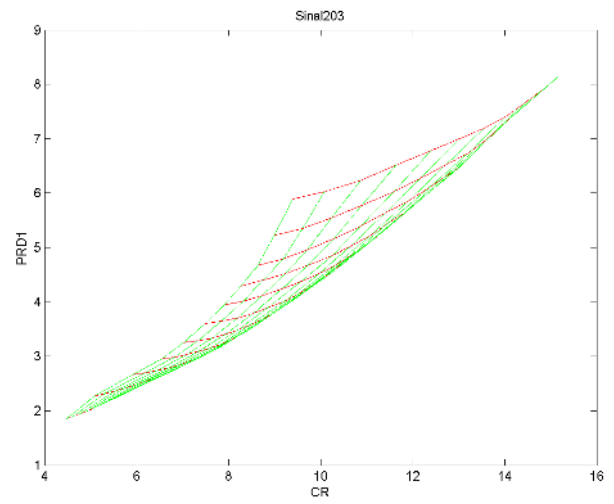
Sinal 124



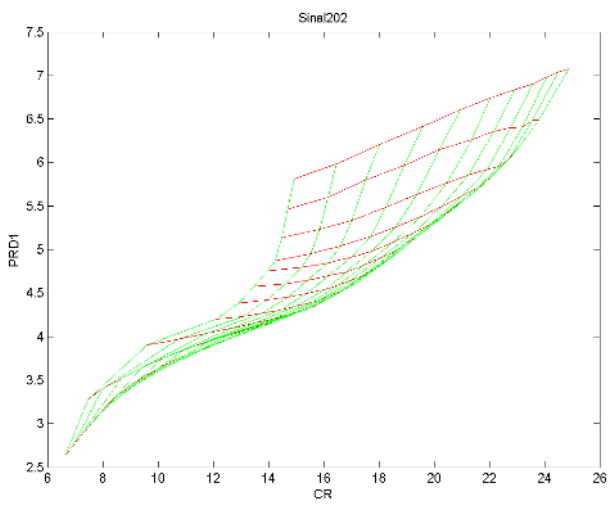
Sinal 200



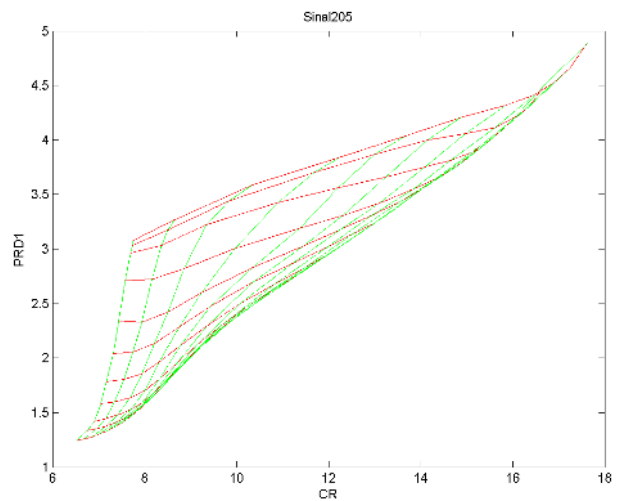
Sinal 201



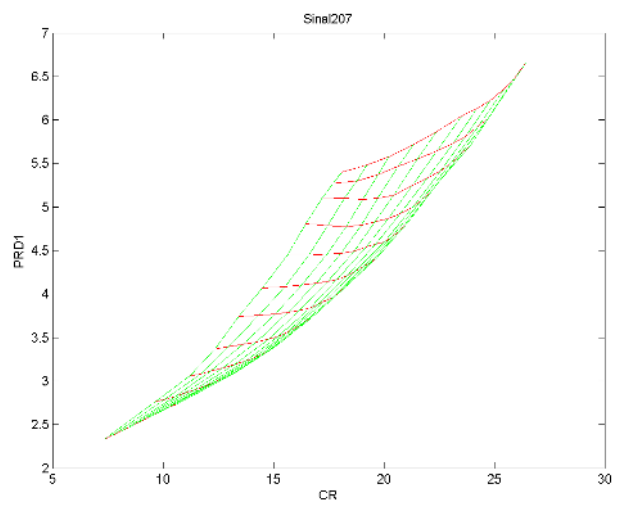
Sinal 203



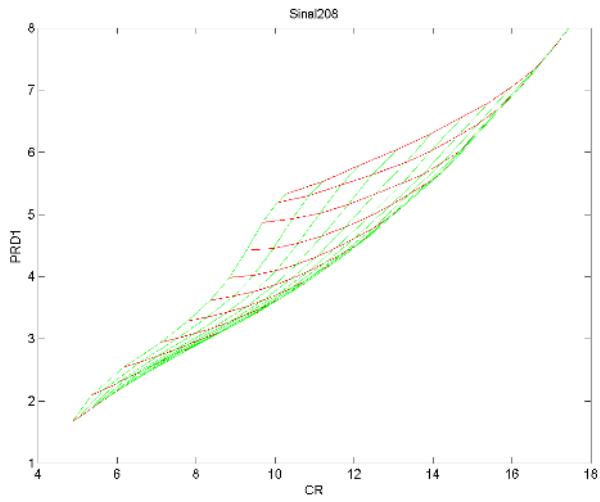
Sinal 202



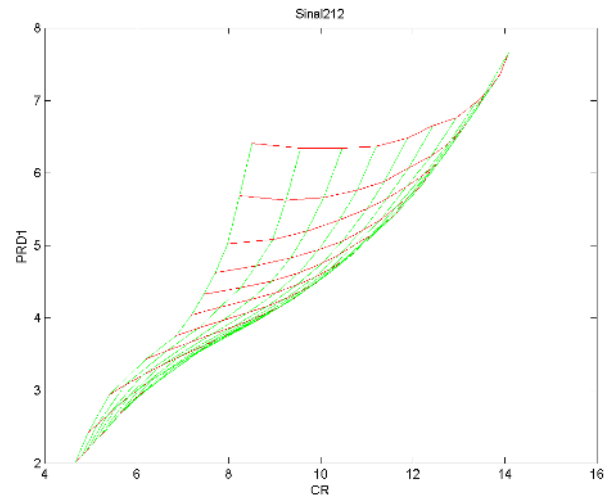
Sinal 205



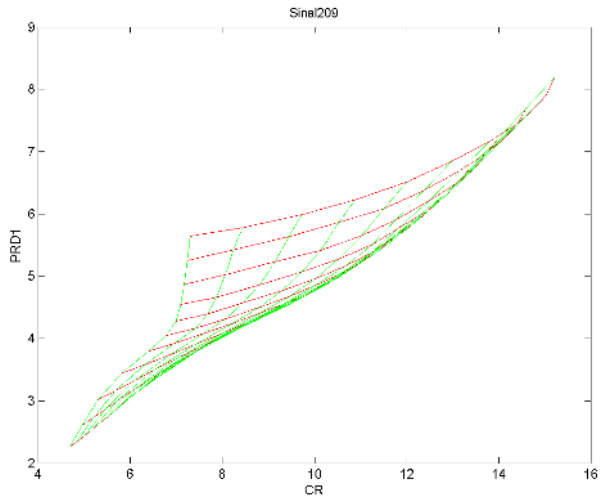
Sinal 207



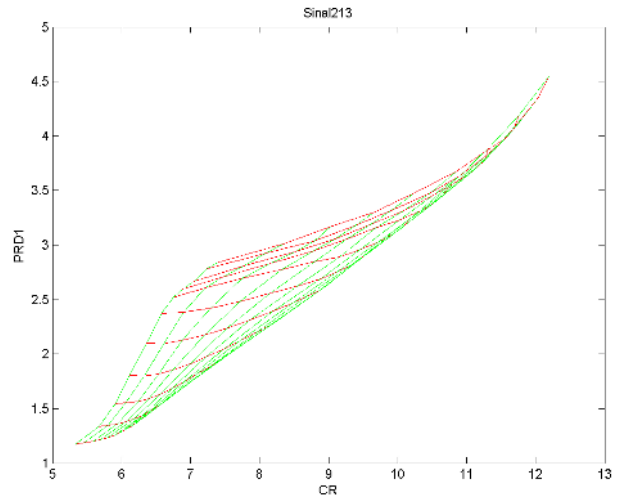
Sinal 208



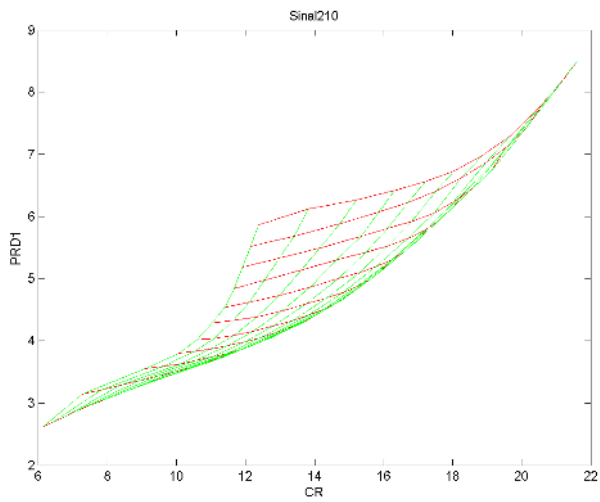
Sinal 212



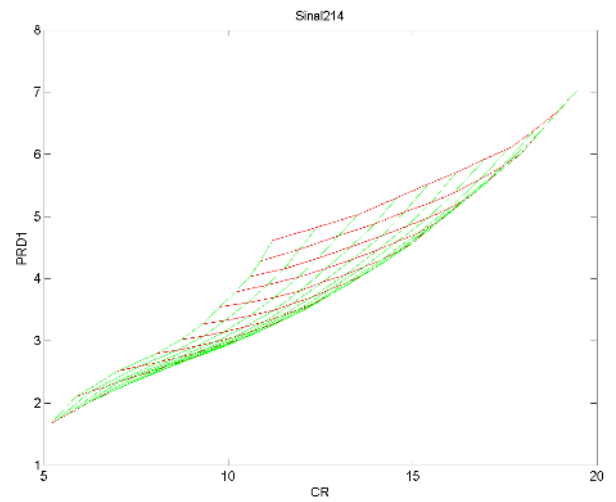
Sinal 209



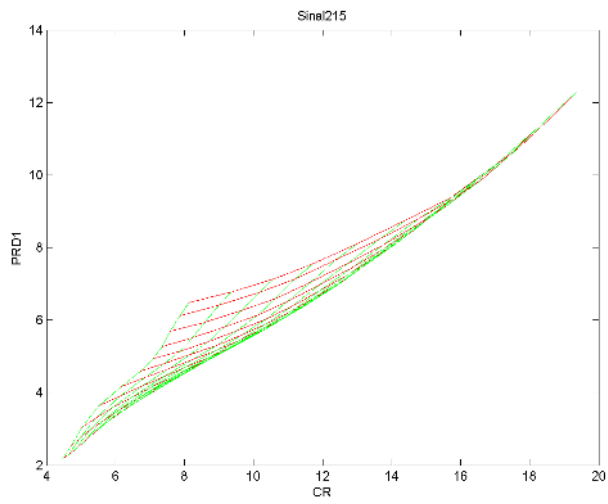
Sinal 213



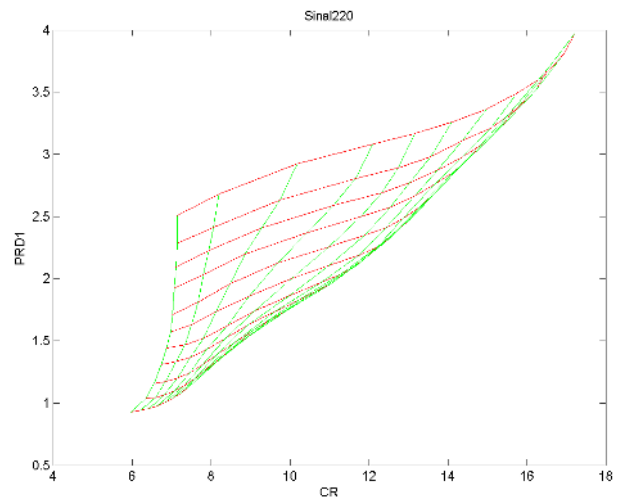
Sinal 210



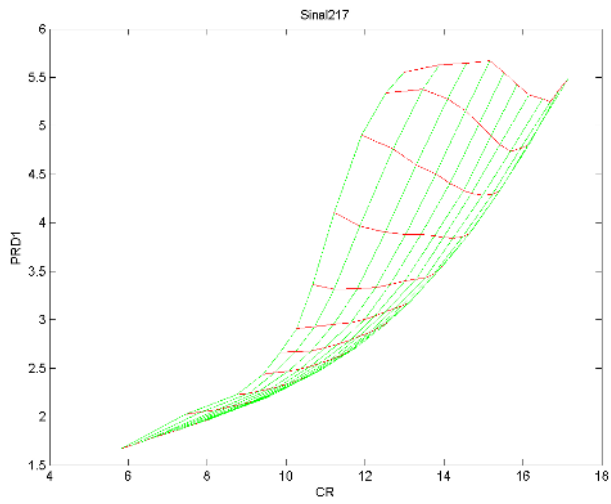
Sinal 214



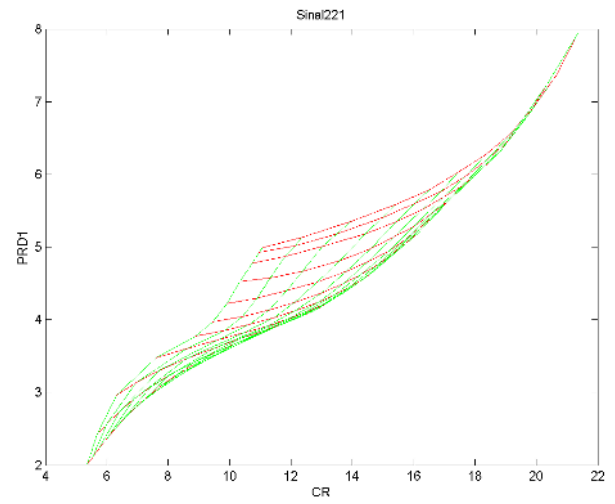
Sinal 215



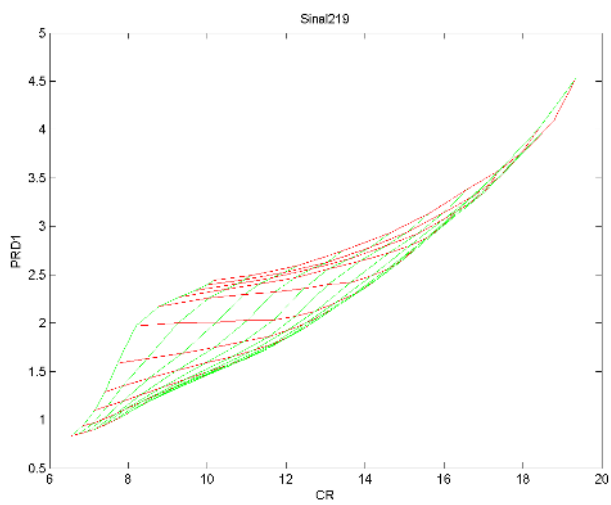
Sinal 220



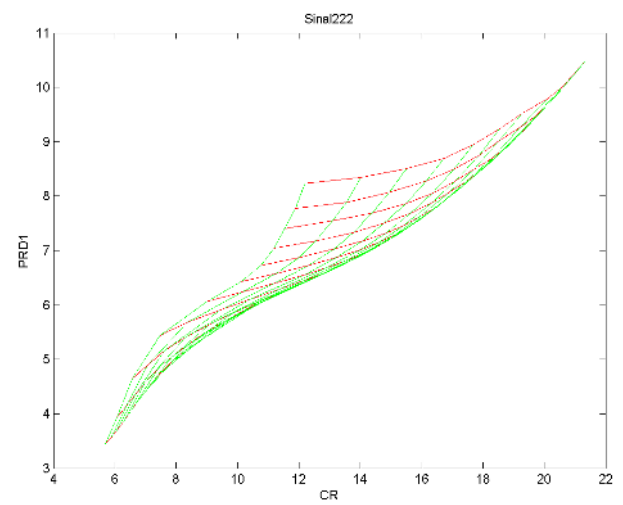
Sinal 217



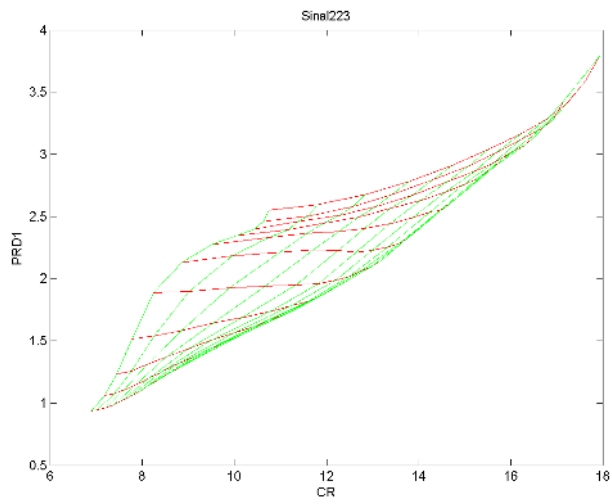
Sinal 221



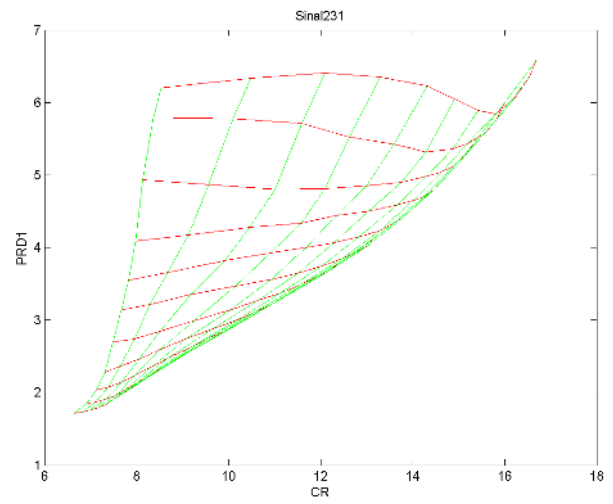
Sinal 219



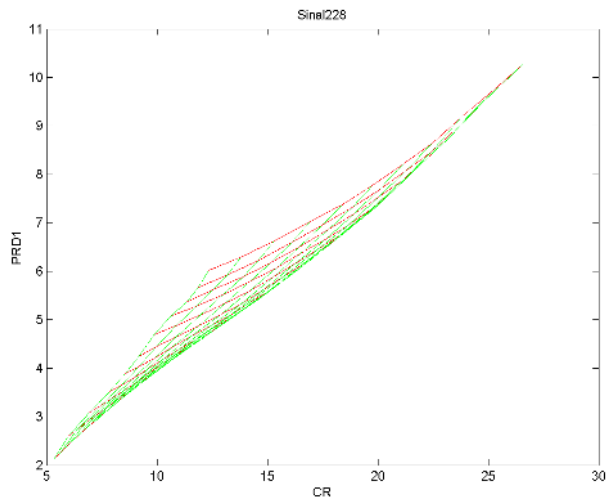
Sinal 222



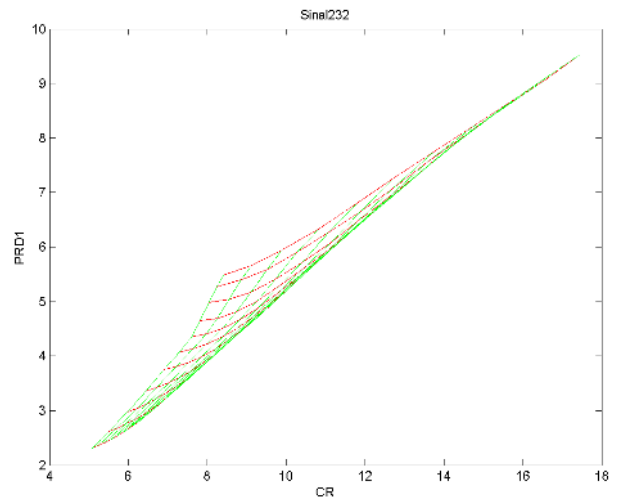
Sinal 223



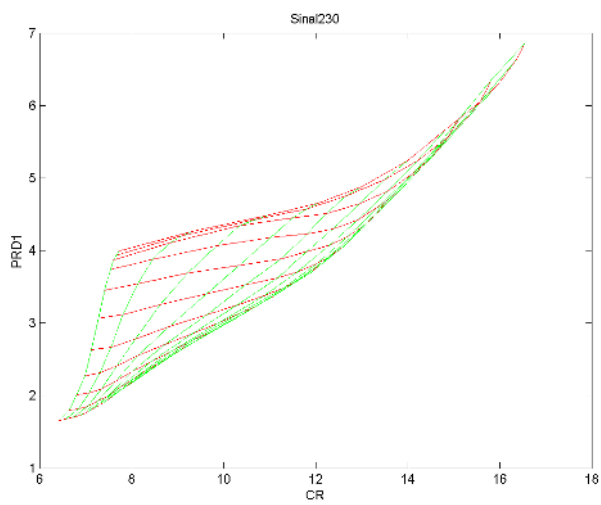
Sinal 231



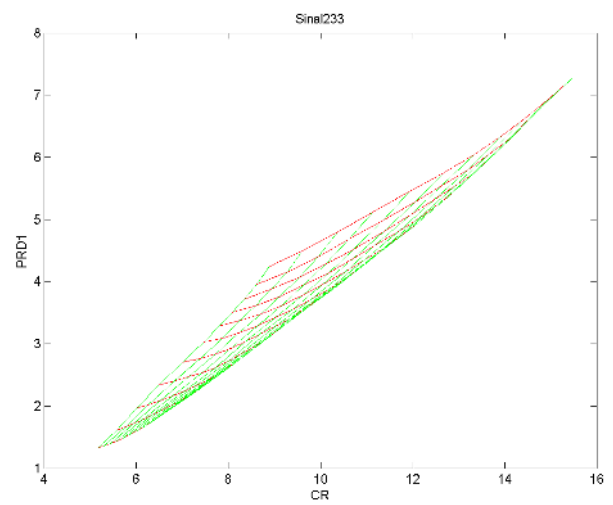
Sinal 228



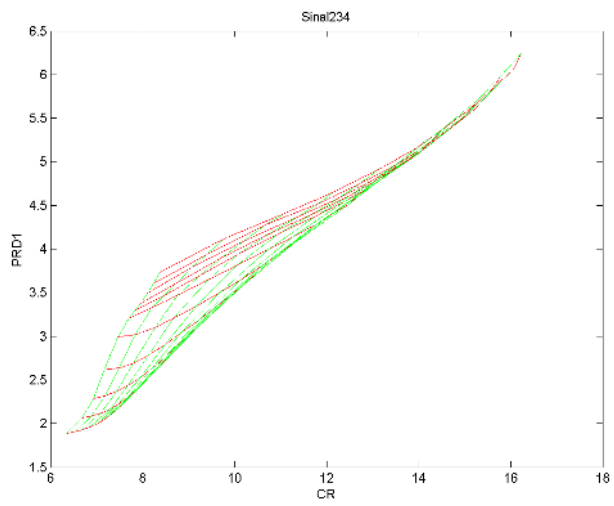
Sinal 232



Sinal 230



Sinal 233



Sinal 234