

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**OTIMIZAÇÃO DE DESEMPENHO DE ALGORITMOS DE  
COMPRESSÃO DE SINAIS BIOLÓGICOS UTILIZANDO  
REDES NEURAS ARTIFICIAIS**

**PEDRO DE AZEVEDO BERGER**

**ORIENTADOR: FRANCISCO ASSIS DE O. NASCIMENTO**

**TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO: PPGENE.TD – 007A/06**

**BRASÍLIA/DF: ABRIL – 06  
UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**OTIMIZAÇÃO DE DESEMPENHO DE ALGORITMOS DE  
COMPRESSÃO DE SINAIS BIOLÓGICOS UTILIZANDO REDES  
NEURAIAS ARTIFICIAIS**

**PEDRO DE AZEVEDO BERGER**

**TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE  
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA  
UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DE GRAU DE DOUTOR.**

**APROVADA POR:**

---

**Francisco Assis de O. Nascimento, Dr.**

---

**Adson Ferreira da Rocha, PhD.**

---

**Lúcio da Silva Martins, Dr.**

---

**Roberto Kawakami, Dr.**

---

**João Souza Neto, Dr.**

**DATA: BRASÍLIA/DF, 6 de Abril de 2006.**

## FICHA CATALOGRÁFICA

BERGER, PEDRO DE AZEVEDO

Otimização de Desempenho de Algoritmos de Compressão de Sinais Biológicos Utilizando Redes Neurais Artificiais [Distrito Federal] 2006.

xx, 200p., 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2006).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Otimização

3. Redes Neurais Artificiais

5. Processamento Digital de Sinais

I. ENE/FT/UnB

2. Compressão

4. Sinais

6. Biológicos

II. Título

## REFERÊNCIA BIBLIOGRÁFICA

BERGER, P. A. (2006). Otimização de Desempenho de Algoritmos de Compressão de Sinais Biológicos Utilizando Redes Neurais Artificiais, Tese de Doutorado, Publicação PPGENE.TD-007A/06, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 200 p.

## CESSÃO DE DIREITOS

AUTOR: Pedro de Azevedo Berger

TÍTULO DA TESE DE DOUTORADO: Otimização de Desempenho de Algoritmos de Compressão de Sinais Biológicos Utilizando Redes Neurais Artificiais.

GRAU: Doutor

ANO: 2006

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. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Pedro de Azevedo Berger

SQS 405 Bloco S Aptº 101

70.239-190 – Brasília/DF – Brasil

## **AGRADECIMENTOS**

Audrey, pelo amor, carinho, companheirismo e paciência. A toda a minha família, em especial aos meus pais a quem devo tudo e um pouco mais, pelo amor dedicado, paciência sem limites e incentivo.

Ao professor Francisco Assis, pela orientação, ensinamentos e amizade.

Aos professores Adson Ferreira, Jake do Carmo e Lúcio Martins por dividirem seus conhecimentos comigo e apoiarem este trabalho.

A CAPES pelo apoio financeiro.

Aos demais professores do ENE – UnB.

## **RESUMO**

### **OTIMIZAÇÃO DE DESEMPENHO DE ALGORITMOS DE COMPRESSÃO DE SINAIS BIOLÓGICOS UTILIZANDO REDES NEURAI ARTIFICIAIS**

**Autor: Pedro de Azevedo Berger**

**Orientador: Francisco Assis de O. Nascimento**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, abril de 2006**

Este trabalho pretende colaborar com o desenvolvimento de tecnologias de codificação de sinais biológicos (sinais eletrofisiológicos e sinais de voz) através da criação, avaliação e, principalmente, a otimização de codificadores utilizando redes neurais artificiais. Durante a pesquisa realizada, desenvolveu-se um codificador para compressão de sinais eletrofisiológicos baseado em transformada *wavelet* discreta e com alocação dinâmica de bits. Na codificação de sinais eletrofisiológicos é necessário garantir a fidelidade da forma de onda do sinal reconstruído, dando certa liberdade para a quantidade de bits necessária para representar a informação. Para outros tipos de sinais, ao contrário, o objetivo é garantir uma taxa de bits por símbolo (quantidade de bits necessária para representar a informação), proporcionando o grau de liberdade para a distorção entre a forma de onda original e a resultante do processo de decodificação. Nesta pesquisa procura-se mostrar que o uso da inteligência artificial pode trazer vantagem em ambos os casos. Para isso, apresenta-se também o desenvolvimento, o estudo e a simulação de um esquema de codificação perceptiva para sinais de voz de banda larga. O esquema é uma versão modificada do codificador de áudio AC-3, para atender, especificamente, às características de sinais de voz de banda larga.

## **ABSTRACT**

### **OPTIMIZATION OF THE PERFORMANCE OF ALGORITHMS OF COMPRESSION OF BIOLOGICAL SIGNALS, BY USING ARTIFICIAL NEURAL NETS**

**Author: Pedro de Azevedo Berger**

**Supervisor: Francisco Assis de O. Nascimento**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, April, 2006**

This paper intends to collaborate with the development of biological signals compression technologies (electrophysiological signals and voice signals) through the creation, evaluation, and mainly, the optimization of codifiers, by using artificial neural nets. During the research, we developed an algorithm for compression of electrophysiological signals based on discrete wavelet transform and with dynamic bit allocation. In the compression of electrophysiological signals it is necessary to guarantee the faithfulness of the wave form of the reconstructed signal, by giving certain liberty to the necessary quantity of bits to represent the required information. For other types of signals, on the contrary, the objective is to guarantee a certain bit rate by symbol (the necessary quantity of bits to represent the information), by providing the right degree of freedom for the distortion between the form of the original wave and that resultant from the decoding process. This research is an attempt to demonstrate that the use of artificial intelligence can bring important advantages in both cases. Therefore, we also present the development, the study and the simulation of a perceptive codification scheme for wideband speech signals. This scheme is a modified version of the audio coder AC-3, to be specifically suitable to the characteristics of wideband voice signals.

# SUMÁRIO

<b>1 - INTRODUÇÃO</b> .....	1
<b>1.1 - OBJETIVO</b> .....	1
<b>1.2 - JUSTIFICATIVAS E REVISÃO BIBLIOGRÁFICA</b> .....	2
1.2.1 - A compressão de sinais eletrofisiológicos .....	2
1.2.2 - A codificação perceptiva para sinais de voz de banda larga. ....	5
<b>1.3 - ORGANIZAÇÃO DO TRABALHO</b> .....	6
<b>2 - CODIFICAÇÃO DE SINAIS UTILIZANDO TRANSFORMADAS ORTOGONAIS</b> 8	
<b>2.1 - A TRANSFORMADA <i>WAVELET</i></b> .....	9
<b>2.2 - A TRANSFORMADA MDCT</b> .....	14
<b>2.3 - CONSIDERAÇÕES FINAIS</b> .....	15
<b>3 - AS REDES NEURAS ARTIFICIAIS E ALOCAÇÃO DINÂMICA DE BITS</b> .....	16
3.1.1 - O processo de treinamento.....	17
3.1.2 - Redes neurais multicamadas .....	20
<b>3.2 - REDES NEURAS DE KOHONEN</b> .....	25
<b>3.3 - CONSIDERAÇÕES FINAIS</b> .....	28
<b>4 - ALGORITMOS PROPOSTOS</b> .....	29
<b>4.1 - ALGORITMO PARA CODIFICAÇÃO DE SINAIS ELETROFISIOLÓGICOS</b> .....	29
4.1.1 - Pré-processamento do sinal eletrofisiológico: <i>de-noising</i> invariante ao deslocamento por transformada <i>wavelet</i> discreta. ....	29
4.1.2 - Descrição do algoritmo de codificação.....	31
<b>4.2 - ALGORITMO PARA CODIFICAÇÃO DE SINAIS DE VOZ DE BANDA LARGA</b> .....	35
4.2.1 - Codificação da envoltória espectral .....	38
4.2.2 - Alocação de bits .....	39
4.2.3 - Considerações finais .....	42
<b>5 - RESULTADOS</b> .....	43

<b>5.1 - RESULTADOS DA COMPRESSÃO DE SINAIS ELETROFISIOLÓGICOS</b>	<b>43</b>
5.1.1 - Resultados do <i>de-noising</i> .....	46
5.1.2 - Resultados da compressão dos sinais eletrofisiológicos .....	51
<b>5.2 - RESULTADOS DO ALGORITMO DE CODIFICAÇÃO DE VOZ DE BANDA LARGA</b>	<b>60</b>
<b>6 - CONCLUSÕES</b> .....	<b>65</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>68</b>
<b>APÊNDICES</b> .....	<b>72</b>
<b>A - RESULTADOS DO <i>DE-NOISING</i> DE SINAIS ELETROFISIOLÓGICOS</b> .....	<b>73</b>
<b>B - RESULTADOS DA COMPRESSÃO DE SINAIS ELETROFISIOLÓGICOS SEM PRÉ-PROCESSAMENTO COM <i>DE-NOISING</i></b> .....	<b>107</b>
<b>C - RESULTADOS DA COMPRESSÃO DE SINAIS ELETROFISIOLÓGICOS SUBMETIDOS AO PRÉ-PROCESSAMENTO COM <i>DE-NOISING</i></b> .....	<b>141</b>
<b>D - RESULTADOS DA COMPRESSÃO DE VOZ DE BANDA LARGA</b> .....	<b>175</b>
<b>E - DETALHES SOBRE O CODIFICADOR AC-3</b> .....	<b>177</b>
<b>E.1 - INTRODUÇÃO</b> .....	<b>177</b>
<b>E.2 - O SISTEMA AC-3</b> .....	<b>177</b>
<b>E.3 - DETALHES DO CODIFICADOR AC-3</b> .....	<b>179</b>
<b>E.4 - A MDCT</b> .....	<b>180</b>
<b>E.4.1 - CODIFICAÇÃO DOS EXPOENTES</b> .....	<b>182</b>
<b>E.4.2 - ALOCAÇÃO DE BITS</b> .....	<b>183</b>
<b>E.4.3 - CÁLCULO DA DENSIDADE ESPECTRAL DE POTÊNCIA</b> .....	<b>183</b>
<b>E.4.4 - INTEGRAÇÃO EM BANDAS CRÍTICAS</b> .....	<b>184</b>
<b>E.4.5 - CÁLCULO DA FUNÇÃO DE ESPALHAMENTO</b> .....	<b>185</b>
<b>E.4.6 - CÁLCULO DA FUNÇÃO DE MASCARAMENTO GLOBAL</b> .....	<b>187</b>
<b>E.4.7 - CÁLCULO DA ALOCAÇÃO DE BITS</b> .....	<b>189</b>
<b>E.5 - QUANTIZAÇÃO DAS MANTISSAS</b> .....	<b>191</b>
<b>E.6 - FUNCIONALIDADES EXTRAS DO SISTEMA AC-3</b> .....	<b>194</b>
<b>E.6.1 - ALOCAÇÃO DELTA DE BITS</b> .....	<b>194</b>
<b>E.6.2 - ACOPLAMENTO DE CANAIS DE ÁUDIO</b> .....	<b>194</b>
<b>E.6.3 - REMATRICIAMENTO DE CANAIS DE ÁUDIO</b> .....	<b>195</b>

## LISTA DE FIGURAS

Figura 2.1 – Codificação por Transformada. ....	9
Figura 2.2 – Análise DWT. ....	14
Figura 3.1 – Treinamento supervisionado. ....	18
Figura 3.2 – Treinamento não-supervisionado. ....	19
Figura 3.3 – Neurônio de McCulloch e Pitts. ....	20
Figura 3.4 – Exemplo de <i>perceptron</i> . ....	21
Figura 3.5 – (a) Coleções linearmente separáveis. (b) A função booleana XOR. ....	22
Figura 3.6 – Uma MLP para resolver o problema XOR. ....	23
Figura 3.7 – Esquema de treinamento de uma rede MLP. O sinal é propagado de forma direta e o erro de forma retropropagada. ....	25
Figura 3.8 – Camada de Kohonen genérica. ....	26
Figura 4.1 – Diagrama de blocos para o <i>de-noising</i> por SIDWT. ....	30
Figura 4.2 – Diagrama de blocos do codificador e decodificador. ....	31
Figura 4.3 – Exemplos de funções de base <i>wavelet</i> da família Daubechies. ....	32
Figura 4.4 – Ilustração do processo de quantização dos coeficientes transformados. ....	34
Figura 4.5 – Janela aplicada ao sinal de voz de banda larga. ....	36
Figura 4.6 - Diagrama de blocos do codificador e decodificador. ....	37
Figura 4.7 – Ilustração da aplicação da MDCT: (a) MDCT direta e (b) MDCT inversa (IMDCT). ....	37
Figura 4.8 – Ajuste da alocação de bits. ....	41
Figura 5.1 – Comparação entre sinais de EMG isométricos e isotônicos. ....	44
Figura 5.2 – Comparação entre o MSE e o limiar de amplitude para os sinais de EMG isotônicos. ....	47
Figura 5.3 – Comparação entre o MSE e o limiar de amplitude para os sinais de EMG isométricos. ....	47
Figura 5.4 – Comparação entre um sinal de EMG original e o seu correspondente após o <i>de-noising</i> . ....	48
Figura 5.5 – Comparação entre um sinal de EMG original e o seu correspondente após o <i>de-noising</i> . ....	49
Figura 5.6 – MSE <i>versus</i> limiar de amplitude ótimo para sinais de ECG. ....	50

Figura 5.7 - Comparação entre um sinal de ECG original e seu correspondente após o <i>de-noising</i> .	50
Figura 5.8 – Comparação entre um sinal de ECG original e seu correspondente após o <i>de-noising</i> .	51
Figura 5.9 – Resultado do algoritmo de compressão para sinais de EMG isométricos sem <i>de-noising</i> .	52
Figura 5.10 – Resultado do algoritmo de compressão para sinais de EMG isotônicos sem <i>de-noising</i> .	53
Figura 5.11 – Comparação entre os resultados do algoritmo proposto neste trabalho e o trabalho de Norris <i>et al</i> para sinais de EMG isométricos.	55
Figura 5.12 – Comparação entre os resultados do algoritmo proposto neste trabalho e o trabalho de Norris <i>et al</i> para sinais de EMG isotônicos.	55
Figura 5.13 - Resultado do algoritmo de compressão para sinais de ECG do banco de dados MIT-BIH Arrhythmia Database, não submetidos ao <i>de-noising</i> .	56
Figura 5.14 – Resultado do algoritmo de compressão para sinais de ECG do banco de dados <i>MIT-BIH Compression Test Database</i> , não submetidos ao <i>de-noising</i> .	57
Figura 5.15 – Comparação dos resultado do algoritmo de compressão para sinais de EMG isotônicos, com e sem <i>de-noising</i> .	58
Figura 5.17 – Comparação dos resultados da compressão de sinais de ECG do banco de dados <i>MIT-BIH Arrhythmia Database</i> , com e sem <i>de-noising</i> .	59
Figura 5.18 – Comparação dos resultados da compressão de sinais de ECG do banco de dados <i>MIT-BIH Compression Test Database</i> , com e sem <i>de-noising</i> .	60
Figura 5.19 – Comparação entre um segmento de sinal de voz original e um segmento de sinal de voz decodificado para uma taxa de 32 kbps.	62
Figura 5.20 – Quantidade de ciclos na busca da alocação de bits ótima – comparação entre o padrão original e o algoritmo proposto.	63
Figura A.1 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 1.	73
Figura A.2 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 2.	73
Figura A.3 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 3.	74
Figura A.4 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 4.	74
Figura A.5 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 5.	75
Figura A.6 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 6.	75
Figura A.7 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 7.	76

Figura A.8 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 8.....	76
Figura A.9 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 9.....	77
Figura A.10 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 10.....	77
Figura A.11 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 11.....	78
Figura A.12 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 12.....	78
Figura A.13 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 13.....	79
Figura A.14 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 14.....	79
Figura A.15 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 15.....	80
Figura A.16 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 16.....	80
Figura A.17 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 17.....	81
Figura A.18 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 18.....	81
Figura A.19 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 19.....	82
Figura A.20 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 20.....	82
Figura A.21 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 21.....	83
Figura A.22 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 22.....	83
Figura A.23 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 23.....	84
Figura A.24 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 24.....	84
Figura A.25 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 25.....	85
Figura A.26 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 26.....	85
Figura A.27 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 27.....	86
Figura A.28 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 28.....	86
Figura A.29 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 29.....	87
Figura A.30 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 30.....	87
Figura A.31 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 31.....	88
Figura A.32 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 32.....	88
Figura A.33 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 33.....	89
Figura A.34 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 34.....	89
Figura A.35 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 35.....	90
Figura A.36 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 36.....	90
Figura A.37 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 37.....	91
Figura A.38 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 38.....	91
Figura A.39 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 39.....	92
Figura A.40 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 40.....	92

Figura A.41 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 41.....	93
Figura A.42 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 42.....	93
Figura A.43 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 43.....	94
Figura A.44 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 44.....	94
Figura A.45 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 45.....	95
Figura A.46 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 46.....	95
Figura A.47 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 47.....	96
Figura A.48 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 48.....	96
Figura A.49 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 49.....	97
Figura A.50 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 50.....	97
Figura A.51 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 51.....	98
Figura A.52 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 52.....	98
Figura A.53 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 53.....	99
Figura A.54 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 54.....	99
Figura A.55 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 55.....	100
Figura A.56 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 56.....	100
Figura A.57 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 57.....	101
Figura A.58 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 58.....	101
Figura A.59 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 59.....	102
Figura A.60 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 60.....	102
Figura A.61 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 61.....	103
Figura A.62 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 62.....	103
Figura A.63 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 63.....	104
Figura A.64 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 64.....	104
Figura A.65 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 65.....	105
Figura A.66 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 66.....	105
Figura A.67 – Resultado do <i>de-noising</i> de sinais eletrofisiológicos n ° 67.....	106
Figura B.1 – Resultado da compressão de sinais eletrofisiológicos sem <i>de-noising</i> n ° 1.....	107
Figura B.2 – Resultado da compressão de sinais eletrofisiológicos sem <i>de-noising</i> n ° 2.....	107
Figura B.3 – Resultado da compressão de sinais eletrofisiológicos sem <i>de-noising</i> n ° 3.....	108
Figura B.4 – Resultado da compressão de sinais eletrofisiológicos sem <i>de-noising</i> n ° 4.....	108
Figura B.5 – Resultado da compressão de sinais de EMG isométricos n ° 5.....	109
Figura B.6 – Resultado da compressão de sinais eletrofisiológicos sem <i>de-noising</i> n ° 6.....	109







Figura C.39 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 39.	160
Figura C.40 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 40.	160
Figura C.41 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 41.	161
Figura C.42 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 42.	161
Figura C.43 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 43.	162
Figura C.44 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 44.	162
Figura C.45 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 45.	163
Figura C.46 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 46.	163
Figura C.47 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 47.	164
Figura C.48 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 48.	164
Figura C.49 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 49.	165
Figura C.50 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 50.	165
Figura C.51 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 51.	166
Figura C.52 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 52.	166
Figura C.53 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 53.	167
Figura C.54 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 54.	167
Figura C.55 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 55.	168
Figura C.56 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 56.	168
Figura C.57 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 57.	169
Figura C.58 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 58.	169
Figura C.59 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 59.	170
Figura C.60 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 60.	170
Figura C.61 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 61.	171
Figura C.62 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 62.	171
Figura C.63 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 63.	172
Figura C.64 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 64.	172
Figura C.65 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 65.	173
Figura C.66 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 66.	173
Figura C.67 – Resultado da compressão de sinais eletrofisiológicos com <i>de-noising</i> n ° 67.	174
Figura D.1 – Exemplos de formas de onda para os sinais de voz codificados a 32, 24 e 16 kbps. ....	176
Figura E.1 – O codificador AC-3. ....	178
Figura E.2 – O decodificador AC-3. ....	179

Figura E.3 – Janela aplicada ao sinal de áudio pleo sistema AC-3.....	180
Figura E.4 – MDCT e IMDCT.....	181
Figura E.5 – Exemplo da MDCT .....	182
Figura E.6 – Exemplo do cálculo da DEP.....	184
Figura E.7 – Resultado da integração em bandas críticas.....	185
Figura E.8 – Modelo da função de espalhamento. ....	186

## LISTA DE TABELAS

Tabela 4.1 – Número de amostras nas 37 bandas em que a faixa entre 0 e 7 kHz é dividida..	40
Tabela 5.1 – Resultados PESQ para diferentes taxas de bits. ....	61
Tabela 5.2 – Escala MOS. ....	61
Tabela 5.3 – Estratégia de codificação e a alocação de bits para as taxas de 32, 24 e 16 kbps. .....	62
Tabela 5.4 – Desempenho do codificador, em termos da razão sinal-ruído. ....	62
Tabela 5.5 – Medida PESQ para todos os sinais de voz do conjunto de teste para uma simulação de perda de 10% dos bloco de coeficientes MDCT. ....	64
Tabela D.1 – Resultados PESQ para 32 kbps. ....	175
Tabela D.2 – Resultados PESQ para 24 kbps. ....	175
Tabela D.3 – Resultados PESQ para 16 kbps ....	175
Tabela E.4 – Parâmetros da função de espalhamento. ....	186
Tabela E.5 – Quantizador assimétrico.....	192
Tabela E.6 – Quantizador de 3 níveis ( $pab(k)=1$ ). ....	192
Tabela E.7 – Quantizador de 5 níveis ( $pab(k)=2$ ). ....	192
Tabela E.8 – Quantizador de 7 níveis ( $pab(k)=3$ ). ....	192
Tabela E.9 – Quantizador de 11 níveis ( $pab(k)=4$ ). ....	193
Tabela E.10 – Quantizador de 15 níveis ( $pab(k)=5$ ) ....	193
Tabela E.11 – Divisão das sub-bandas no acoplamento de canais.....	195
Tabela E.12 – Divisão do espectro em bandas críticas. ....	197
Tabela E.13 – Limiar absoluto de audição .....	198
Tabela E.14 – Ponteiro de Alocação de bits versus SNR.....	199

## LISTA DE SÍMBOLOS, NOENCLATURA E ABREVIACÕES

ADPCM – *Adaptive Differential Pulse Code Modulation*

ANU – Agrupamento Não Uniforme

AU – Agrupamento Uniforme

AZTEC – *Amplitude Zone Time Epoch Coding*

BIH – *Beth Israel Hospital*

CF – Fator de Compressão

DCT – *Discrete Co-sine Transform*

DEP – Densidade Espectral de Potência

DPCM – *Differential Pulse Code Modulation*

DWT – *Discrete Wavelet Transform*

ECG – Eletrocardiografia

EMG – Eletromiografia

EZW – *Embedded Zero-tree Wavelet*

ITU-T – *International Telecommunication Union*

LTP – *Long-Term Prediction*

MDCT – *Modified Discrete Co-sine Transform*

MIT – *Massachusetts Institute of Technology*

MLP – *Multi-Layer Perceptron*

MOS – *Mean Opinion Score*

MSE – *Mean Squared Error*

PESQ – *Perceptual Evaluation of Speech Quality*

PRD – *Diferença Percentual Residual*

SC – *Subband Coding*

SIDWT – *Shift Invariant Discrete Wavelet Transform De-noising*

SNR – *Signal to Noise Ratio*

TDAC – *Time Domain Aliasing Cancellation*

TP – *Turning Point*

# 1 - INTRODUÇÃO

## 1.1 - OBJETIVO

A compressão de sinais tem sido um dos tópicos de relevância na comunidade científica em face da rápida expansão das redes digitais de comunicação e dos diversos sistemas multimídia que estão sendo disponibilizados em todos os segmentos da sociedade. A função da compressão de sinais é reduzir a taxa de bits despendida para representar a informação. Contudo, na compressão de sinais de voz, áudio e vídeo pode-se tolerar níveis mais significativos de distorção no sinal reconstruído em prol de uma maior taxa de compressão. Além disso, o critério de qualidade (ou de similaridade) usado envolve características subjetivas da percepção humana, que não requer uma reprodução exata da forma de onda dos sinais. Por outro lado, as aplicações biomédicas requerem uma excelente reconstrução da forma de onda, mesmo que para isso seja preciso reduzir a taxa de compressão.

Tem-se procurado investigar técnicas adaptativas, combinações híbridas e o uso de inteligência artificial, com o objetivo de se obter uma melhoria de desempenho, seja na fidelidade da forma de onda do sinal reconstruído, seja na minimização da taxa de bits por símbolo.

O uso de inteligência artificial pode proporcionar maior eficiência, capacidade de adaptação e flexibilização nas técnicas de compressão de sinais. Os sinais de interesse neste trabalho são sinais biológicos, quais sejam sinais eletrofisiológicos e sinais de voz. Este trabalho tem por objetivo cumprir duas metas, a saber: (1) o desenvolvimento de um algoritmo específico para a compressão de sinais eletrofisiológicos, como os de eletromiografia e eletrocardiografia, utilizando transformada de *wavelet* e alocação dinâmica inteligente de bits por meio de uma rede neural artificial; (2) o uso de inteligência artificial para a melhoria de desempenho de uma técnica padronizada para a compressão de voz de banda larga. No segundo caso está se referindo ao padrão de codificação de áudio AC-3.

Cabe ressaltar as características dos codificadores que justificaram esta pesquisa. Na meta (1) é estipulado o nível de distorção máximo admitido ou o grau de fidelidade requerido e busca-se reduzir tanto quanto possível a taxa de bits. No caso (2) tem-se o contrário, é estipulada a taxa de bits (quantidade de bits disponível para representar a informação), e busca-se reduzir

tanto quanto possível o nível de distorção. Com este trabalho procura-se mostrar que o uso da inteligência artificial pode trazer vantagem em ambos os casos.

É importante ressaltar que na meta (1), o codificador para sinais eletrofisiológicos foi inteiramente concebido no escopo deste trabalho, enquanto que na meta (2) utilizou-se o padrão AC-3 que foi adaptado pela introdução da inteligência artificial na alocação dinâmica de bits e algumas outras modificações importantes, para que o codificador proposto pudesse tornar-se eficiente na codificação de sinais de voz de banda larga.

## **1.2 - JUSTIFICATIVAS E REVISÃO BIBLIOGRÁFICA**

### **1.2.1 - A compressão de sinais eletrofisiológicos**

A interpretação, decomposição e aplicação de sinais eletrofisiológicos, incluindo o eletrocardiograma (ECG), o eletromiograma (EMG) e outras formas de manifestações fisiológicas, têm fascinado muitos pesquisadores.

O eletromiograma é a técnica utilizada para analisar a atividade elétrica dos músculos. A capacidade de geração de eletricidade dos músculos foi primeiramente constatada por Francesco Redi (século XVII). Os estudos de Redi foram posteriormente comprovados e aprofundados por Guicard Duverney e Luigi Galvani (século XVIII). Desde então, cientistas como Duchêne de Boulogne, L. Lindstrom, J. V. Basmajian e C. J. De Luca contribuíram significativamente para disseminação e aperfeiçoamento do estudo dos sinais eletromiográficos.

O eletrocardiograma é o registro extracelular das variações do potencial elétrico do músculo cardíaco em atividade. As ondas de despolarização e re-polarização que se propagam ao longo das fibras cardíacas podem ser consideradas dipolos em movimento como momentos dipolares variáveis. Estes dipolos geram campos elétricos variáveis que podem ser detectados pela medida da diferença de potencial feita por meio de eletrodos colocados na superfície cutânea.

Atualmente o eletrocardiograma e o eletromiograma são ferramentas valiosas no tratamento de diversas doenças cardíacas e musculares, no estudo da ergonomia e da biomecânica. Nos últimos anos, a área de telemedicina, na qual sinais fisiológicos são transmitidos para especialistas à distância, tem se tornado crescentemente importante. Porém, a transmissão de

sinais eletrofisiológicos pode se tornar um desafio porque esses sinais ou têm uma largura de banda muito grande ou, muitas vezes, têm necessidade de serem registrados em vários canais por uma grande quantidade de tempo. Um dos métodos que podem ser usados para superar esse desafio é o uso de técnicas de compressão de sinais para buscar uma forma mais eficiente de representar a informação.

A compressão de sinais de ECG tem sido extensamente estudada pela comunidade científica. Em termos gerais, podemos classificar as técnicas de compressão de ECG em dois grupos principais:

1. Técnicas dedicadas:

- a. Algoritmos heurísticos: *Amplitude Zone Time Epoch Coding* (AZTEC) (Cox *et al.*, 1968), *Turning Point* (TP) (Mueller, 1978), *Coordinate Reduction Time Encoding System* (Abenstein e Tompkins, 1982), algoritmo FAN (Dipersio e Barr, 1985) e algoritmos de codificação no domínio do tempo modificados, como os algoritmos SLOPE (Tai, 1991) e *Scan Along Polygonal Approximation* (SAPA) (Shahein e Abbas, 1994).
- b. Algoritmos de otimização: *Long-Term Prediction* (LTP) (Nave e Cohen, 1993), algoritmos de análise por síntese (ASEC) (Jalaleddine *et al.*, 1990), e a técnica de *Cardinality Constrained* (Nygaard e Hauglan, 1998).

2. Técnicas genéricas: estas podem ser usadas em uma grande variedade de sinais, incluindo voz, imagem e vídeo. Elas incluem *Differential Pulse Code Modulation* (DPCM), *Subband Coding* (SC) e codificação por transformadas (Istepanian, 2002; Hilton, 1997; Lu *et al.*, 2000).

Atualmente, os algoritmos que têm melhores resultados na compressão de sinais de ECG utilizam a codificação por transformadas ortogonais, e a mais utilizada é a transformada *wavelet* (Istepanian, 2002; Hilton, 1997; Lu *et al.*, 2000). O papel da transformada ortogonal é concentrar a energia do sinal em um número pequeno de coeficientes transformados, reduzindo a quantidade de informação a ser transmitida ou armazenada. Entretanto, em praticamente todos os estudos investigados, os coeficientes transformados são quantizados e codificados com uma quantidade fixa de bits e, portanto, o ganho de compressão é resultado exclusivamente da redução do número de coeficientes a serem armazenados. Uma estratégia

ótima seria calcular, a cada bloco, a quantidade de bits que cada coeficiente significativo realmente necessita, o que chamamos de alocação dinâmica de bits. Porém, o decodificador necessita da alocação de bits para o processo de reconstrução do sinal, e essa informação requer uma quantidade de bits adicional que deve ser armazenada ou transmitida integralmente. Esse impasse também tem acompanhado os estudos sobre a compressão dos sinais eletromiográficos conforme discutiremos a seguir.

As principais técnicas disponíveis para a compressão de EMG podem ser encontradas em (Norris e Lovely, 1995; Guerrero e Maihes, 1997; Wellig *et al*, 1998; Norris *et al*, 2001; Berger *et al*, 2003). Norris e Lovely (1995) investigaram a compressão de sinais de EMG, usando ADPCM (do inglês, *Adaptive Differential Pulse Code Modulation*). Guerrero e Maihes (1997) usaram diferentes métodos de compressão sem perda que foram comparados com outros métodos baseados em transformadas ortogonais. Os métodos baseados em transformadas ortogonais obtiveram melhor desempenho, quando se comparou a taxa de compressão e a relação sinal ruído. Recentemente, o uso da transformada *wavelet* na compressão dos sinais de EMG foi estudado por Wellig *et al*, 1998, Norris *et al*, 2001 e Berger *et al*, 2003. Wellig *et al*, 1998, e Norris *et al*, 2001, propuseram um algoritmo de compressão usando transformada EZW (do inglês, *Embedded Zero-tree Wavelets*) e compararam seus resultados com a transformada *wavelet* tradicional. Porém, somente o trabalho de Norris *et al*, 2001, relatou o uso de sinais de EMG em esforços de tensão isotônicos nos testes do algoritmo de compressão. Mesmo assim, o esquema de alocação de bits usado na codificação foi o mesmo para os sinais de EMG em esforços de tensão isométricos e isotônicos. Em (Guerrero e Maihes 1997; Wellig *et al*, 1998; Norris *et al*, 2001; Berger *et al*, 2003, os coeficientes transformados são quantizados e codificados com uma quantidade fixa de bits, a exemplo dos estudos mais recentes sobre a compressão do eletrocardiograma.

A solução que propomos para comprimir os sinais de ECG e EMG é utilizar uma camada de Kohonen para aproximar a alocação de bits de cada bloco por um vetor de alocação de bits pertencente a um conjunto pré-determinado no treinamento da rede neural (dicionário). Dessa forma, somente o índice do vetor de alocação de bits precisa ser transmitido, viabilizando a alocação dinâmica de bits. Além disso, o algoritmo de compressão proposto é adaptativo. Nesse sentido, a compressão dos sinais eletrofisiológicos se molda às características locais do sinal e ao tipo do sinal (seja ele eletromiograma em esforços isométricos, isotônicos ou

eletrocardiograma). Isto é feito porque entendemos que as características locais dos sinais eletromiográficos em esforços isotônicos e dos sinais de eletrocardiograma requerem uma estratégia adaptativa de alocação de bits.

### **1.2.2 - A codificação perceptiva para sinais de voz de banda larga.**

A maioria dos atuais codificadores de voz realiza a compressão do sinal pelo uso de modelos de produção da voz e pela exploração do fenômeno do mascaramento auditivo. Contudo, esses modelos não são, em geral, apropriados para outros sinais tais como voz ruidosa, sinal contendo vozes de vários locutores, música e ruído de fundo. Por outro lado, codificadores baseados em transformadas e modelos perceptivos podem comprimir eficientemente diferentes tipos de sinais, uma vez que eles não são tão especializados quanto os codificadores baseados em modelos de produção da voz. Entretanto, os codificadores baseados em transformadas não têm em seus modelos perceptivos parâmetros específicos para os sinais de voz de banda larga.

Com o uso crescente da transmissão de sinais de voz por redes de pacote, a robustez dos codificadores em relação à perda de pacotes passou a ser uma característica de grande importância. A informação contida nos pacotes perdidos deve ser substituída de uma forma que minimize a deterioração do sinal de voz reproduzido. Nos codificadores baseados em predição linear, os efeitos dessa perda podem se propagar por vários quadros através da realimentação feita pelo preditor. Os codificadores baseados em transformadas podem, a princípio, apresentar maior robustez a esse tipo de degradação. Primeiro, porque a propagação dos efeitos da perda de um bloco de informações será mais restrita. Segundo, porque talvez os esquemas baseados em transformadas permitam que a informação perdida seja substituída de forma mais satisfatória por versões obtidas por interpolação.

Um outro aspecto a ser considerado é a largura de banda do sinal de voz a ser codificado. Larguras de banda maiores que a largura de banda telefônica (300 a 3400 Hz) propiciam significativas melhorias subjetivas na qualidade da voz codificada. Uma largura de banda de 50 a 7000 Hz não só melhora a inteligibilidade e naturalidade da voz, mas acrescenta também um sentimento de transmissão transparente e facilita o reconhecimento do locutor. Sinais de voz com essa largura de banda são denominados sinais de voz de banda larga. Várias aplicações relevantes demandam esse tipo de voz, por exemplo: sistemas de teleconferência e comunicações audiovisuais interativas multipontos.

Este trabalho propõe um algoritmo de codificação de voz de banda larga, utilizando a compressão por transformadas ortogonais e alocação de bits dinâmica, baseada em um modelo perceptivo da audição humana. Este algoritmo é uma versão adaptada do codificador de áudio AC-3 (Todd *et al.*, 1994; Davidson *et. al.*, 2001; ATSC *Revision A*” Doc. A/52<sup>a</sup>, 2001). As modificações realizadas no padrão original do AC-3 tiveram o objetivo de tornar o modelo perceptivo do mesmo com características voltadas especificamente para os sinais de voz de banda larga e sua transmissão em redes de comutação de pacotes. Dentre as modificações mais importantes podemos destacar:

- A codificação diferencial dos expoentes foi modificada para adequar-se à largura de banda de 50-7000Hz (largura de banda para sinais de voz de banda larga);
- A reutilização de expoentes codificados foi reduzida em relação ao esquema original do AC-3, uma vez que este procedimento torna o codificador muito sensível à perda de pacotes;
- A integração dos coeficientes transformados em bandas críticas foi alterada para atender às novas especificações de largura de banda;
- O cálculo dos limiares de mascaramento foi também alterado para adequar-se ao novo esquema de integração em bandas críticas;
- Foi incluída uma rede neural artificial do tipo MLP (do inglês, *Multi-Layer Perceptron*) na fase de alocação de bits para melhorar o desempenho da busca pela alocação de bits ótima;

### **1.3 - ORGANIZAÇÃO DO TRABALHO**

Ao longo deste trabalho procurou-se, além de apresentar os algoritmos desenvolvidos e seus resultados, revisar os principais conceitos adotados na codificação dos sinais que são objeto da pesquisa de modo a apresentar os subsídios teóricos importantes para o entendimento dos métodos aqui desenvolvidos. Desta forma, o trabalho foi organizado em 6 capítulos.

O capítulo 0 apresenta a organização do documento, com destaque para o objetivo deste trabalho, as principais contribuições realizadas na área de compressão dos sinais objeto do estudo e a metodologia de pesquisa empregada na composição deste documento.

O capítulo 2 apresenta alguns conhecimentos importantes relacionados com a codificação de sinais utilizando transformadas ortogonais, com maior ênfase nas transformadas *wavelet* e MDCT (do inglês, *Modified Discrete Co-sine Transform*), objetos do estudo.

O capítulo 3 mostra os conceitos básicos da alocação dinâmica de bits, principalmente com a utilização das redes neurais artificiais.

O capítulo 4 descreve os algoritmos implementados ao longo deste trabalho, bem como alguns detalhes da implementação.

O capítulo 5 apresenta os resultados de simulação obtidos com os algoritmos implementados.

As conclusões obtidas ao longo deste trabalho e as sugestões para trabalhos futuro, são apresentadas no capítulo 6.

## 2 - CODIFICAÇÃO DE SINAIS UTILIZANDO TRANSFORMADAS ORTOGONAIS

Com o crescente número de aplicações em telecomunicações, armazenar e transmitir eficientemente a informação tornou-se indispensável, visto que a quantidade de memória para armazenar ou a largura de banda necessária para transmitir a informação são meios limitados. Para que a transmissão e o armazenamento sejam feitos de forma eficiente, a compressão desses sinais é indispensável.

As técnicas de compressão de sinais podem ser divididas basicamente em duas categorias: compressão livre de erro e compressão com erro. Métodos de codificação sem perda apresentam fatores de compressão relativamente pequenos quando comparados aos métodos de codificação com perda. Uma das mais importantes técnicas de codificação com perda é a codificação por transformada, e a transformada mais popular, em termos de sistemas implementados e padronizados, é a transformada de co-seno discreta, DCT (do inglês, *Discrete Cosine Transform*). A transformada DCT é usada no padrão mais popular de compressão de imagens, o JPEG (*Joint Picture Expert Group*). Outra transformada muito importante na área de codificação de sinais é a transformada *wavelet* discreta, DWT (do inglês, *Discrete Wavelet Transform*), utilizada no estado da arte para compressão de imagens, no padrão JPEG 2000. Neste capítulo será feita uma breve introdução sobre os conceitos em que está baseada a compressão por transformadas ortogonais, bem como será apresentada a teoria básica da codificação utilizando a transformada *wavelet* discreta e uma variação da transformada DCT amplamente utilizada nos codificadores de áudio, a transformada de co-seno discreta modificada, MDCT (do inglês, *Modified Discrete Co-sine Transform*).

A Figura 2.1 ilustra a estrutura típica de um codificador por transformada. Considere que uma seqüência de  $N$  amostras do sinal a ser codificado seja representada por um vetor  $\underline{x}$   $N$ -dimensional. A idéia básica subjacente à compressão por transformadas é usar a transformação ortogonal para concentrar a energia do sinal em  $M < N$  componentes do vetor transformado. Apenas esses  $M$  coeficientes significativos são quantizados, codificados e transmitidos, os outros são descartados.

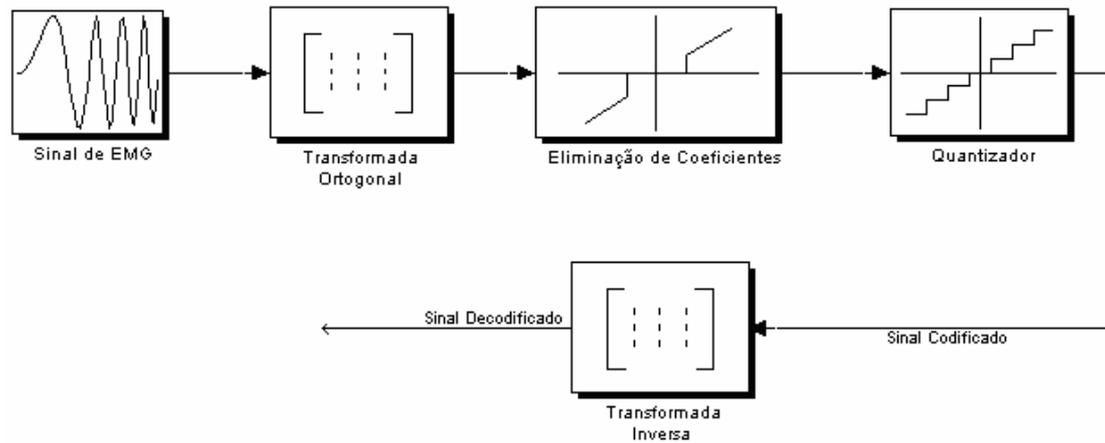


Figura 2.1 – Codificação por Transformada.

Em aplicações para sinais eletrofisiológicos é necessário garantir a fidelidade da forma de onda do sinal reconstruído, enquanto se busca reduzir a quantidade de bits necessária para representar a informação. Desta forma, a eliminação dos coeficientes transformados deve ser realizada de forma controlada e observando-se uma métrica de distorção do sinal reconstruído em relação ao sinal original.

Já na codificação dos sinais de voz realiza-se uma análise do conteúdo espectral do sinal para identificar os componentes do sinal que o ser humano efetivamente ouve. Somente esses componentes são transmitidos, diminuindo-se assim a quantidade de dados a serem enviados. O sistema não envia sinais fracos mascarados pela presença de outros sinais, com frequência próxima da sua, e deixará de codificar quadros (pequenos intervalos de tempo em que dividimos o sinal) que estejam dentro do tempo de mascaramento (cerca de 20 ms antes e 200 ms após). Para isso, é criado um modelo psicoacústico da audição humana baseado em experiências práticas que lograram quantificar, da forma mais precisa possível, este efeito de mascaramento. A partir desse efeito pode ser desenvolvido um codificador perceptivo que analisa o sinal de áudio e determina que coeficientes deverão ser eliminados.

## 2.1 - A TRANSFORMADA *WAVELET*

Nesta última década a transformada *wavelet* mostrou ser uma ferramenta valiosa em muitos campos de aplicação. A transformada *wavelet* de interesse neste trabalho emprega uma base ortogonal com elementos que possuem suporte compacto e determinadas características em frequência.

A transformada *wavelet* é uma decomposição em tempo-escala do sinal analisado por funções de base que são translações e dilatações de uma função protótipo chamada *wavelet* mãe. A *wavelet* mãe  $\psi(t)$  é uma função de média zero, isto é:

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (2.1)$$

As funções de base *wavelet*  $\psi_{s,\tau}(t)$  para um fator de escala  $s$  e uma translação  $\tau$ , são definidas pela equação:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \quad (2.2)$$

Assumindo apenas funções reais, a transformada *wavelet* contínua e sua transformada inversa são definidas como:

$$W(s, \tau) = \int_{-\infty}^{+\infty} x(t) \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) dt \quad (2.3)$$

$$x(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} W(s, \tau) \psi_{s,\tau}(t) d\tau \frac{ds}{s^2}$$

onde  $C_\psi$  é uma constante que depende de  $\psi(t)$ .

Discretizando os parâmetros de translação e escala e escolhendo a *wavelet* mãe apropriadamente, é possível obter-se uma base ortonormal.

Uma opção natural para discretizar o parâmetro de escala  $s$  é fazê-lo de forma logarítmica ( $s=s_0^{-j}$ ). Esse tipo de discretização se assemelha muito à forma como a audição humana divide o espectro de frequências. Para discretizar  $\tau$  pode-se usar a regra de amostragem em qualquer escala ( $\tau=k s_0^{-j}T$ ). As funções de base *wavelet* tornam-se então:

$$\psi_{s,\tau}(t) = s_0^{j/2} \psi(s_0^j t - kT), \text{ onde } j \text{ e } k \text{ são inteiros} \quad (2.4)$$

Se assumirmos que as funções *wavelets* são ortonormais, teremos:

$$\langle \psi_{j,k}(t), \psi_{l,m}(t) \rangle \equiv \int_{-\infty}^{+\infty} \psi_{j,k}(t) \psi_{l,m}(t) dt = \begin{cases} 1, & \text{para } j=l \text{ e } k=m \\ 0, & \text{para } j \neq l \text{ e } k \neq m \end{cases} \quad (2.5)$$

Nesse caso, a transformada *wavelet* e sua inversa são obtidas como:

$$\begin{aligned}
W(k, j) &= \int_{-\infty}^{+\infty} x(t) \psi_{j,k}(t) dt \\
x(t) &= \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} W(k, j) \psi_{j,k}(t)
\end{aligned} \tag{2.6}$$

Sob certas circunstâncias é possível usar dois conjuntos diferentes de funções *wavelets*, um para a análise e outro para a síntese. Esses dois conjuntos devem satisfazer uma condição de biortogonalidade. Seja  $\psi_{j,k}$  o conjunto de funções *wavelet* de análise, e  $\tilde{\psi}_{j,k}$  o conjunto de síntese, o par de equações da transformada *wavelet* e sua inversa e a condição de biortogonalidade são:

$$\begin{aligned}
W(k, j) &= \int_{-\infty}^{+\infty} x(t) \psi_{j,k}(t) dt \\
x(t) &= \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} W(k, j) \tilde{\psi}_{j,k}(t) \\
\langle \psi_{j,k}(t), \tilde{\psi}_{l,m}(t) \rangle &= \begin{cases} 1, & \text{para } j = l \text{ e } k = m \\ 0, & \text{para } j \neq l \text{ e } k \neq m \end{cases}
\end{aligned} \tag{2.7}$$

Para sinais discretos no tempo, a equação (2.4) é geralmente usada com  $s_0=2$ , o que é uma escolha natural para cálculos computacionais. Normalizando o período T para 1 e assumindo que ele é igual ao período de amostragem do sinal, a função *wavelet* mãe  $\psi(t)$  leva à seguinte parametrização bidimensional das funções  $\psi_{j,k}(t)$ :

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k); j, k = 1, 2, \dots \tag{2.8}$$

Os mais atuais sistemas de processamento digital baseados nas transformadas *wavelets* têm seus algoritmos desenvolvidos a partir do que é chamado na literatura de análise em multiresolução. Para explicar a análise em multiresolução usa-se uma função  $\varphi(t)$  chamada de função de escala. Uma família bidimensional de funções de escala, semelhantes às da equação (2.8), pode ser criada pela equação a seguir:

$$\varphi_{j,k}(t) = 2^{j/2} \varphi(2^j t - k); j, k = 1, 2, \dots \tag{2.9}$$

Assim, qualquer função contínua  $x(t)$ , pode ser representada, em uma dada escala ou resolução  $j_0$ , por uma seqüência de coeficientes da expansão:

$$P_{j_0,x}(t) = \sum_{k=-\infty}^{+\infty} x_{j_0}(k) \varphi_{j_0,k}(t) \quad (2.10)$$

Ou seja, a seqüência  $x_{j_0}(k)$  é o conjunto de amostras da função contínua  $x(t)$  em uma dada escala ou resolução  $j_0$ , e  $P_{j_0,x}(t)$  é o projeto de  $x(t)$  no espaço gerado pelas funções  $\varphi_{j_0,k}(t)$ ,  $k \in Z$ .

Sinais discretos podem ser considerados seqüências de amostras de funções contínuas em uma dada escala ou resolução. Nesse caso, não se pode obter informação dos componentes de resolução mais alta do sinal. Entretanto, é útil obter-se, a partir dessas amostras, uma representação de mais baixa resolução do mesmo sinal. Isso pode ser feito impondo-se algumas propriedades às funções de escala. A propriedade principal a ser imposta é o aninhamento dos espaços gerados pelas funções de escala, isto é, para qualquer inteiro  $j$ , o espaço de funções gerado pelas funções de escala  $\{\varphi_{j,k}(t); \text{ para } k=1,2,\dots\}$  deve ser um subespaço do espaço de funções gerado por  $\{\varphi_{j+1,k}(t); k=1,2,\dots\}$ .

O espaço aninhado de funções gerado por  $\varphi(2^j t - k)$  pode ser obtido impondo-se que  $\varphi(t)$  seja representado pelo espaço de funções gerados por  $\varphi(2t)$ . Assim a função de menor resolução  $\varphi(t)$  pode ser representada por uma soma ponderada de versões transladadas da mesma função de escala na resolução seguinte,  $\varphi(2t)$ , conforme a equação:

$$\varphi(t) = \sum_{k=-\infty}^{\infty} g(k) \sqrt{2} \varphi(2t - k) \quad (2.11)$$

O conjunto dos coeficientes  $g(k)$  é chamado de coeficientes de função de escala. Esta equação recursiva é de fundamental importância na teoria de análise em multiresolução e é chamada de equação de refinamento ou dilatação.

Características importantes de um sinal podem ser observadas definindo-se um conjunto de funções *wavelets* que gerem o espaço da diferença entre os espaços gerados por várias escalas das funções de escalas. Ou seja, assume-se que as funções *wavelets* de interesse residem no espaço gerado por uma função de escala em uma resolução imediatamente mais alta,  $\varphi(2t)$ . Dessa forma, as funções *wavelets* podem ser representadas por uma soma ponderada de versões transladadas de uma função de escala na resolução seguinte,  $\varphi(2t)$ :

$$\psi(t) = \sum_{k=-\infty}^{\infty} h(k)\sqrt{2}\varphi(2t-k), \text{ para } k = 1, 2, \dots \quad (2.12)$$

O conjunto de coeficientes  $h(k)$  é chamado de coeficientes da função *wavelet* ou do filtro *wavelet*. As funções calculadas pela equação (2.12) formam as *wavelet* mães para a família de funções definida na equação (2.8).

Os coeficientes de função de escala e os coeficientes da função *wavelet* podem ser relacionados uns aos outros impondo-se algumas restrições. Por exemplo, forçando-se a condição de ortogonalidade nas translações inteiras da função *wavelet* e a condição de que as *wavelets* devem gerar os espaços da diferença, pode-se mostrar que:

$$h(k) = \pm(-1)^k g(1-k), \text{ para } k = 1, 2, \dots \quad (2.13)$$

Agora, impondo-se uma simetria entre a resposta em frequência dos dois filtros, tem-se:

$$h(k) = \pm(-1)^k g(k), \text{ para } k = 1, 2, \dots \quad (2.14)$$

Pode-se mostrar que qualquer função contínua pode ser representada pela seguinte equação definida em termos das funções de escala e das *wavelets*:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_{j_0}(k)\varphi_{j_0,k}(t) + \sum_{k=-\infty}^{+\infty} \sum_{j=j_0}^{+\infty} d_j(k)\psi_{j,k}(t) \quad (2.15)$$

Nessa equação o primeiro somatório fornece a função de resolução  $j_0$ , que é uma aproximação de baixa resolução de  $x(t)$ . No segundo somatório, a cada incremento de  $j$  uma função de maior resolução é adicionada a  $x(t)$ , e dessa forma são adicionados os detalhes de  $x(t)$ . O conjunto de coeficientes representados pela equação (2.15) é chamado de transformada *wavelet* discreta, DWT (do inglês, *Discrete Wavelet Transform*).

Os coeficientes da DWT em uma escala  $j$  podem ser relacionados aos coeficientes na escala  $j+1$  pelas equações:

$$\begin{aligned} c_j(k) &= \sum_{n=-\infty}^{\infty} g(n-2k)c_{j+1}(n) \\ d_j(k) &= \sum_{n=-\infty}^{\infty} h(n-2k)c_{j+1}(n) \end{aligned} \quad (2.16)$$

As equações acima apresentam uma forma recursiva para o cálculo dos coeficientes da DWT. Os coeficientes  $c_j(k)$  são chamados de coeficientes de aproximação e os coeficientes  $d_j(k)$  são chamados de coeficientes de detalhe.

Na prática, assume-se que um sinal discreto na sua escala de resolução original corresponde aos coeficientes de aproximação. Então, se o conjunto de filtros  $h(k)$  e  $g(k)$  são escolhidos de forma que tenham um número finito de coeficientes não nulos, ou seja, são filtros FIR (do inglês, *Finite Impulse Response*), a DWT pode ser calculada pelo banco de filtros da Figura 2.2 até a escala  $j_0$ , onde  $\hat{h}(k)=h(-k)$  e  $\hat{g}(k)=g(-k)$ .

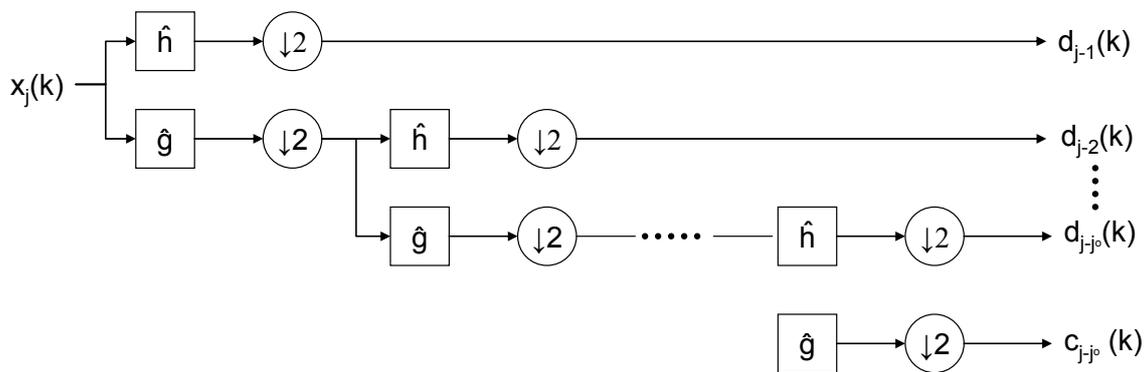


Figura 2.2 – Análise DWT.

## 2.2 - A TRANSFORMADA MDCT

Devido à limitação imposta pelo atraso de codificação, o sinal de áudio ou de voz de banda larga tem que ser subdividido em blocos de amostras transformados individualmente. Para solucionar os diversos problemas provenientes dessa segmentação, várias técnicas foram desenvolvidas, sendo uma das mais utilizadas a MDCT. Esta transformada faz uso de janelas com sobreposição de 50. Esses coeficientes são decimados num fator de dois, o que potencialmente geraria *aliasing* no domínio do tempo. Porém, ao se realizar a transformada inversa, as componentes de *aliasing* se apresentam com polaridade invertida de uma janela para outra, cancelando-se. Essa técnica é chamada de TDAC (*Time Domain Aliasing Cancellation*). Assim, dado um bloco de entrada  $x(n)$ ,  $0 \leq n \leq 2M-1$ , a MDCT direta pode ser definida como (Malvar, 1990; Painter e Spanias):

$$X(k) = \sum_{n=0}^{2M-1} x(n) h_k(n), \quad 0 \leq k \leq M-1 \quad (2.17)$$

onde

$$h_k(n) = w(n) \sqrt{\frac{2}{M}} \cos\left[\frac{(2n+M+1)(2k+1)\pi}{4M}\right], \quad 0 \leq k \leq M-1 \quad (2.18)$$

Como  $w(n)$  é uma janela temporal de comprimento  $2M$  na abordagem de banco de filtros, ela corresponde ao filtro passa-baixas, protótipo a partir do qual são gerados os filtros passa-faixas  $h_k(n)$ ,  $0 \leq k \leq M-1$ , que comporão o banco de filtros. A MDCT inversa (IMDCT), nesse caso, é definida como:

$$x(n) = \sum_{k=0}^{M-1} [X(k)h(k,n) + X^P(k)h(k,n+M)], \quad 0 \leq n \leq M-1 \quad (2.19)$$

onde  $X^P(k)$  denota o bloco precedente de coeficientes MDCTs.

Para que a reconstrução de  $x(n)$  seja perfeita, é preciso que a janela  $w(n)$  satisfaça às seguintes condições:

$$\left. \begin{array}{l} w(2M-1-n) = w(n) \\ w^2(n) + w^2(n+M) = 1 \end{array} \right\}, \quad \text{para } 0 \leq n \leq M-1 \quad (2.20)$$

### 2.3 - CONSIDERAÇÕES FINAIS

No próximo capítulo serão apresentados os conceitos acerca das redes neurais artificiais. A combinação das técnicas de codificação por transformadas ortogonais, aqui estudadas, com os conceitos do próximo capítulo proporcionou a investigação de técnicas adaptativas, combinações híbridas e o uso de inteligência artificial, com o objetivo de se obter uma melhoria de desempenho nos algoritmos de codificação objeto deste trabalho.

### **3 - AS REDES NEURAIS ARTIFICIAIS E ALOCAÇÃO DINÂMICA DE BITS**

O cérebro humano é considerado o mais fascinante processador baseado em carbono existente, sendo composto por aproximadamente 10 bilhões de neurônios (Kovács, 1997). Todas as funções e movimentos do organismo estão relacionados ao funcionamento destas pequenas células. Os neurônios estão conectados uns aos outros através de sinapses e, juntos, formam uma grande rede, chamada rede neural. As sinapses transmitem estímulos através de diferentes concentrações de Na<sup>+</sup> (Sódio) e K<sup>+</sup> (Potássio) e o resultado disto pode ser estendido por todo o corpo humano. Essa grande rede proporciona uma fabulosa capacidade de processamento e armazenamento de informação.

A partir do momento em que as máquinas começaram a evoluir, um grande desejo do homem tem sido a criação de uma máquina que possa operar independentemente do controle humano. Uma máquina cuja independência seja desenvolvida de acordo com seu próprio aprendizado e que tenha a capacidade de interagir com ambientes incertos, uma máquina que possa ser chamada de autônoma, inteligente ou cognitiva.

As redes neurais artificiais consistem em um método de solucionar problemas de inteligência artificial, construindo um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas (Haykin, 1999). São mais que isso: são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento, enquanto os cérebros dos mamíferos podem ter muitos bilhões de neurônios.

A rede neural artificial é um sistema de neurônios ligados por conexões sinápticas e dividido em neurônios de entrada, que recebem estímulos do meio externo, neurônios internos ou ocultos, e neurônios de saída, que se comunicam com o exterior. Em outras palavras, pode-se dizer que uma rede é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma rede neural artificial vem das interações entre as unidades de processamento da rede.

A maioria dos modelos de redes neurais possui alguma regra de treinamento (ou aprendizado), onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, elas aprendem através de exemplos (Kovács, 1997; Haykin, 1999). Arquiteturas neurais são tipicamente organizadas em camadas, com unidades que podem estar conectadas às unidades da camada posterior.

A rede neural passa por um processo de treinamento a partir dos casos reais conhecidos, adquirindo, a partir daí, a sistemática necessária para executar adequadamente o processamento desejado nos dados fornecidos. Sendo assim, a rede neural é capaz de extrair regras básicas a partir de dados reais, diferindo da computação programada, onde são necessários um conjunto de regras rígidas pré-fixadas e algoritmos.

### **3.1.1 - O processo de treinamento**

Redes neurais artificiais possuem a capacidade de aprender por exemplos e fazer interpolações e extrapolações do que aprenderam (Woods, 1986). Um conjunto de procedimentos bem definidos para adaptar os parâmetros de uma rede neural artificial para que a mesma possa aprender uma determinada função é chamado de algoritmo de treinamento. Não há um único algoritmo de treinamento, mas um conjunto de ferramentas representadas por diversos algoritmos (que diferem basicamente pela maneira pela qual o ajuste dos pesos é realizado), cada qual com suas vantagens e desvantagens.

A utilização de uma rede neural artificial na solução de uma tarefa passa inicialmente por uma fase de aprendizagem, onde a rede extrai informações relevantes de padrões de informação que lhe são apresentados, criando assim uma representação própria para o problema. A etapa de aprendizagem consiste em um processo iterativo de ajuste de parâmetros da rede, onde os pesos das conexões entre as unidades de processamento, que guardam, ao final do processo, o conhecimento que a rede adquiriu do ambiente em que está operando.

Diversos métodos para treinamento de redes foram desenvolvidos e podem ser agrupados em dois paradigmas principais (Haykin, 1999): treinamento supervisionado e treinamento não supervisionado.

### 3.1.1.1 Treinamento Supervisionado

Este método de treinamento é o mais comum no treinamento das redes neurais artificiais, sendo chamado treinamento supervisionado (Rosenblatt, 1958) porque a entrada e saída desejadas para a rede são fornecidas por um “professor” externo. O objetivo é ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos. A Figura 3.1 ilustra o mecanismo de treinamento supervisionado. O “professor” indica explicitamente um comportamento adequado ou inadequado para a rede, visando direcionar o processo de treinamento. A rede tem sua saída atual (calculada) comparada com a saída desejada, recebendo informações do supervisor sobre o erro da resposta atual. A cada padrão de entrada submetido à rede, compara-se a resposta desejada (que representa uma ação adequada para ser realizada pela rede) com a resposta calculada, e ajustam-se os pesos das conexões para minimizar o erro. A minimização da diferença é incremental, já que pequenos ajustes são feitos nos pesos a cada etapa de treinamento, de tal forma que estes caminhem - se houver solução possível - para uma solução. A soma dos erros quadráticos de todas as saídas é normalmente utilizada como medida de desempenho da rede e também como função de custo a ser minimizada pelo algoritmo de treinamento.

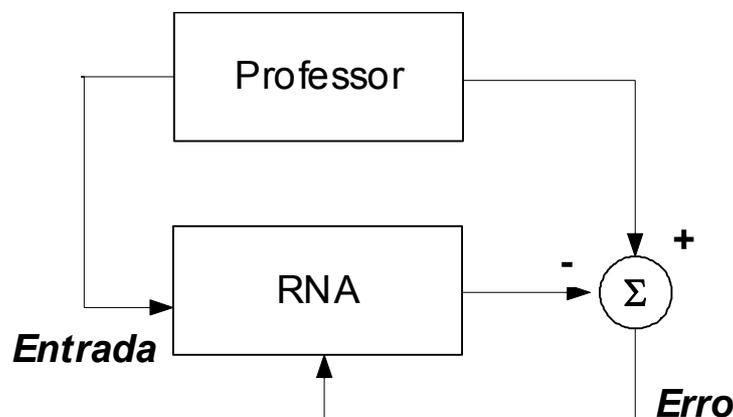


Figura 3.1 – Treinamento supervisionado.

Os exemplos mais conhecidos de algoritmos para treinamento supervisionado são a Regra Delta (Widrow, 1962) e a sua generalização para redes de múltiplas camadas (Rumelhart, 1986), o algoritmo *Backpropagation*.

O treinamento supervisionado pode ser implementado basicamente de duas formas: *off-line* e *on-line*. Para treinamento *off-line*, os dados do conjunto de treinamento não mudam, sendo que uma vez obtida uma solução para a rede, esta deve permanecer fixa. Caso novos dados

sejam adicionados ao conjunto de treinamento, um novo treinamento, envolvendo também os dados anteriores, deve ser realizado para se evitar interferência no treinamento anterior.

Por sua vez, no treinamento *on-line*, o conjunto de dados muda continuamente, sendo que a rede deve estar em contínuo processo de adaptação.

### 3.1.1.2 Treinamento Não-Supervisionado

No treinamento não-supervisionado, como o próprio nome sugere, não há um “professor” para acompanhar o processo de treinamento (Hebb, 1949; Kohonen, 1988). Este método é ilustrado na Figura 3.2.

Apesar da semelhança entre o treinamento supervisionado com o treinamento dos seres humanos, muitos dos sistemas biológicos ocorrem através de treinamento não-supervisionado como, por exemplo, os estágios iniciais dos sistemas de visão e audição.

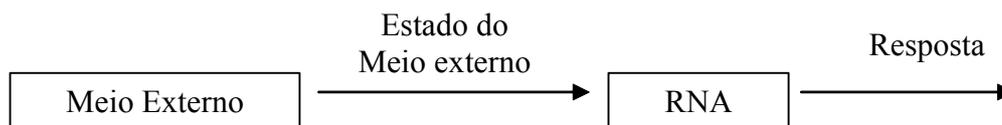


Figura 3.2 – Treinamento não-supervisionado.

Para esses algoritmos, somente os padrões de entrada estão disponíveis para a rede, ao contrário da aprendizagem supervisionada, onde o conjunto de treinamento possui pares de entrada e saída. A partir do momento em que a rede estabelece uma harmonia com as regularidades estatísticas da entrada de dados, desenvolve-se nela uma habilidade de formar representações internas para codificar características da entrada e criar novas classes ou grupos automaticamente. Este tipo de treinamento só se torna possível quando existe redundância entre os padrões de entrada. Sem redundância seria impossível encontrar quaisquer padrões ou características desses dados.

A estrutura do sistema de treinamento não-supervisionado pode tomar uma variedade de formas diferentes. Ela pode, por exemplo, consistir de uma camada de entrada, uma camada de saída, conexões diretas da entrada para a saída e conexões laterais entre os neurônios da camada de saída. Um outro exemplo é uma rede de conexões diretas com múltiplas camadas, onde a livre organização procede na base de camada por camada.

Nesses dois exemplos, o processo de treinamento consiste em modificar repetidamente o peso sináptico de todas as conexões do sistema em resposta às entradas.

### 3.1.2 - Redes neurais multicamadas

Dentre os vários tipos de redes neurais artificiais existentes, as redes multicamadas ou MLP (do inglês, *Multi-layer Perceptrons*) são as redes mais utilizadas, parte por sua facilidade de implementação, parte por sua simplicidade.

#### 3.1.2.1 O Perceptron

O primeiro modelo matemático de um neurônio foi proposto por McCulloch e Pitts em 1943. O neurônio de McCulloch e Pitts era um dispositivo de entradas e saídas binárias, com um limiar fixo de ativação. Este modelo está representado na Figura 3.3.

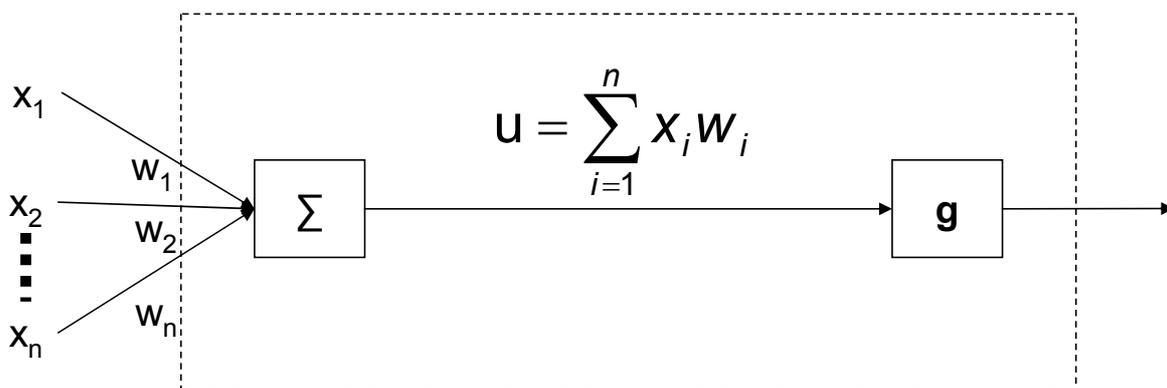


Figura 3.3 – Neurônio de McCulloch e Pitts.

Onde:

- $x_1, x_2, \dots, x_n$  são as entradas;
- $w_1, w_2, \dots, w_n$  são os pesos associados às  $n$  estradas do neurônio;
- $u$  é a função de entrada;
- $g$  é a função de ativação do neurônio.

Mas foi somente com o trabalho de Frank Rosenblatt (1958), que o conceito de treinamento em redes neurais artificiais foi introduzido. O modelo proposto por Rosenblatt (1958), conhecido como *perceptron*, era composto por uma estrutura de conexões diretas de três

camadas de neurônios de McCulloch e Pitts (1943). A primeira camada é um *buffer*, onde os dados de entrada são armazenados. Os elementos da primeira camada são conectados inteiramente ou arbitrariamente aos elementos da segunda camada, chamada de “camada intermediária”. Cada neurônio desta camada combina informações vindas da camada de entrada e representa uma possível característica. Os neurônios da camada intermediária são inteiramente conectados aos neurônios dispostos na saída, chamada de “camada de saída”. Os pesos da camada de entrada associados à camada intermediária são fixos. Essa é a razão pela qual o *perceptron* é comumente chamado de “rede neural de camada unitária”. A Figura 3.4 mostra um *perceptron* único contendo dois elementos na camada de entrada, um elemento na camada de saída e  $w_0$  é um valor de polarização.

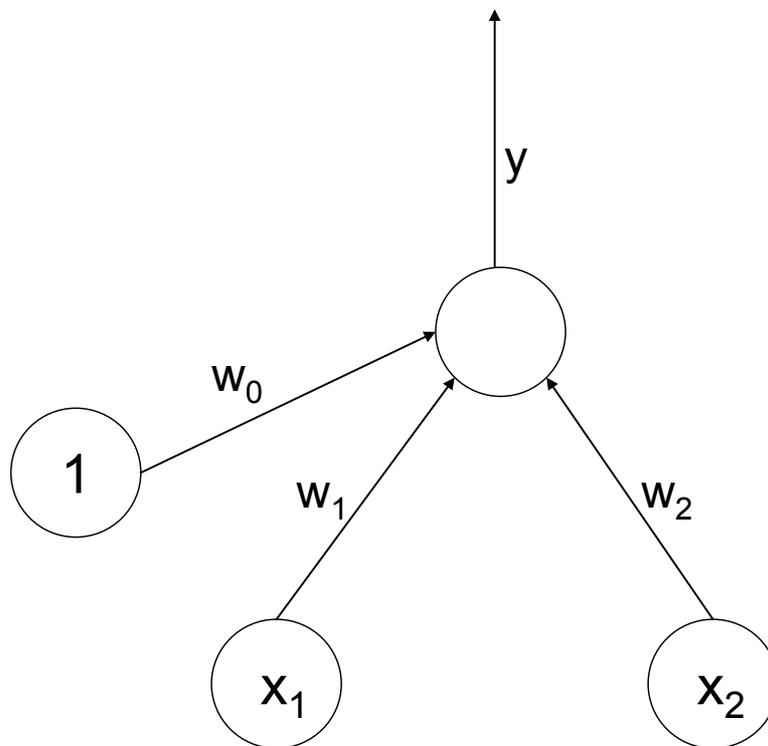


Figura 3.4 – Exemplo de *perceptron*.

As redes neurais de camada unitária resolvem apenas problemas linearmente separáveis. Assim, elas conseguem representar um hiperplano que divide o espaço euclidiano  $R^n$  em duas regiões. A Figura 3.5 a) mostra um exemplo deste tipo de problema para  $n=2$  e com valor de polarização  $w_0=0$ . Nesse caso o *perceptron* é capaz de classificar padrões que podem ser separados por uma reta passando pela origem. Se o *perceptron* fosse usado para implementar funções booleanas de duas variáveis, das 16 funções possíveis, somente o XOR (ou exclusivo) e o seu complemento não poderiam ser representadas corretamente. A Figura 3.5 b) mostra o

exemplo do XOR, os pontos claros significam que, para as entradas indicadas, a saída do perceptron deveria ser 0 e o ponto escuro que ela deveria ser 1. Observe que não é possível separar esses ponto apenas com uma reta. A solução para problemas não linearmente separáveis passa pelo uso de redes com uma ou mais camadas intermediárias, ou escondidas. Uma rede com apenas uma camada intermediária pode implementar qualquer função contínua. A utilização de duas ou mais camadas intermediárias permite a aproximação de qualquer função.

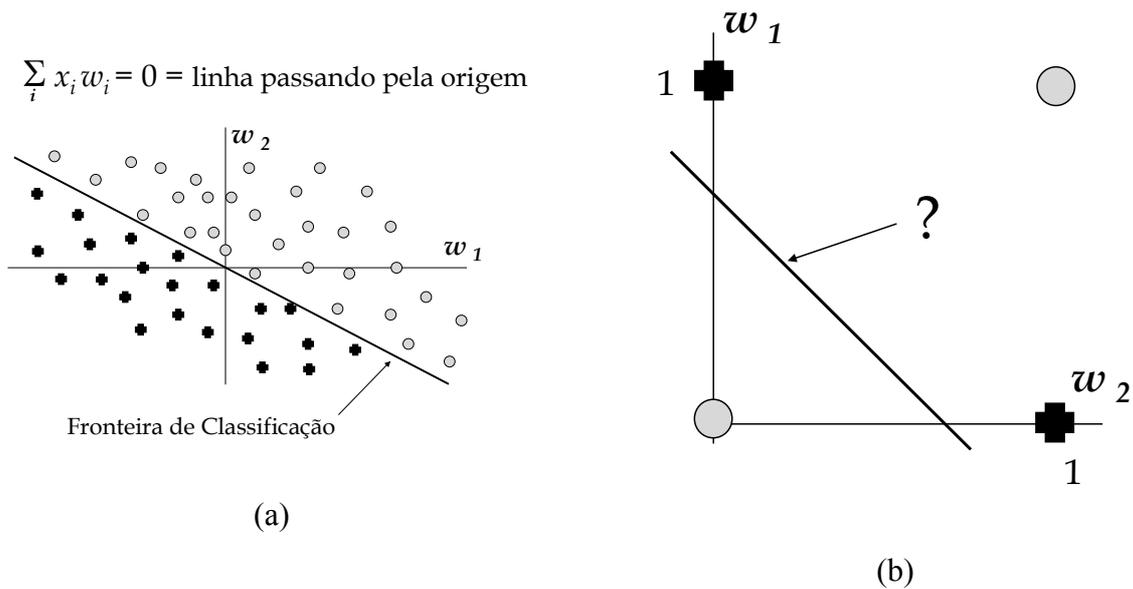


Figura 3.5 – (a) Coleções linearmente separáveis. (b) A função booleana XOR.

Para superar as limitações dos *perceptrons* de se trabalhar somente com problemas linearmente separáveis, as redes MLP foram introduzidas. Uma MLP consiste de uma camada de entrada, pelo menos uma camada intermediária e uma camada de saída. Os neurônios de cada camada são completamente conectados (exceto para algumas aplicações) aos neurônios da camada seguinte. A Figura 3.6 mostra um exemplo de uma MLP para resolver o problema XOR.

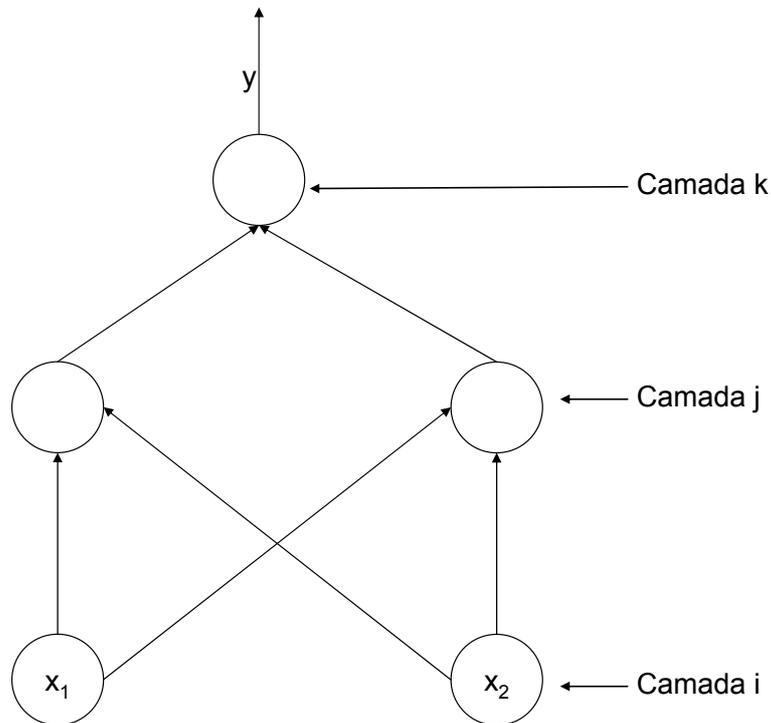


Figura 3.6 – Uma MLP para resolver o problema XOR.

Os neurônios de uma MLP têm valores de entrada e saída contínuos, funções de entrada somatórias e funções de ativação não lineares. Uma regra de gradiente descendente pode ser usada para encontrar conexões ótimas entre os pesos  $w_{ij}$  e  $w_{jk}$  que minimizem uma medida de erro  $E$ . Dessa forma, a variação dos pesos  $\Delta w_{ij}$  e  $\Delta w_{jk}$  na iteração  $(l+1)$  é calculada na direção inversa do gradiente do erro  $E$ , como mostra a equação (3.1).

$$\Delta w_{ij}(l+1) = -\lambda \left( \frac{\partial E}{\partial w_{ij}(t)} \right)$$

e

$$\Delta w_{jk}(l+1) = -\lambda \left( \frac{\partial E}{\partial w_{jk}(t)} \right)$$

(3.1)

onde  $\lambda$  é a taxa de aprendizagem.

Com a regra do gradiente descendente, depois de um certo número de iterações, o erro  $E$  tende a alcançar um valor mínimo, ou o “platô mais baixo”, se o erro  $E$  for representado como uma superfície no espaço vetorial dos pesos  $w$ . Um erro global para todos os exemplos de treinamento pode ser calculado pela equação (3.2).

$$E = \sum_{(p)} \sum_{(j)} \text{Err}_j^{(p)} \quad (3.2)$$

Onde o erro para cada exemplo (p),  $\text{Err}_j^{(p)}$ , pode ser calculado como um erro quadrático pela equação (3.3).

$$\text{Err}_j^{(p)} = \frac{(y_j^{(p)} - o_j^{(p)})^2}{2} \quad (3.3)$$

onde  $o_j^{(p)}$  é saída produzida pelo neurônio j, quando apresentado à rede o exemplo p.

A regra do gradiente descendente da equação (3.1) pode ser generalizada pela regra delta mostrada na equação (3.4).

$$\Delta w_{ij}(l+1) = \lambda \cdot \text{Err}_j \cdot g'(u_j) \cdot o_i \quad (3.4)$$

e

$$\Delta w_{jk}(l+1) = \lambda \cdot \text{Err}_k \cdot g'(u_k) \cdot o_j$$

onde  $g'$  é a derivada da função de ativação  $g$  em relação à entrada da rede  $u$ .

Cada iteração de treinamento pode ser definida como o processo de propagação de um ou vários exemplos através da rede e cálculo do erro  $E$  para cada um deles. O processo de treinamento consiste de dois passos:

1. Passo direto, onde as entradas são fornecidas e propagadas através das camadas intermediárias à camada de saída;
2. Passo *de* retroalimentação, onde um erro é calculado na saída e retropropagado para a modificação dos pesos da rede.

Este procedimento é ilustrado na Figura 3.7.

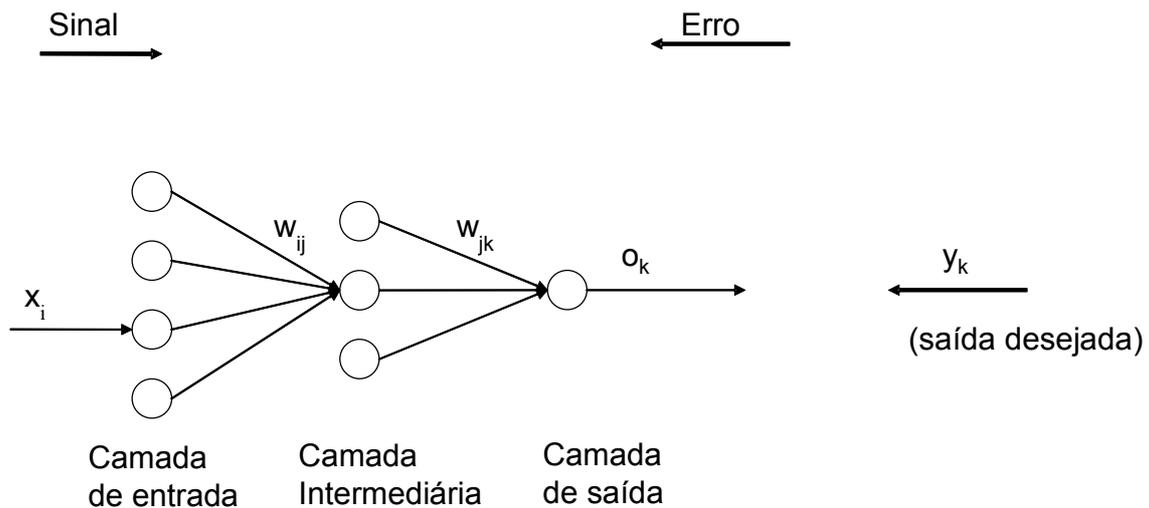


Figura 3.7 – Esquema de treinamento de uma rede MLP. O sinal é propagado de forma direta e o erro de forma retropropagada.

Durante o passo de retropropagação, um erro  $Err_i$  para um nó intermediário  $i$  é calculado multiplicando-se os erros  $Err_j$  de todos os neurônios  $j$  ao qual o neurônio  $i$  está conectado pelo peso  $w_{ij}$ . Este erro é usado para ajustar os pesos dos neurônios  $k$ , da camada anterior, conectados ao neurônio  $i$ . O processo de treinamento é repetido por várias iterações até que o erro global  $E$  seja suficientemente pequeno. Cada peso pode ter sua taxa de aprendizagem individual. Esse algoritmo é chamado de algoritmo de aprendizagem por retropropagação ou algoritmo *backpropagation*.

### 3.2 - REDES NEURAIIS DE KOHONEN

Redes neurais de aprendizagem competitiva constituem um dos tópicos mais interessantes dentro do estudo de redes neurais artificiais (Kohonen, 1988; Rumelhart e Zipser, 1985). Este tipo de rede pode detectar correlações e regularidades nos padrões de entrada e ajustar sua resposta futura de acordo com essas entradas. Neurônios de uma rede competitiva são capazes de aprender a reconhecer conjuntos de entradas semelhantes. Esse tipo de treinamento supervisionado, ao contrário do processo de retropropagação, ocorre sem o auxílio de um “professor”.

Imagine que seja apresentado a uma criança um conjunto de cubos azuis misturados a esferas vermelhas, e que seja pedido a esta criança que os separe, formando grupos que contenham objetos semelhantes. Mesmo que não haja um professor ao lado da criança, dizendo-lhe se o que ela está fazendo é certo ou errado, certamente esta criança não terá dificuldades para

separar os objetos em dois conjuntos, um contendo apenas os cubos azuis e ou outro contendo apenas as esferas vermelhas. É possível construir uma rede capaz de realizar este tipo de aprendizagem.

As redes de aprendizagem competitiva são redes simples, compostas por apenas duas camadas: a camada de entrada (não computacional), e a camada de saída, também chamada de camada de Kohonen. De uma maneira bem simplificada, a camada de Kohonen funciona segundo o critério de “o vencedor leva tudo”, ou seja, para um vetor de entrada, apenas um neurônio apresenta-se ativo, estando todos os outros inativos. Para a rede da Figura 3.8,

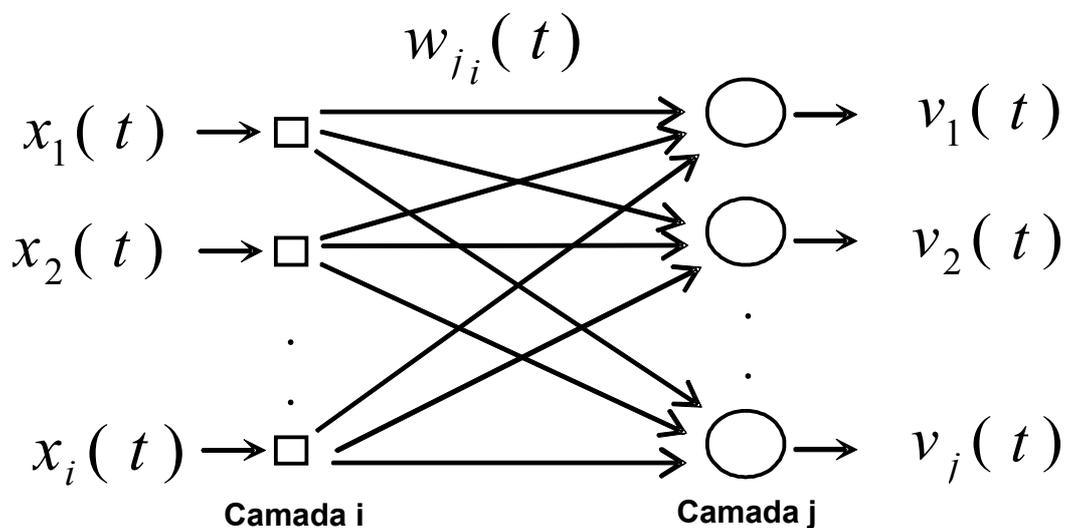


Figura 3.8 – Camada de Kohonen genérica.

onde, os valores de entrada  $x_i(t)$  formam um vetor  $\bar{x}(t)$  de entrada e os pesos  $w_{i,j}(t)$  formam, para cada neurônio  $j$ , um vetor de pesos  $\bar{w}_j(t)$ .

O neurônio declarado “vencedor” é aquele cujos pesos apresentam a maior semelhança (ou maior campo local induzido) com as entradas. A semelhança é medida por uma norma, geralmente utiliza-se a norma L2 euclidiana da diferença. Assim, o campo local induzido de cada neurônio é calculado para cada instante de tempo pela equação (3.5):

$$v_j = w_{1,1}x_1 + w_{1,2}x_2 + \dots + w_{j,i}x_i \quad (3.5)$$

O neurônio que apresentar o maior campo local induzido terá sua saída em nível lógico 1. Todas as outras saídas estarão no nível lógico 0.

O processo de treinamento que permite a auto-organização da rede é baseado em um procedimento muito simples. Sabe-se que os neurônios devem separar os vetores de entrada em grupos de vetores semelhantes. Para atingir este objetivo, os pesos são ajustados de maneira que entradas semelhantes sempre ativem o mesmo neurônio. Como se trata de auto-organização, é difícil (e mesmo desnecessário) saber, antes de o treinamento acontecer, qual o padrão que será associado a cada um dos neurônios. A única exigência é que padrões diferentes ativem neurônios diferentes. O campo local induzido de cada neurônio é usado como medida de similaridade no processo de aprendizagem. Quando uma nova entrada é apresentada à rede, o neurônio de maior campo local induzido recebe o *status* de vencedor. Isto quer dizer que a nova entrada apresenta maior similaridade com o padrão identificado por este neurônio do que com os padrões identificados pelos demais neurônios. Assim, o vencedor tem suas sinapses atualizadas de maneira a torná-lo ainda mais similar à entrada que o ativou.

O neurônio vencedor está localizado no centro de uma vizinhança com uma topologia formada por neurônios “colaboradores”. Assim, a interação lateral entre neurônios vizinhos é realizada de forma que um neurônio tenha uma conexão de excitação forte com ele mesmo e algumas poucas conexões de excitação com seus vizinhos, dentro de uma certa vizinhança.

Seja  $h_{i,j}$  a vizinhança com centro no neurônio  $i$  e compreendendo um conjunto de neurônios colaboradores, dos quais um neurônio típico seria denotado por  $j$ . Seja  $d_{i,j}$  a distância lateral entre o neurônio vencedor  $i$  e um neurônio ativado  $j$ . Então, pode-se assumir que a vizinhança  $h_{i,j}$  seja uma função unimodal da distância lateral  $d_{i,j}$  e satisfaça duas condições:

- A vizinhança  $h_{i,j}$  é simétrica em relação ao ponto de máximo definido por  $d_{i,j}=0$ ;
- O tamanho da vizinhança  $h_{i,j}$  diminui, por exemplo, monotonicamente com o aumento da distância  $d_{i,j}$ , tendendo a zero quando  $d_{i,j} \rightarrow \infty$ .

Uma escolha típica de  $h_{i,j}$  que satisfaz essas duas condições é a função Gaussiana:

$$h_{i,j}(\bar{x}) = \exp\left(-\frac{d_{i,j}^2}{2\sigma^2(t)}\right) \quad (3.6)$$

onde  $\sigma(t)$  é um parâmetro que controla o tamanho da vizinhança  $h_{i,j}$  e diminui com o tempo.

O processo de treinamento não supervisionado proposto por Teuvo Kohonen (Kohonen, 1988) é descrito a seguir:

1) Inicializa-se com números aleatórios pequenos os vetores de pesos  $\bar{w}_j$  do mapa de saída;

2) Apresenta-se um vetor de entrada  $\bar{x}$  à rede no instante  $t$ ;

3) Calcula-se a distância  $d_j$  entre  $\bar{x}$  e os vetores de pesos  $\bar{w}_j(t)$  de cada neurônio  $j$ . No espaço Euclidiano essa distância é calculada como:

$$d_{i,j} = \text{sqrt}(\sum(x_i - w_{ij})^2) \quad (3.7)$$

4) O neurônio  $j$  com maior campo local induzido é declarado vencedor e torna-se o centro de uma vizinhança  $H$  de tamanho  $h_{i,j}$ ;

5) Incrementam-se os vetores de pesos dentro da vizinhança do neurônio vencedor:

$$\bar{w}_j(t+1) = \bar{w}_j(t) + \alpha(t)[\bar{x} - \bar{w}_j(t)], \text{ se } j \in H \quad (3.8)$$

Nesse processo, depois que todos os padrões de entrada são apresentados, um neurônio é declarado vencedor e as conexões dentro de sua vizinhança  $H$  são alteradas, enquanto os pesos das conexões fora da vizinhança permanecem inalterados. Os passos de 2) a 5) são repetidos com todos os padrões de treinamento. Os valores de  $\alpha$  e do tamanho da vizinhança  $h_{i,j}$  diminuem com o tempo. Todo o processo de treinamento é repetido até a convergência ou ao atingir-se um número máximo de iterações.

### 3.3 - CONSIDERAÇÕES FINAIS

No próximo capítulo, serão apresentados os algoritmos de codificação objeto de estudo desta tese e nela desenvolvidos. Os conceitos apresentados neste capítulo foram valiosos no estudo de técnicas adaptativas e combinações híbridas para melhora de desempenho, seja na fidelidade da forma de onda do sinal reconstruído, seja na minimização da taxa de bits por símbolo destes algoritmos.

## 4 - ALGORITMOS PROPOSTOS

Neste capítulo dar-se-á ênfase aos algoritmos propostos e às principais contribuições deste trabalho. Será descrita a técnica desenvolvida para a compressão de sinais eletrofisiológicos como os de eletromiografia e eletrocardiografia. Serão abordados os detalhes dos diversos módulos que constituem o codificador e o decodificador. Para esses algoritmos ainda foi estudado um pré-processamento para minimização de ruído (ou *de-noising*). A técnica escolhida neste caso foi o *de-noising* por transformada *wavelet* invariante ao deslocamento, SIDWT (do inglês, *Shift Invariant Discrete Wavelet Transform*), conforme descrito na seção 4.1.1.

Também será apresentada a descrição do codificador para voz de banda larga, incluindo as modificações necessárias no padrão AC-3 e a introdução da inteligência artificial no processo de alocação dinâmica de bits.

### 4.1 - ALGORITMO PARA CODIFICAÇÃO DE SINAIS ELETROFISIOLÓGICOS

#### 4.1.1 - Pré-processamento do sinal eletrofisiológico: *de-noising* invariante ao deslocamento por transformada *wavelet* discreta.

A presença de ruído é uma característica inerente ao sinal eletromiográfico, resultado do processo de aquisição desses sinais. Entre os tipos de ruídos mais comuns no sinal eletromiográfico estão (de Luca, 1979):

- Ruído provocado por componentes eletrônicos – é um ruído de banda larga, do tipo rosa ou branco;
- Ruídos provenientes do ambiente ou radiações eletromagnéticas – possuem origem em fontes eletromagnéticas nas proximidades do local de coleta do sinal eletromiográfico. O ruído dessa natureza mais comum é o de 60Hz, provocado pela rede de alimentação;
- Ruído de artefato de movimento – é gerado pela movimentação relativa do eletrodo na pele e/ou nas conexões do eletrodo com o eletromiógrafo. Possui largura de banda de até 20Hz.

Para melhorar o desempenho do algoritmo de compressão decidimos aplicar um algoritmo de minimização de ruído aos sinais eletrofisiológicos. A técnica escolhida neste caso foi o *de-noising* invariante ao deslocamento por transformada *wavelet* (SIDWT), conforme mostra o diagrama de blocos na Figura 4.1.

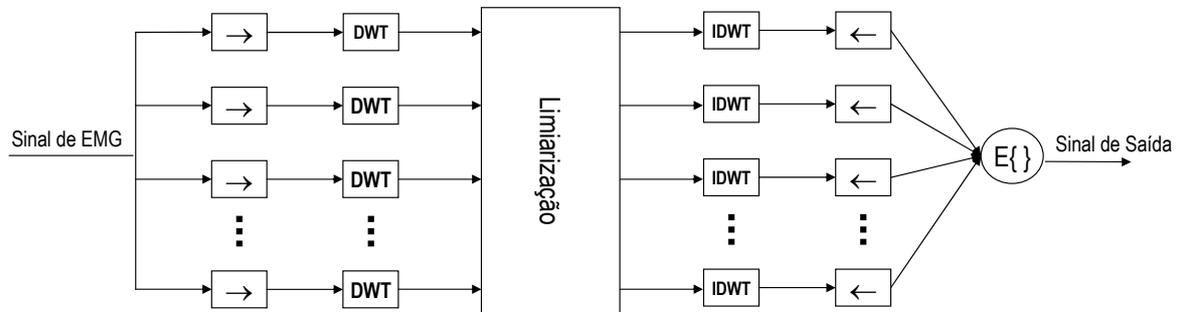


Figura 4.1 – Diagrama de blocos para o *de-noising* por SIDWT.

O procedimento representado por esse diagrama de blocos pode ser resumido como: 1) calcula-se M versões do sinal de EMG aplicando-se M deslocamentos circulares ( $\rightarrow$ ) ao bloco do sinal original; 2) calcula-se a DWT para cada sinal deslocado circularmente; 3) faz-se igual a zero os coeficientes transformados de menor amplitude aplicando-se um limiar de amplitude fixo; 4) calcula-se a DWT inversa para os coeficientes restantes; 5) aplicam-se os deslocamentos circulares inversos ( $\leftarrow$ ) e calcula-se a média dos sinais resultantes.

O *de-noising* usando a SIDWT é mais eficiente do que o método tradicional (sem os deslocamentos) porque uma versão deslocada do sinal original pode resultar em uma transformação para o domínio da DWT com a energia compactada em um número menor de coeficientes do que a DWT do sinal original. Uma vez que a compactação da energia é mais eficiente, o *de-noising* tem seu desempenho melhorado. O método ótimo seria procurar pelo deslocamento que resulta em melhor desempenho. Entretanto, nem sempre existe um único deslocamento que resulta na melhor compactação de energia no domínio da DWT. Portanto, uma estratégia subótima é tomar a média dos resultados de todos os deslocamentos.

Uma metodologia para encontrar o limiar de amplitude ótimo para o descarte de coeficientes transformados foi desenvolvida. Essa metodologia consiste em adicionar uma quantidade fixa e controlada (tal que a relação sinal-ruído fosse de 10 dB) de ruído gaussiano branco aditivo aos sinais originais, para cada sinal com ruído processado pelo algoritmo de *de-noising*, variando-se o limiar de amplitude. Para cada resultado obtido pela minimização de ruído, foi calculado o erro médio quadrático normalizado, MSE (do inglês, Mean Squared Error), entre

o resultado e o sinal de EMG simulado original. O limiar de amplitude que resulta no menor MSE é considerado ótimo em termos de ganho de codificação.

Uma discussão mais detalhada do assunto pode ser encontrada em (Burrus *et al*, 1989; e Mallat 2001).

#### 4.1.2 - Descrição do algoritmo de codificação.

O algoritmo de codificação de sinais eletrofisiológicos proposto neste trabalho pode ser representado pelo diagrama de blocos da Figura 4.2.

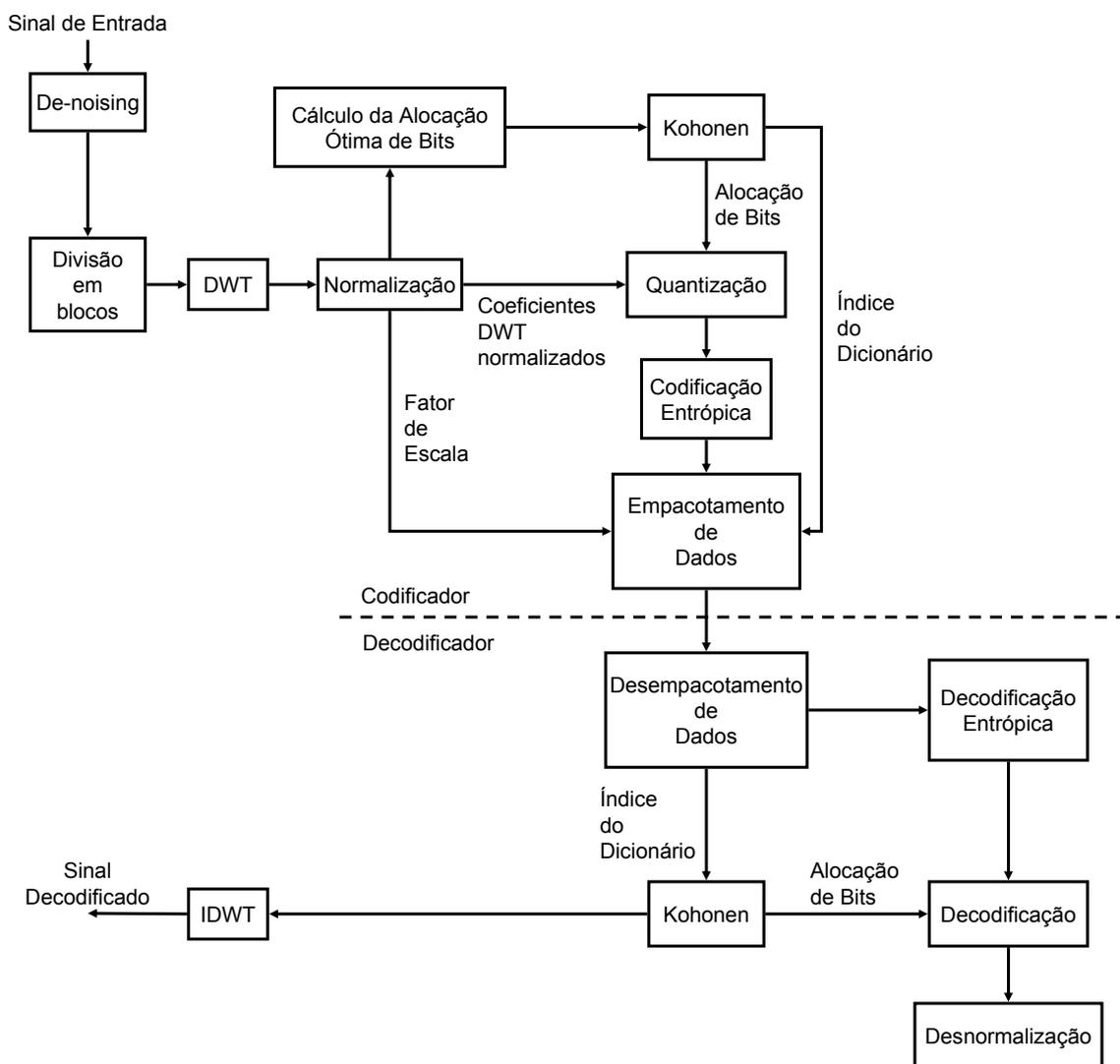


Figura 4.2 – Diagrama de blocos do codificador e decodificador.

Conforme foi descrito na seção anterior, os sinais eletrofisiológicos podem ser processados com um algoritmo de *de-noising*. Depois da minimização de ruído o sinal de EMG é dividido

em blocos de tamanho fixo (2048 amostras), e cada bloco é então transformado para o domínio *wavelet*. Em seguida, um algoritmo de normalização calcula um fator de escala para cada bloco transformado. Esse fator de escala deve ser transmitido integralmente ao decodificador. Após a normalização dos coeficientes transformados é calculada a alocação de bits ótima para cada coeficiente. Essa alocação de bits é a entrada de um dicionário, formado pela rede de Kohonen, que faz uma aproximação. O bloco transformado é quantizado usando a quantidade de bits determinada pela camada de Kohonen. Uma vez quantizados, os coeficientes são codificados por um codificador de entropia, formado por um codificador de *Huffman* e uma codificação *Run-length*. Esses dados são empacotados com a informação lateral necessária à decodificação e estão prontos para a transmissão ou armazenamento.

#### 4.1.2.1 Escolha da transformada *wavelet*

A função de base *wavelet* utilizada neste algoritmo de codificação é da família de *wavelets* Daubechies. A Figura 4.3 mostra algumas das funções de base *wavelet* da família Daubechies. A escolhida nesta pesquisa foi a Daubechies-4. Essa escolha foi feita baseada nos resultados apresentados em (Berger *et al*, 2003), segundo os quais, o uso da *wavelet* Daubechies-4 resulta no melhor desempenho em termos da decorrelação espacial do sinal de EMG.

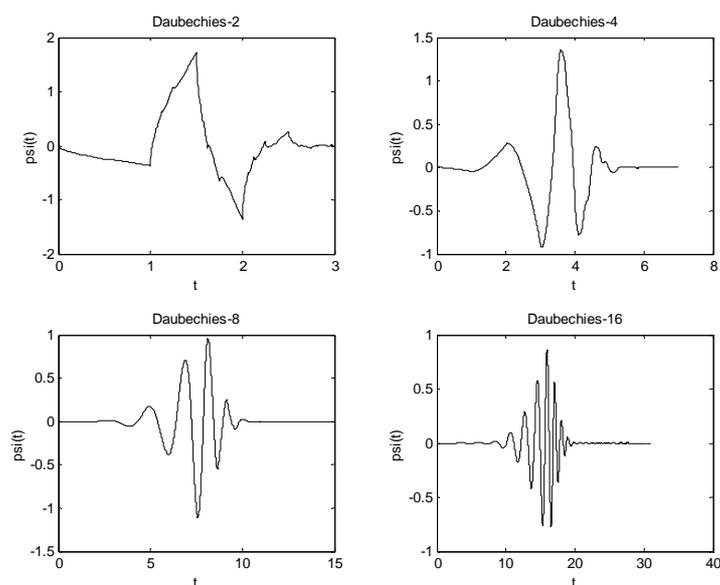


Figura 4.3 – Exemplos de funções de base *wavelet* da família Daubechies.

O trabalho de Berger *et al*, 2003, também apresenta um estudo sobre a escolha da escala de resolução da transformada *wavelet*. Seus resultados mostraram que o valor ótimo para este parâmetro é de pelo menos oito níveis de resolução, uma vez que o ganho de desempenho com escalas de resolução maiores não justifica o aumento do esforço computacional. Assim, nesta pesquisa, adotou-se também, oito níveis de resolução na transformada *wavelet*.

#### 4.1.2.2 Normalização dos coeficientes transformados

No diagrama de blocos da Figura 4.2, cada bloco transformado é submetido a um processo de normalização, onde os seus coeficiente transformados são multiplicados por um fator de escala  $\lambda_i$  calculado da seguinte forma:

$$\lambda_i = \frac{2^Q}{\max\{X_i[k]\}} \quad (4.1)$$

onde  $i$  é o índice do bloco que está sendo processado,  $0 \leq Q \leq 12$  é um fator de qualidade e  $X_i[k]$  são os coeficientes transformados do  $i$ -ésimo bloco.

O processo de normalização juntamente com a DWT provoca o surgimento de muitos coeficientes nulos na fase de quantização. Ou seja, esta combinação resulta no equivalente a uma limiarização que se adapta à estatística do sinal, proporcionando melhor relação sinal-ruído que os codificadores por transformada tradicionais, que estabelecem um limiar fixo para cada bloco.

Outra vantagem desse procedimento de normalização é que ele fornece um método para controlar a qualidade do sinal reconstruído. Pode-se observar que quanto menor o fator de qualidade  $Q$ , maior será a quantidade de coeficientes nulos ou quantizados com poucos bits, resultantes da combinação: DWT + Normalização + Quantização. Isto é, quanto menor for o valor de  $2^Q$  em relação ao valor  $\max\{X[k]\}$ , na equação (4.1), menor será o fator de normalização  $\lambda_i$ , e, portanto menor será a quantidade de bits alocada para os coeficientes transformados. Essa análise pode ser melhor compreendida na seção a seguir, que descreve a alocação de bits e o processo de quantização dos coeficientes transformados.

#### 4.1.2.3 Alocação de bits e quantização dos coeficientes transformados

O processo de quantização dos coeficientes transformados é ilustrado na Figura 4.4. Depois da normalização, cada bloco de 2048 coeficientes transformados é dividido em 16 grupos de 128 coeficientes. Para cada grupo é calculada a quantidade de bits que o maior coeficiente transformado necessita (calculando o logaritmo na base dois do módulo do maior coeficiente), ou seja, um vetor de alocação de bits provisório de 16 amostras. Este vetor é aproximado pela camada de Kohonen, que retorna o vetor de alocação de bits definitivo, uma espécie de versão quantizada vetorialmente do vetor provisório.

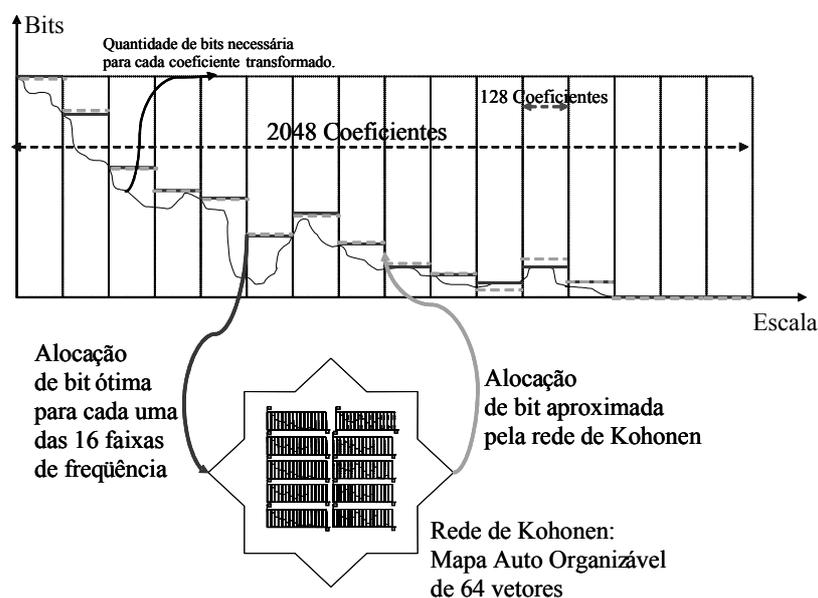


Figura 4.4 – Ilustração do processo de quantização dos coeficientes transformados.

A camada de Kohonen utilizada possui uma topologia retangular de 8x8 neurônios e representa um dicionário com 64 vetores, portanto são necessários seis bits por bloco para transmitir ao decodificador o índice do vetor de alocação de bits definitivo. Essa rede foi treinada com um banco de sinais de treinamento de tamanho correspondente a 40% do tamanho do conjunto de dados de validação. O algoritmo de treinamento utilizado foi aquele descrito na seção 3.2 com função de vizinhança gaussiana. O número máximo de iterações do treinamento foi de 1500 épocas. A taxa de aprendizado foi de 0.9 para as primeiras 1000 épocas e de 0.01 para o restante.

Uma vez determinado o vetor de alocação de bits definitivo, o bloco transformado é quantizado e codificado usando codificação por entropia. Por fim é realizado o empacotamento de dados, onde é adicionada ao bloco codificado a informação lateral

necessária à decodificação. Esta informação lateral é de 22 bits por bloco, onde 6 bits são necessários para transmitir a classe da alocação de bits e 16 bits para o fator de normalização.

A decodificação é realizada seguindo o caminho inverso, como mostra o diagrama de blocos da Figura 4.2.

#### **4.2 - ALGORITMO PARA CODIFICAÇÃO DE SINAIS DE VOZ DE BANDA LARGA**

A Figura 4.6 mostra o diagrama de blocos do esquema de codificação investigado: uma versão simplificada e adaptada do codificador de áudio AC-3 (Todd *et al*, 1994; G. Davidson *et al*, 2001; ATSC *Revision A*” Doc. A/52<sup>a</sup>, 2001). A codificação é feita no domínio da frequência, usando uma MDCT de 2M pontos com 50% de superposição (Primcen e Bradley, 1986; Malvar, 1990; Koilpillai e Vaidyanathan, 1992; Painter e Spanias, 2000). Isto é, a transformada é aplicada a blocos de 2M amostras PCM (do inglês, *Pulse Code Modulation*), com 50% de superposição entre blocos — veja a Figura 4.7. Contudo, somente M coeficientes MDCTs são gerados para cada bloco de entrada de 2M amostras; ou seja, essa transformação é equivalente a um banco de M filtros cujas saídas são criticamente amostradas. As funções de base MDCTs estendem-se, portanto, por dois blocos no tempo, levando à eliminação virtual dos efeitos de bloco que degradam o sinal reconstruído quando a transformada usada é sem superposição.

No codificador AC-3, a transformada do sinal de áudio para o domínio da frequência é realizada através da MDCT que toma blocos no domínio do tempo de  $L=512$  amostras com superposição de  $M=256$  amostras. Cada bloco de áudio  $x(n)$  é multiplicado por uma janela  $w(n)$  do tipo Kaiser-Bessel derivada, mostrada na Figura 4.5.

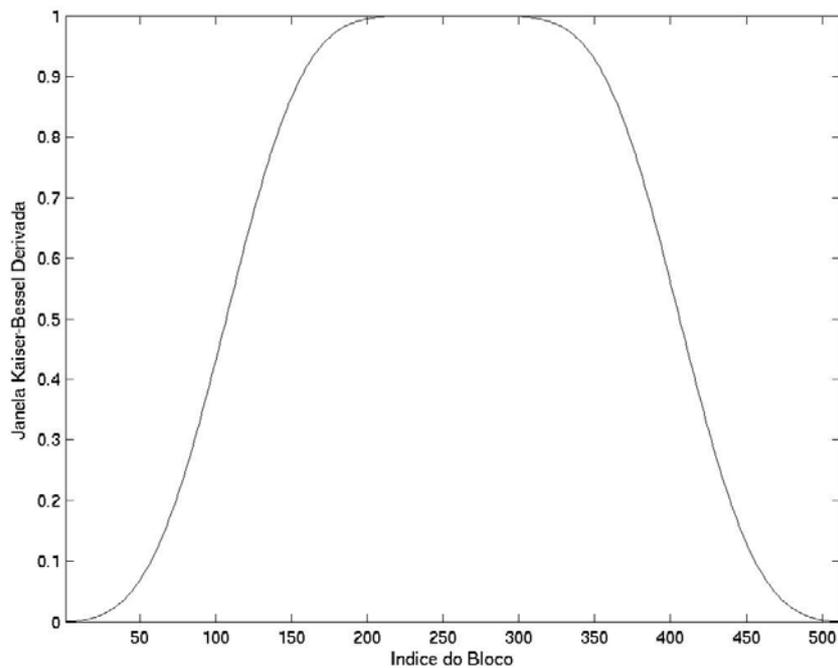


Figura 4.5 – Janela aplicada ao sinal de voz de banda larga.

Cada um dos coeficientes MDCTs é representado na notação binária de ponto flutuante como um expoente e uma mantissa. O conjunto de expoentes é codificado como uma representação grosseira do espectro do sinal que é referida como envoltória espectral. Com base nessa envoltória espectral, é obtida, pela aplicação de um modelo perceptivo, uma curva de limiar de mascaramento, que indica qual o nível máximo que o ruído de codificação pode ter para que ele não seja percebido pelo ouvinte. Essa curva é usada pelo algoritmo de alocação de bits para determinar quantos bits serão usados para codificar cada uma das mantissas.

O processo de decodificação é basicamente o inverso do processo de codificação. Uma vez que os expoentes tenham sido decodificados, eles são usados para reproduzir o modelo perceptivo e a alocação de bits usados pelo codificador. Os bits que representam as mantissas são, então, decodificados. Os coeficientes MDCTs reconstruídos são transformados de volta para o domínio do tempo para produzir as amostras PCMs decodificadas — veja ilustração na Figura 4.6 e a equação (2.19).

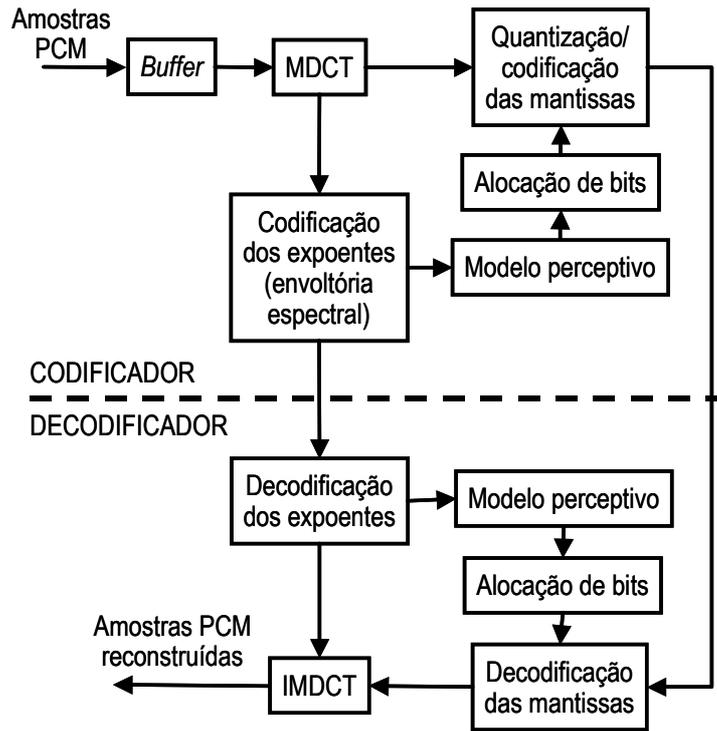


Figura 4.6 - Diagrama de blocos do codificador e decodificador.

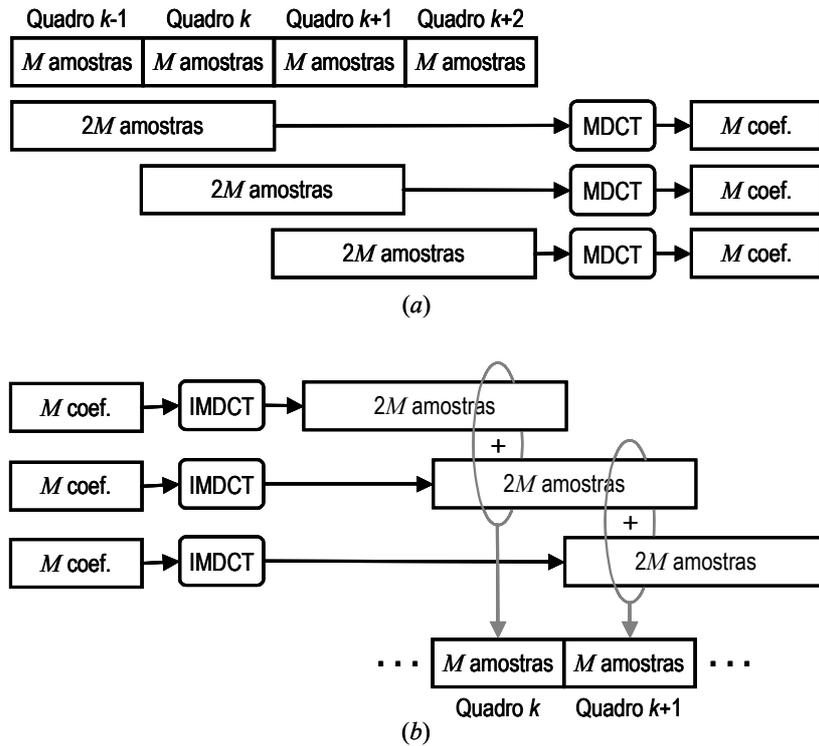


Figura 4.7 – Ilustração da aplicação da MDCT: (a) MDCT direta e (b) MDCT inversa (IMDCT).

O algoritmo de alocação de bits usado nesse trabalho é do tipo *backward*, ou seja, ele não requer a transmissão de qualquer informação lateral entre o codificador e o decodificador. Ele

é uma versão simplificada do algoritmo *backward/forward* híbrido usado pelo codificador AC-3 (Todd *et al.*, 1994; G. Davidson *et al.*, 2001; ATSC *Revision A Doc. A/52<sup>a</sup>*, 2001) Nas seções seguintes são mostrados os detalhes desse algoritmo.

#### 4.2.1 - Codificação da envoltória espectral

No padrão AC-3, desenvolvido para aplicações de áudio, os expoentes dos coeficientes transformados são todos codificados. No algoritmo proposto neste trabalho, voltado para a codificação de voz de banda larga, os coeficientes MDCTs de índice 224 até 255 correspondem a frequências fora da banda do sinal (50-7000 Hz) e, por isso, seus expoentes e mantissas não são transmitidos: o decodificador os faz iguais a zero. Os expoentes dos 224 coeficientes restantes, de índice 0 até 223, são divididos em grupos. De cada grupo, apenas o expoente de maior valor é codificado e enviado, e os demais expoentes do grupo são feitos iguais a esse de maior valor. A mantissa que tem seu expoente alterado é escalonada para compensar a mudança feita no expoente. Apenas um expoente é enviado integralmente, codificado em uma palavra de 5 bits. Os demais expoentes são codificados diferencialmente, com deltas de 0,  $\pm 1$  ou  $\pm 2$ . Os deltas são divididos em conjuntos de 3 deltas e cada conjunto é codificado em uma palavra de 7 bits. Foram experimentados dois tipos de agrupamento dos expoentes:

- i) Agrupamento Uniforme (AU): os 224 expoentes são divididos em 112 grupos, cada um contendo um par de expoentes. Portanto, são gastos  $5 + (111 \div 3) \times 7 = 264$  bits para representar os expoentes de um bloco de coeficientes MDCTs.
- ii) Agrupamento Não-Uniforme (ANU): os 224 expoentes são divididos em 67 grupos, que podem ter 2, 3, 4 ou 5 elementos. Grupos correspondentes a faixas de frequências baixas contêm 2 elementos e grupos que correspondem a faixas de frequências altas podem conter até 5 elementos. Essa divisão não uniforme é inspirada nas bandas críticas da audição humana, que também têm largura crescente com a frequência. Nesse caso, são gastos  $5 + (66 \div 3) \times 7 = 159$  bits para representar os expoentes de um bloco de coeficientes MDCTs.

Uma forma de reduzir o dispêndio de bits com a codificação dos expoentes é a reutilização dos expoentes codificados de um bloco de coeficientes MDCTs em blocos subseqüentes. No codificador AC-3, essa reutilização pode-se estender a até cinco blocos subseqüentes. Nas

simulações realizadas nesse trabalho, apenas a reutilização em um bloco subsequente foi considerada. Isso porque essa reutilização torna o codificador muito sensível à perda de informações a exemplo da perda de pacotes em uma transmissão baseada em pacotes.

Com uma reutilização, o dispêndio médio de bits para representar os expoentes dos coeficientes MDCTs é de 132 bits por bloco (ou por 15 ms), caso seja implementado o agrupamento uniforme. E é de 79,5 bits, se for usado o agrupamento não-uniforme.

#### 4.2.2 - Alocação de bits

Uma vez fixada a taxa de bits  $R_b$  (bps) a ser despendida na codificação do sinal, o número de bits  $B_Q$  que pode ser gasto com cada quadro é dado por  $B_Q = R_b T_Q$ , onde  $T_Q = 15$  ms é a duração de um quadro. No codificador simulado,  $B_Q = B_E + B_M$ , onde  $B_E$  e  $B_M$  são, respectivamente, as quantidades de bits gastas na codificação dos expoentes e das mantissas dos coeficientes MDCTs. A codificação dos expoentes e o valor de  $B_E$  foram assuntos da seção 4.2.1. Nesta seção discute-se o processo pelo qual os  $B_M$  bits restantes são distribuídos entre as mantissas dos coeficientes MDCTs de modo a minimizar a distorção audível presente no sinal reconstruído. Essa distribuição é dinâmica, enquanto que a distribuição dos  $B_E$  bits entre os expoentes é estática.

O objetivo é determinar  $b(k)$ ,  $k = 0, 1, \dots, M-1$ , sujeito à seguinte restrição:

$$\sum_{k=0}^{M-1} b(k) = B_E \quad (4.2)$$

onde  $b(k)$  é o número de bits alocado para representar a  $k$ -ésima mantissa e que define o número de níveis do quantizador a ser usado. O critério usado é o da distorção perceptiva. Bits são distribuídos buscando fazer com que a envoltória do espectro de potência do ruído de quantização fique sempre abaixo do limiar de mascaramento. Todos os componentes de ruído com nível inferior a esse limiar são encobertos pelo próprio sinal de voz (ou de áudio) e não são percebidos pela audição humana (Painter e Spanias, 2000). O limiar de mascaramento é calculado de acordo com um modelo perceptivo (ou psicacústico). Nos parágrafos seguintes, descreve-se de forma sucinta o modelo psicacústico usado nesse trabalho: ele é uma versão simplificada do modelo psicacústico usado pelo AC-3 (ATSC *Revision A Doc. A/52<sup>a</sup>*, 2001).

O cálculo do limiar de mascaramento é realizado em três etapas (Davidson et al., 1994; ATSC *Revision A Doc. A/52<sup>a</sup>*, 2001). Primeiro, os expoentes codificados são transformados em uma densidade espectral de potência (DEP) logarítmica da seguinte forma:

$$\text{dep}(k) = 3072 - \exp(k) \times 128 \quad (4.3)$$

Na escala logarítmica resultante, 128 unidades correspondem a 6 dB. Então, as amostras da DEP são agrupadas em bandas cuja largura corresponde, aproximadamente, à metade da largura das bandas críticas da audição humana. No codificador simulado nesse trabalho, o espectro na faixa entre 0 e 7000 Hz ( $0 \leq k \leq 223$ ) é dividido em 37 bandas, contendo o número de amostras indicado na Tabela 4.1, com o espaçamento entre duas amostras adjacentes de 31,25 Hz. As amostras contidas em uma única banda são somadas (em escala linear), resultando, então, uma DEP integrada com apenas 37 amostras. O propósito dessa integração é reduzir o esforço computacional requerido para realizar a terceira etapa.

Finalmente, o limiar de mascaramento versus frequência é obtido convoluindo a DEP integrada com um protótipo de função de espalhamento — isto é, um modelo da curva de mascaramento da audição humana. O modelo usado nas simulações é o modelo proposto para o AC-3, com parâmetros estáticos (ATSC *Revision A Doc. A/52<sup>a</sup>*, 2001). O resultado desse cálculo é uma curva que indica, para cada banda, o nível de mascaramento estimado. Essa curva é comparada com limiar absoluto da audição humana, e o maior entre os dois é mantido.

Tabela 4.1 – Número de amostras nas 37 bandas em que a faixa entre 0 e 7 kHz é dividida.

Banda	1-9	10-16	17-21	22-24	25-26	27-28	29-30	31-32	33	34	35	36	37
Quantidade de coeficientes	2	3	4	5	6	7	9	11	14	15	17	18	20

A curva estimada de mascaramento é subtraída da densidade espectral de potência para determinar a razão sinal ruído, SNR (do inglês, *Signal to Noise Ratio*) desejada para cada coeficiente MDCT. A cada bit adicional alocado a um quantizador uniforme, a SNR aumenta por 6 dB aproximadamente. Assim, para cada coeficiente MDCT, pode-se mapear a SNR desejada no número de bits  $b(k)$  que deve ser alocado para o quantizador. O critério dos 6 dB por bit é impreciso para um número pequeno de bits alocados. O mapa utilizado nas

simulações é o usado pelo AC-3, que considera um ganho de SNR menor que 6 dB por bit para  $b(k) \leq 4$ . O valor máximo para  $b(k)$  é 16.

Uma vez que  $b(k)$ ,  $k = 0, 1, \dots, M-1$ , esteja determinado, as mantissas são quantizadas usando quantizadores uniformes. Os quantizadores com até 15 níveis são simétricos e os que possuem mais de 15 níveis são assimétricos.

Várias mantissas não recebem bits para a sua codificação. Isso porque o componente espectral correspondente tem nível abaixo do limiar de mascaramento e, portanto, não é audível — logo, ele não precisa ser transmitido. Mas mantissas não recebem bits também porque os bits disponíveis acabaram, tendo sido gastos com mantissas mais relevantes. Seja qual for o caso, a providência natural a ser tomada pelo decodificador seria substituir o valor real dessas mantissas por zero. Contudo, em (ATSC *Revision A Doc. A/52<sup>a</sup>*, 2001) é recomendado que os valores dessas mantissas sejam substituídos por valores aleatórios. Testes subjetivos mostraram que essa estratégia realmente melhora a qualidade subjetiva do sinal reconstruído: ele soa menos abafado, apresentando maior riqueza espectral de alta frequência. Contudo, percebe-se também um ruído causado por essa estratégia. Assim, a amplitude máxima desses números aleatórios deve ser escolhida com cuidado. Nas simulações realizadas, foram utilizados números aleatórios com distribuição uniforme entre  $-0,35$  e  $+0,35$ .

A soma do número de bits alocados para cada mantissa é comparada com a quantidade de bits disponíveis para cada quadro. Deste ponto em diante o sistema começa uma rotina de ajuste da alocação de bits.

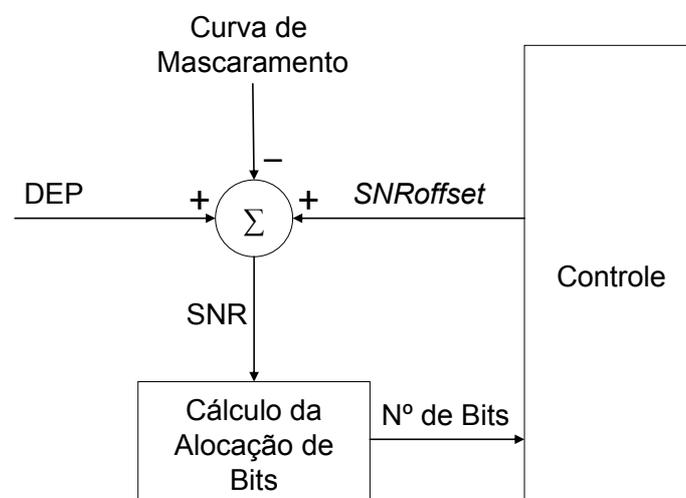


Figura 4.8 – Ajuste da alocação de bits.

No ajuste da alocação de bits, a rotina de controle identifica se a quantidade de bits calculada foi menor do que a quantidade de bits disponíveis e caso isso aconteça, um parâmetro chamado *SNROffset* é somado da SNR e recomeça-se uma nova alocação de bits. A forma como este parâmetro é adaptado não é completamente especificada no padrão original, e a busca exaustiva demanda uma grande quantidade de processamento. A sugestão do documento A/52 é que esse parâmetro seja ajustado em incrementos de 3 dB, começando de um valor inicial padrão. Quando a quantidade de bits calculada excede a quantidade de bits disponíveis, o valor de *SNROffset* imediatamente anterior é tomado e inicia-se um ajuste fino. No ajuste fino o valor de *SNROffset* é incrementado de 3/8 dB e o mesmo processo anterior se repete até que a quantidade bits disponíveis seja excedida novamente.

A solução que propomos nesse trabalho para diminuir a quantidade de processamento na busca do parâmetro *SNROffset* é utilizar uma rede neural artificial do tipo MLP, no bloco de controle da Figura 4.8, para encontrar um valor inicial mais próximo possível do valor ótimo. Esta rede neural substitui o ajuste em incrementos de 3 db. A rede MLP sugerida tem uma camada de entrada com três neurônios, uma camada intermediária com 16 neurônios e um neurônio na camada de saída. As camadas de entrada e saída têm função de ativação linear, e a camada intermediária tem função de ativação do tipo tangente hiperbólica. Os parâmetros de entrada são: a variância da DEP, a variância da curva estimada de mascaramento e a variância da quantidade de bits alocada ao bloco anterior. A variância da DEP apresenta para a rede uma referência do nível de intensidade das componentes do espectro do sinal de voz. A variância da curva estimada de mascaramento apresenta para a rede uma referência da distorção máxima permitida pelo processo de quantização. E por fim, a variância da quantidade de bits alocada ao bloco anterior apresenta para a rede uma memória do mapeamento em bits da relação sinal-ruído de quantização.

#### **4.2.3 - Considerações finais**

Este capítulo descreveu os processos e algoritmos mais importantes desenvolvidos no decorrer das atividades de pesquisa. A ênfase foi dada às principais contribuições do trabalho.

No capítulo 5 serão abordados os resultados das simulações, bem como procurar-se-á discutir esses resultados, relacionando-os e comparando o desempenho obtido com os de outros trabalhos científicos publicados.

## 5 - RESULTADOS

Neste capítulo são apresentados os resultados dos testes dos dois algoritmos de compressão objetos da pesquisa. Inicialmente são apresentados os resultados do algoritmo de compressão de sinais eletrofisiológicos. No item 5.1 são apresentados os resultados do algoritmo de compressão de sinais eletrofisiológicos. Na seção 5.1.1 mostram-se os resultados da minimização de ruído destes sinais, e, na seção 5.1.2, são apresentados os resultados que medem o desempenho dos algoritmos de codificação segundo duas métricas: o fator de compressão e o erro diferencial percentual. Os resultados da compressão dos sinais eletrofisiológicos estão separados por tipo de sinal: sinais de EMG e sinais de ECG. As curvas apresentadas representam na maior parte dos casos o resultado médio para todos os sinais testados, embora eventualmente tenham sido mostrados os resultados para cada sinal testado. Por fim, na seção 5.2, são apresentados os testes com o algoritmo de compressão dos sinais de voz de banda larga.

### 5.1 - RESULTADOS DA COMPRESSÃO DE SINAIS ELETROFISIOLÓGICOS

Os sinais de EMG isométricos utilizados para testar o algoritmo de compressão foram coletados do músculo *biceps brachii* de 14 indivíduos em esforços de tensão isométricos, mantendo 60% da sua contração voluntária máxima. Os sinais de EMG isotônicos utilizados foram coletados do músculo *vastus lateralis* de outros 14 indivíduos pedalando em um simulador de ciclismo (Cateye CS1000, USA). Nesses dois experimentos foram utilizados eletrodos de superfície pré-amplificados (modelo DE-02, DelSys Inc. Boston MA, USA). Os sinais foram alimentados em uma placa de aquisição com LABVIEW (NI-DAQ for Windows, National Instruments, USA). Todos os sinais foram amostrados a 2 kHz e quantizados com 12-bits. A duração de cada sinal varia de 3 a 6 minutos. A Figura 5.1 mostra um exemplo de sinal de EMG isométrico e um exemplo de sinal de EMG isotônico. Observe que o sinal isotônico apresenta variações mais acentuadas na sua faixa dinâmica. Essas variações indicam mudanças no recrutamento das unidades motoras do músculo e, portanto, variações no tempo das características espectrais do sinal, o que torna o processo de compressão mais difícil e exige um algoritmo adaptativo.

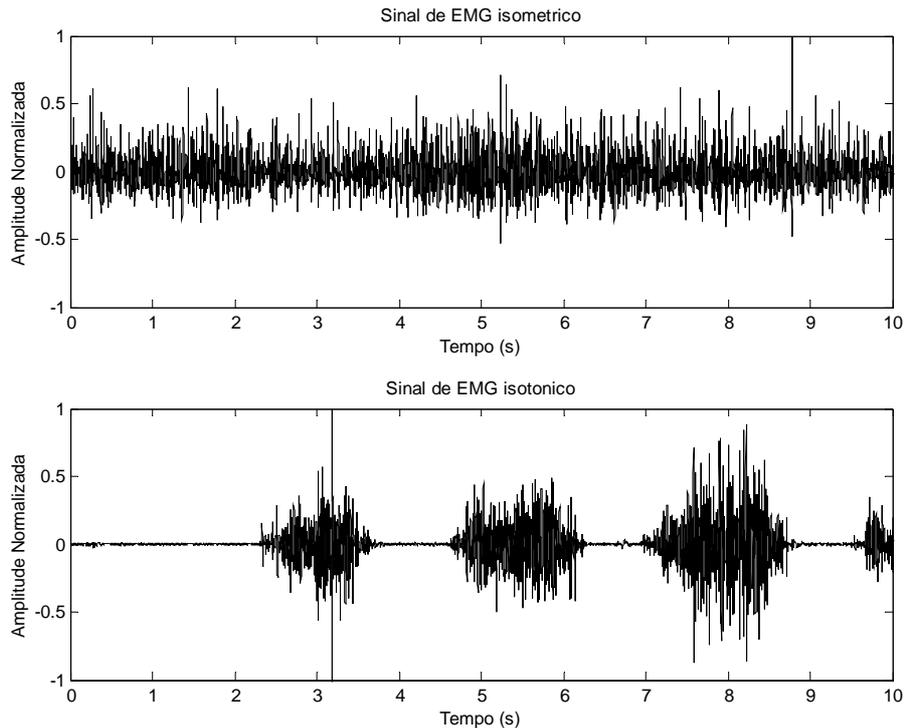


Figura 5.1 – Comparação entre sinais de EMG isométricos e isotônicos.

Os sinais de ECG que fazem parte deste trabalho são as amostras 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 117, 118 e 119 do *MIT-BIH Arrhythmia Database* e as amostras 08730\_01, 08730\_02, 08730\_03, 08730\_04, 08730\_05, 08730\_06, 08730\_07, 11442\_01, 11442\_02, 11442\_03, 11442\_04, 11950\_01, 11950\_02, 11950\_03, 11950\_04, 11950\_05, 12247\_01, 12247\_02, 12247\_03, 12247\_04 do banco de dados do *MIT-BIH Compression Test Database*.

Nas simulações realizadas, os sinais de voz têm largura de banda de 50 a 7000 Hz e foram amostrados à taxa de 16 kHz. O comprimento dos quadros foi fixado em 512 amostras ou, equivalentemente, 32 ms. Portanto, são gerados 256 coeficientes espectrais (ou MDCTs) a cada 16 ms. O espaçamento frequência entre coeficientes é de  $8.000 \div 256 = 31,25$  Hz.

O desempenho do algoritmo de compressão dos sinais eletrofisiológicos foi objetivamente medido usando dois critérios: O fator de compressão (CF) e a diferença percentual residual (PRD). Esses dois critérios são atualmente os mais utilizados pela comunidade científica para a avaliação da compressão de sinais eletrofisiológicos. O fator de compressão é definido na equação (5.1).

$$CF = \frac{O_s - C_s}{O_s} \times 100 \% \quad (5.1)$$

onde  $O_s$  é a quantidade de bits necessária para armazenar os dados originais e  $C_s$  é a quantidade de bits necessária para armazenar os dados comprimidos.

A diferença percentual residual é definida na equação (5.2):

$$PRD = \sqrt{\frac{\sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2}{\sum_{n=0}^{N-1} x^2[n]}} \times 100 \% \quad (5.2)$$

onde  $x$  é o sinal original,  $\hat{x}$  é o sinal reconstruído e  $N$  é o tamanho do seguimento de sinal.

O *de-noising* dos sinais eletrofisiológicos foi avaliado por outros dois critérios. Um é o erro quadrático médio normalizado (MSE) entre o sinal original e o sinal depois de processado pelo algoritmo de minimização de ruído; o outro é o percentual da quantidade de energia do sinal original que foi preservada no sinal após a minimização de ruído (%E). O MSE é definido na equação (5.3) e o %E é definido na equação (5.4).

$$MSE = \frac{\sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2}{\sum_{n=0}^{N-1} x[n]^2} \quad (5.3)$$

$$\%E = \frac{\sum_{n=0}^{N-1} \hat{x}[n]}{\sum_{n=0}^{N-1} x[n]} \times 100 \% \quad (5.4)$$

onde  $x$  é o sinal original,  $\hat{x}$  é o sinal reconstruído e  $N$  é o tamanho do seguimento de sinal.

O desempenho do algoritmo para a compressão de sinais de voz de banda larga e, quando conveniente, também o algoritmo para compressão de sinais eletrofisiológicos, foi medido utilizando a razão sinal ruído, SNR (do inglês, *Signal to Noise Ratio*) definida na equação (5.5) e o ganho de compressão (G) definido como a razão entre a quantidade de bits

despendida para representar o sinal original e a quantidades de bits resultante após a compressão.

$$SNR = 10 \log_{10} \left[ \frac{\sigma_x^2}{\sigma_e^2} \right] \quad (5.5)$$

onde  $\sigma_x^2$  é a variância do sinal original e  $\sigma_e^2$  é variância do erro de reconstrução.

Na avaliação do algoritmo para a compressão de sinais de voz de banda larga também foi utilizada a medida objetiva PESQ (*Perceptual Evaluation of Speech Quality*). O PESQ compara um sinal de voz original com um sinal degradado e sua saída é uma predição da percepção de qualidade que seria obtida por um indivíduo em teste subjetivo de escuta. A pontuação final de qualidade varia entre -0.5 e 4.5, embora na maioria dos casos a faixa de saída deva assumir valores semelhantes nos testes de escuta do tipo MOS (pontuação opinião média da qualidade de escuta), entre 1,0 e 4,5. O PESQ é um padrão formalmente estabelecido, documentado e divulgado pelo ITU-T (*International Telecommunications Union*) e, portanto, uma medida objetiva de qualidade confiável.

Sempre que não informado, os resultados apresentados correspondem à média dos resultados de todos os sinais que estão sendo estudados.

### 5.1.1 - Resultados do *de-noising*

Conforme mencionamos anteriormente, foi desenvolvida uma metodologia para encontrar o limiar de amplitude ótimo para o descarte de coeficientes transformados. Essa metodologia consistiu em adicionar uma quantidade fixa e controlada de ruído gaussiano branco aos sinais originais e depois processá-los com o algoritmo de minimização de ruído, variando o limiar de amplitude. Nessa simulação, calculou-se o MSE entre o sinal resultante da minimização de ruído e o sinal original. Assim, o limiar de amplitude que resulta no menor MSE pode ser considerado ótimo em termos de ganho de codificação.

Essa metodologia foi realizada em todos os sinais de EMG coletados. Na Figura 5.2 mostramos a média dos resultados (MSE *versus* o limiar de amplitude) de todos os sinais de EMG isotônicos coletados, e na Figura 5.3 a média dos resultados de todos aos sinais de EMG isométricos coletados.

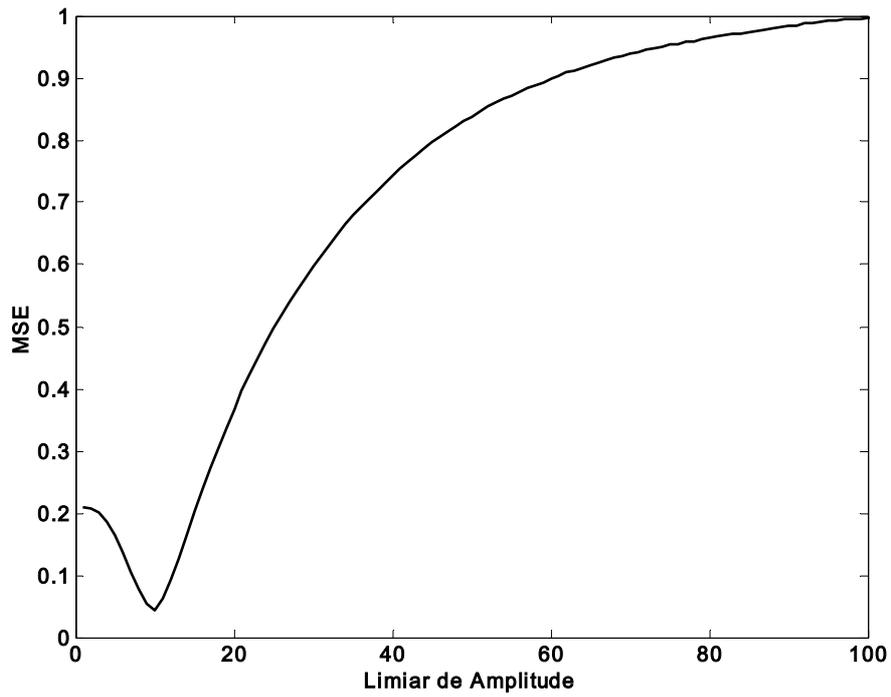


Figura 5.2 – Comparação entre o MSE e o limiar de amplitude para os sinais de EMG isotônicos.

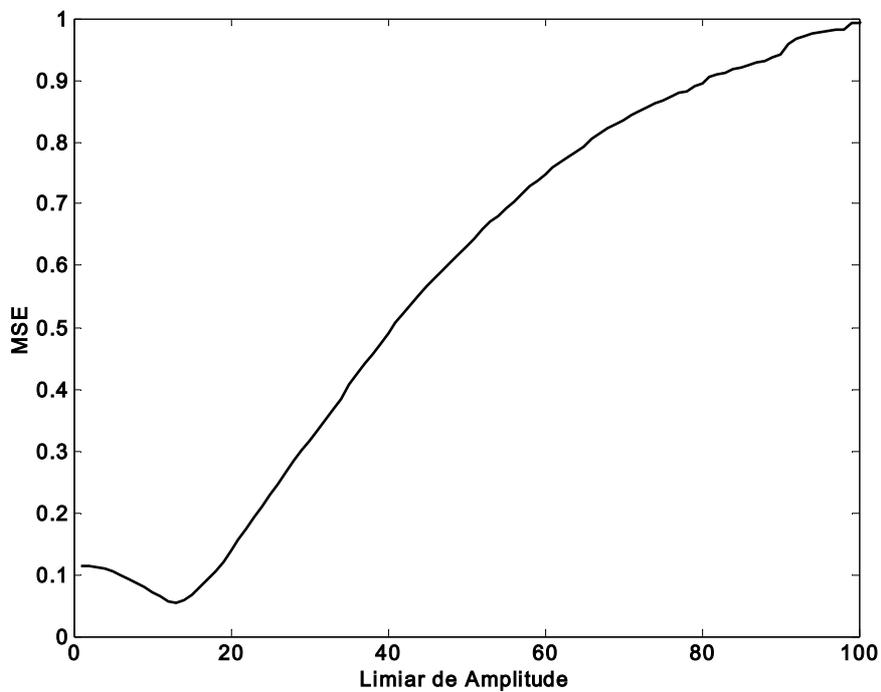


Figura 5.3 – Comparação entre o MSE e o limiar de amplitude para os sinais de EMG isométricos.

Dos resultados da Figura 5.2 e da Figura 5.3 pode-se chegar à conclusão que o limiar de amplitude ótimo para os sinais de EMG isotônicos é de 10%, e de 16% para os isométricos. Conforme já mencionamos, os sinais de EMG isotônicos possuem maiores variações no tempo das suas características espectrais do que os sinais de EMG isométricos. Por isso o limiar de amplitude ótimo para os sinais isotônicos é mais baixo, ou seja, esses sinais são mais sensíveis ao “descarte” de coeficientes transformados.

Outra metodologia empregada para avaliar o *de-noising* foi observar a quantidade de energia do sinal original que é preservada, calculada pela métrica %E, quando se aumenta progressivamente o limiar de amplitude. Para o limiar de amplitude ótimo a quantidade de energia preservada é, em média e aproximadamente, %E=93,04 para os sinais de EMG.

Além das medidas objetivas apresentadas até agora, a Figura 5.4 mostra uma comparação, que serve para uma análise subjetiva, entre um sinal de EMG original e o seu correspondente após a *de-noising*. A Figura 5.5 mostra a mesma comparação com uma resolução maior no eixo do tempo.

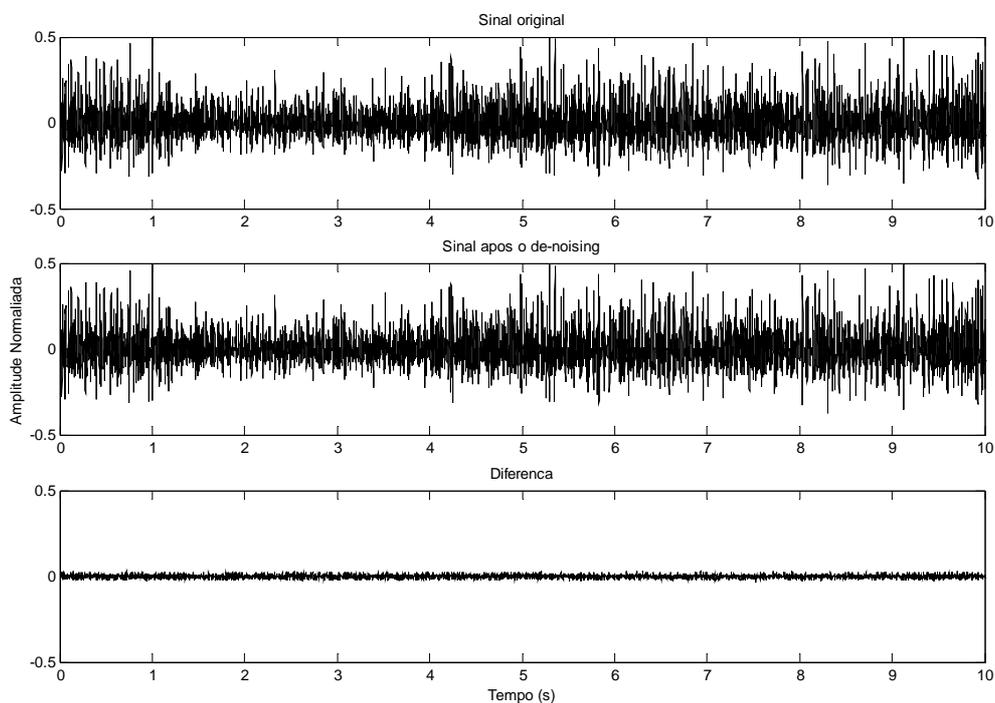


Figura 5.4 – Comparação entre um sinal de EMG original e o seu correspondente após o *de-noising*.

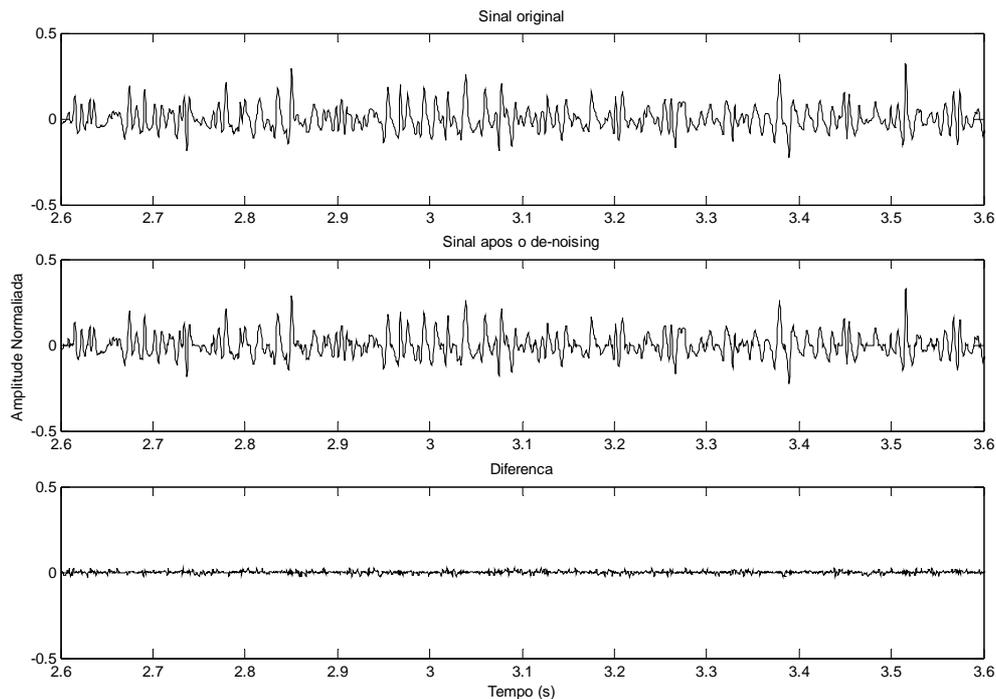


Figura 5.5 – Comparação entre um sinal de EMG original e o seu correspondente após o *de-noising*.

Com essa resolução no eixo do tempo não há distorção visível. Desta forma, se este sinal fosse usado por um especialista para um diagnóstico baseado em uma inspeção visual, o *de-noising* não teria impacto na sua decisão.

A Figura 5.6 mostra o MSE *versus* o limiar de amplitude aplicado aos coeficientes transformados dos sinais de ECG, onde se pode observar que, nesse caso, o limiar de amplitude ótimo foi de 9%.

A Figura 5.7 apresenta uma comparação entre um sinal de ECG original e seu correspondente depois do *de-noising* e a Figura 5.8 apresenta a mesma comparação com uma resolução maior nos eixos do tempo e da amplitude.

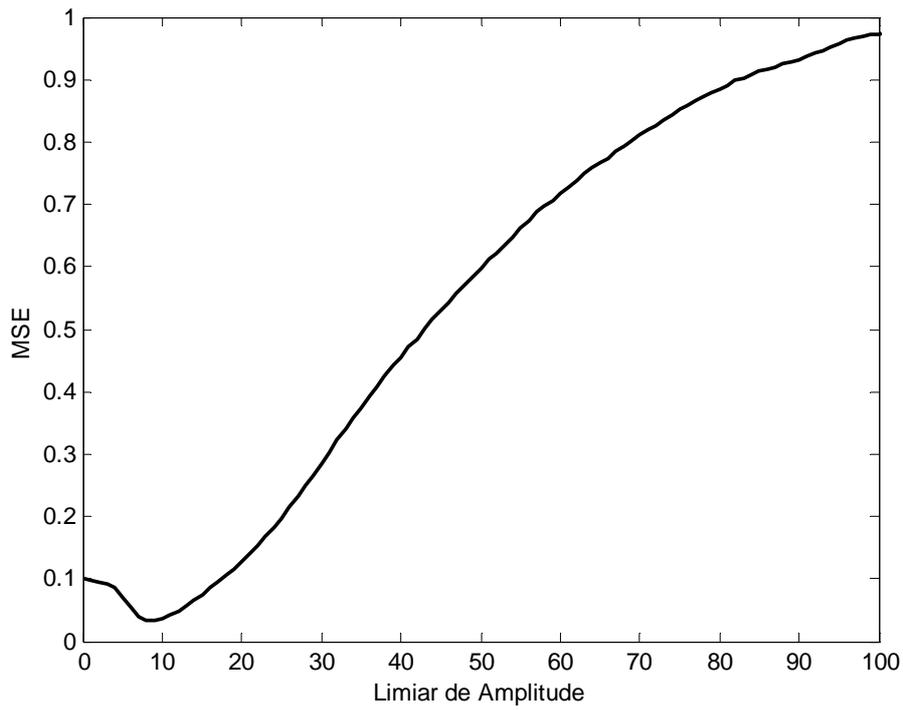


Figura 5.6 – MSE *versus* limiar de amplitude ótimo para sinais de ECG.

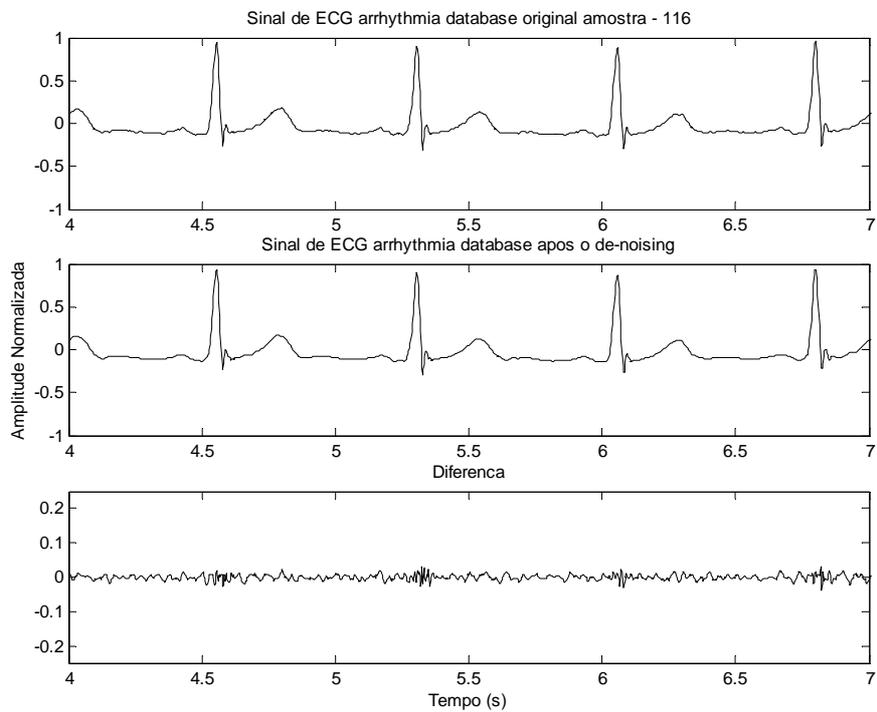


Figura 5.7 - Comparação entre um sinal de ECG original e seu correspondente após o *de-noising*.

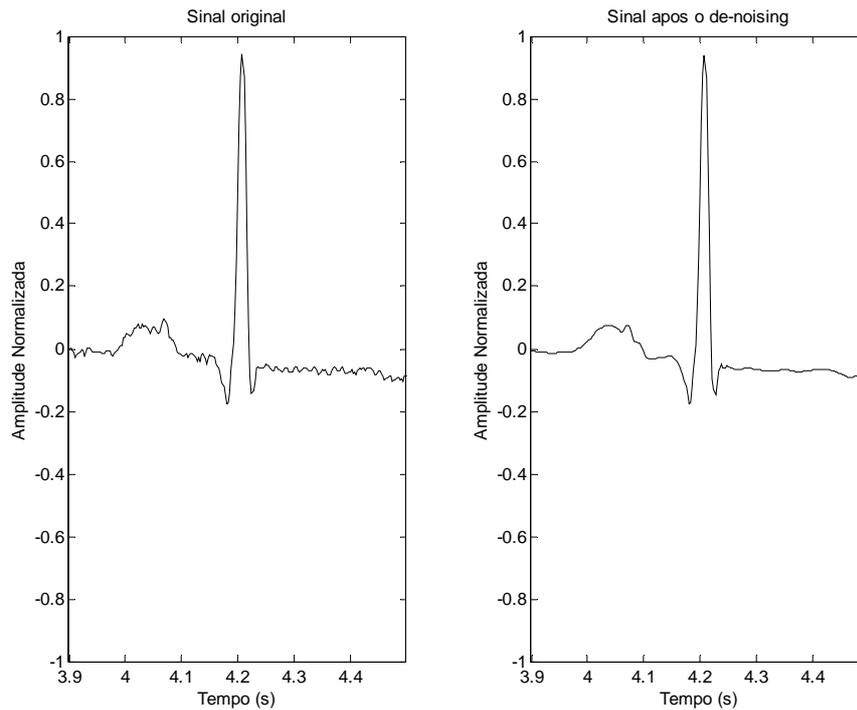


Figura 5.8 – Comparação entre um sinal de ECG original e seu correspondente após o *de-noising*.

Pode-se observar na Figura 5.8, a fidelidade da representação da forma de onda decodificada quando comparada com a forma de onda original. Existe um perfeito ajuste na faixa dinâmica, como também, se tem uma boa reprodução das pequenas amplitudes. As variações de larguras espectrais também são bem reproduzidas. Além disso, é visível o papel do *de-noising* eliminando as componentes espúrias de baixas amplitudes e alta frequência.

### 5.1.2 - Resultados da compressão dos sinais eletrofisiológicos

Conforme foi acompanhado na seção anterior, a compressão dos sinais eletrofisiológicos pode ser precedida por um processo de *de-noising*. O *de-noising* pode ser controlado através do limiar de amplitude escolhido durante o descarte dos coeficientes transformados. A escolha deste limiar influencia o desempenho do algoritmo de compressão. Assim, primeiro apresentaremos os resultados do algoritmo de compressão sem realizar o *de-noising* nos sinais eletrofisiológicos. A seguir, apresentaremos os resultados com o *de-noising* como pré-processamento.

A Figura 5.9 mostra o fator de compressão *versus* a diferença percentual residual resultante do algoritmo de compressão, quando os sinais de entrada são sinais de EMG isométricos não

submetidos ao algoritmo de *de-noising*. Estes resultados incluem uma curva para cada sinal isométrico pertencente ao banco de sinais coletados, e uma curva média para todos os resultados. A variação do CF é alcançada variando-se o fator de qualidade definido na equação (4.1). A Figura 5.10 mostra o mesmo tipo de resultado da figura anterior para os sinais de EMG isotônicos.

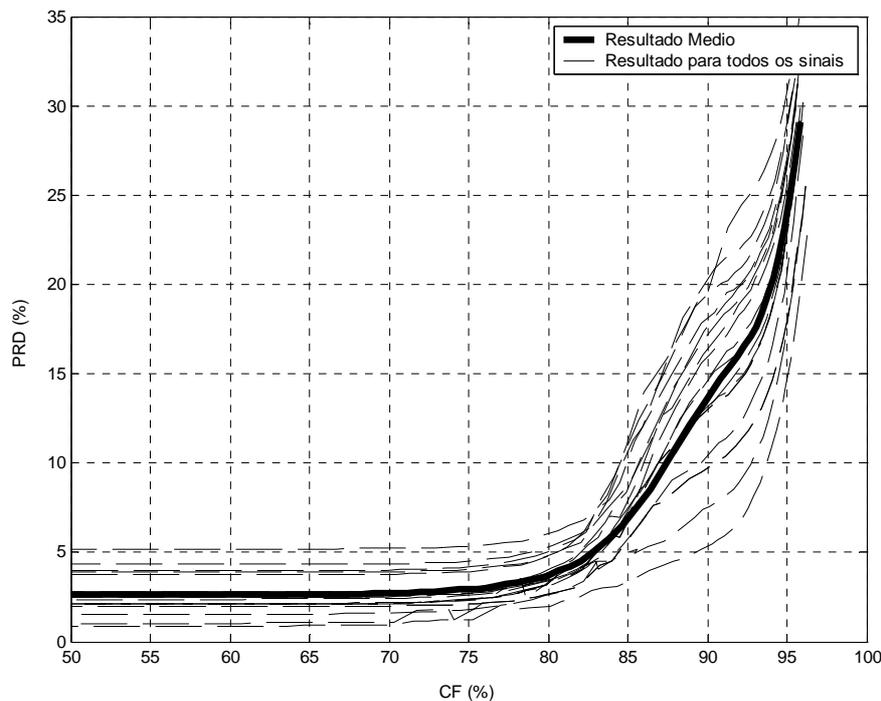


Figura 5.9 – Resultado do algoritmo de compressão para sinais de EMG isométricos sem *de-noising*.

Na Figura 5.9, a variação do fator de compressão CF é alcançada variando-se o fator de qualidade Q. À medida que se diminui o fator de qualidade Q, o fator de compressão CF aumenta. Como resultado do aumento de CF, a diferença residual percentual PRD permanece quase inalterada até cerca de 85% de compressão. A partir desse ponto, qualquer pequeno aumento no fator de compressão ocasiona uma grande deterioração do sinal reconstruído após a decodificação.

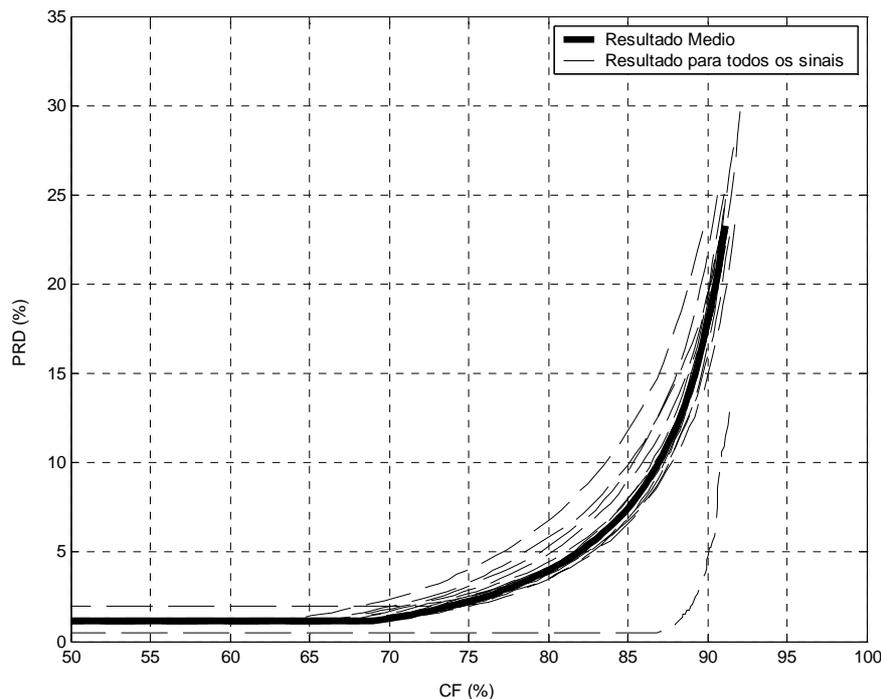


Figura 5.10 – Resultado do algoritmo de compressão para sinais de EMG isotônicos sem *de-noising*.

Os resultados apresentados na Figura 5.10 demonstram o mesmo comportamento observado nos resultados da Figura 5.9. Vale ressaltar, que nesse resultado, os sinais de EMG isotônicos não foram submetidos ao algoritmo de *de-noising*.

Neste ponto não é possível comparar, de forma precisa e confiável, os resultados do algoritmo proposto com aqueles algoritmos baseados na transformada *wavelet* apresentados na revisão bibliográfica, visto que tais algoritmos foram testados em conjuntos de dados diferentes. O algoritmo proposto por Wellig *et al* (1998) foi testado em sinais de EMG intramusculares durante esforços de tensão isométricos e tais sinais foram amostrados a 10 kHz e quantizados com 12 bits. Esses sinais são bem diferentes dos sinais utilizados nesta tese e a comparação com o algoritmo proposto aqui não é apropriada.

O algoritmo proposto por Norris *et al* (2001) foi testado com sinais de EMG medidos durante esforços de tensão isométricos e isotônicos. Os sinais de EMG isométricos foram medidos no músculo *biceps brachii*, amostrados a 2 kHz e quantizados com 12 bits. Então, à primeira vista esses sinais são semelhantes aos utilizados no presente trabalho. Entretanto, detalhes como a distância entre eletrodos e a %MVC não foram relatados no trabalho de Norris. Assim, é possível que os sinais utilizados por Norris, sejam diferentes daqueles utilizados

nesta pesquisa. Para os sinais de EMG isotônicos, a referência em questão usou sinais medidos no músculo *biceps brachii* durante contrações isotônicas cíclicas, enquanto os sinais objeto deste estudo foram medidos no músculo *vastus lateralis* durante atividades de ciclismo. Assim, esses dois conjuntos de sinais são de fato diferentes uma vez que envolvem músculos e volumes condutores diferentes. Além disso, existem outras diferenças nos dois protocolos, como as velocidades, as cargas utilizadas e a duração dos sinais. Contudo, buscando enriquecer a discussão dos resultados, será apresentada a seguir uma comparação entre esses dois trabalhos, com o alerta de que essas comparações devem ser analisadas com cautela.

Para avaliar a taxa de compressão e a distorção resultante da compressão, o trabalho de Norris *et al* (2001) também utilizou a CF e a PRD. Eles compararam o desempenho de um algoritmo de compressão baseado na transformada EZW com o desempenho da compressão utilizando a transformada *wavelet* padrão. Nas análises seguintes serão levados em consideração somente os resultados do algoritmo baseado na EZW.

A Figura 5.11 apresenta uma comparação entre os resultados obtidos por Norris *et al* (2001) e o algoritmo proposto nesta tese. Esses resultados mostram que os dois algoritmos têm “pontos de quebra” semelhantes (CF em torno de 85%) e o algoritmo proposto neste trabalho apresenta um desempenho um pouco melhor. No geral, a diferença média na PRD entre os dois algoritmos, para a faixa útil, foi de cerca de 0.8%.

A comparação para os resultados isotônicos é apresentada na Figura 5.12. Esses resultados mostram que, para esses conjuntos de sinais, os dois algoritmos também têm “pontos de quebra” semelhantes (CF em torno de 85%). Novamente, o algoritmo proposto nesta pesquisa tem um desempenho um pouco melhor. No geral, a diferença média na PRD entre os dois algoritmos, para a faixa útil, foi de cerca de 3.6%. É coerente assumir que a diferença de desempenho foi maior para o conjunto de sinais isotônicos por causa da alocação dinâmica de bits.

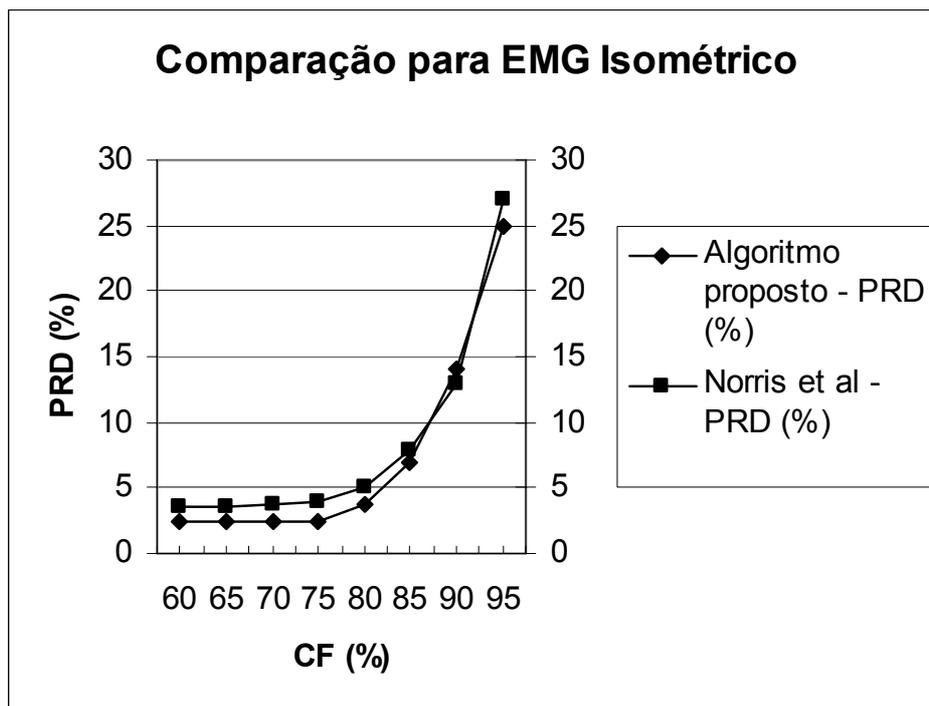


Figura 5.11 – Comparação entre os resultados do algoritmo proposto neste trabalho e o trabalho de Norris *et al* para sinais de EMG isométricos.

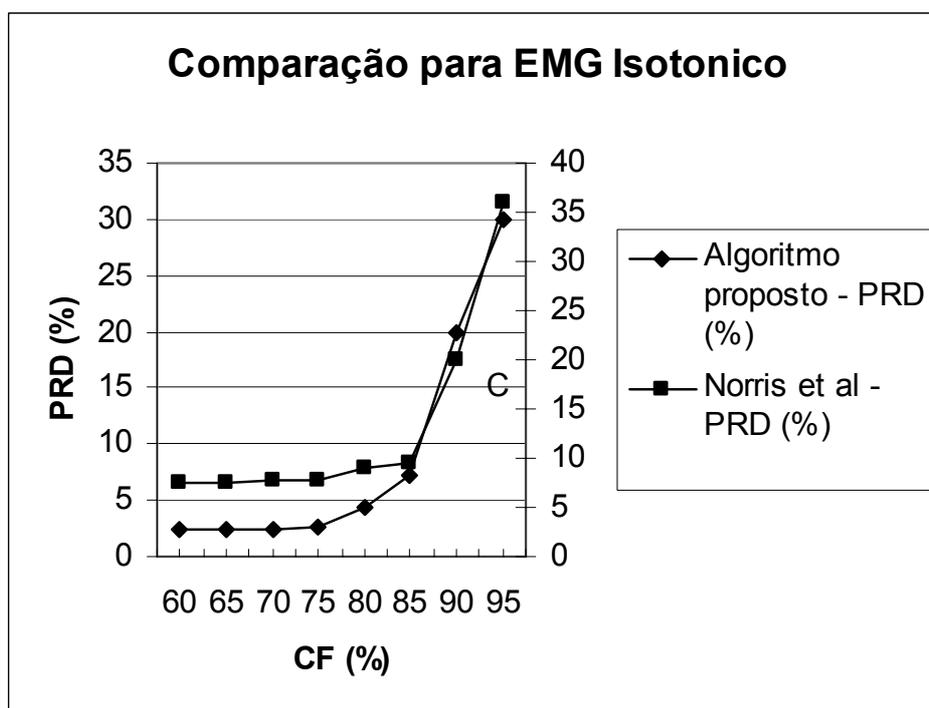


Figura 5.12 – Comparação entre os resultados do algoritmo proposto neste trabalho e o trabalho de Norris *et al* para sinais de EMG isotônicos.

A Figura 5.13 mostra o fator de compressão *versus* a diferença residual percentual resultante do algoritmo de compressão, quando os sinais de entrada são sinais de ECG do banco de

dados *MIT-BIH Arrhythmia Database*, não submetidos ao *de-noising*. A Figura 5.14 mostra o fator de compressão *versus* a diferença residual percentual resultante do algoritmo de compressão, quando os sinais de entrada são sinais de ECG do banco de dados *MIT-BIH Compression Test Database*, não submetidos ao *de-noising*.

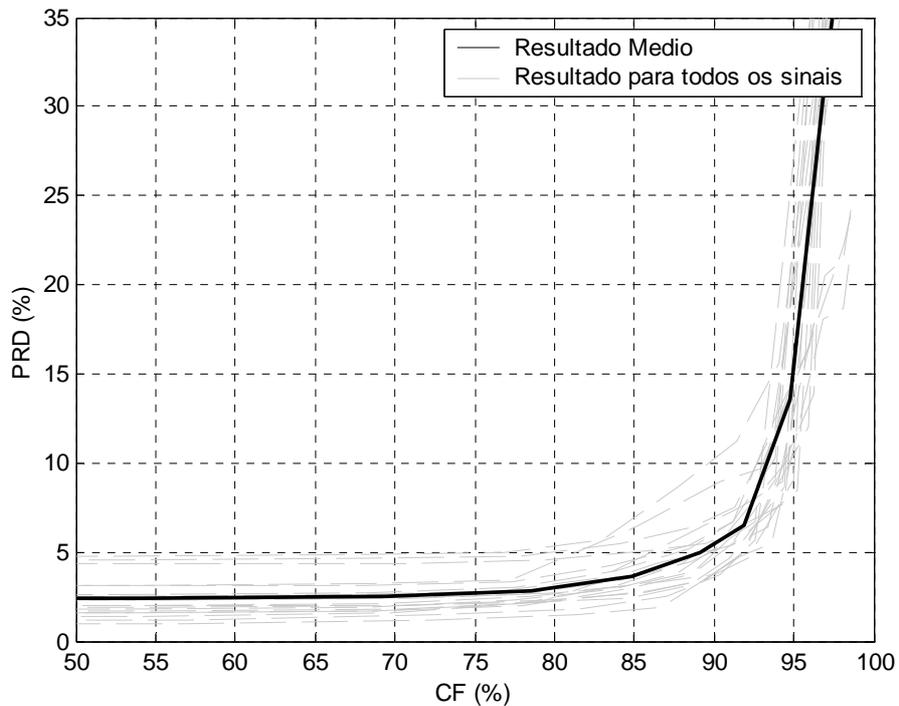


Figura 5.13 - Resultado do algoritmo de compressão para sinais de ECG do banco de dados MIT-BIH Arrhythmia Database, não submetidos ao *de-noising*.

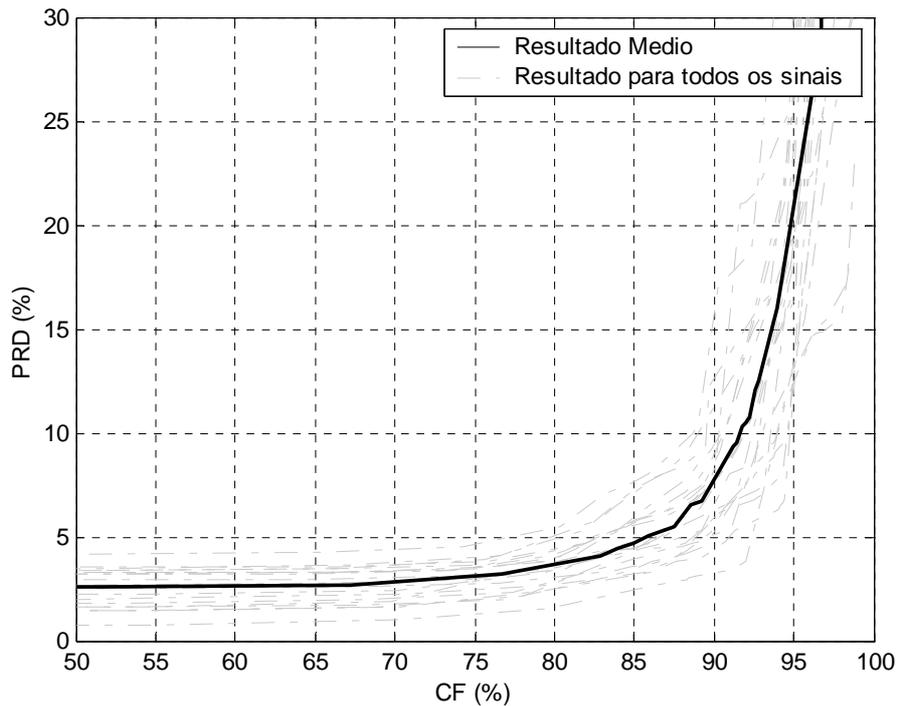


Figura 5.14 – Resultado do algoritmo de compressão para sinais de ECG do banco de dados *MIT-BIH Compression Test Database*, não submetidos ao *de-noising*.

É interessante notar que a compressão dos sinais de ECG do banco de dados *MIT-BIH Compression Test Database* é realmente mais difícil, por isso a PRD na Figura 5.14 é maior do que na Figura 5.13.

Quanto aos resultados da compressão dos sinais de ECG nota-se que o algoritmo proposto apresenta resultados comparáveis aos resultados apresentados em (Istepanian *et al*, 2001). Com relação aos trabalhos de (Hilton, 1997; Lu *et al*, 2000), pode-se notar uma melhora de desempenho (diminuição de pelo menos 2% na PRD para o mesmo fator de compressão). Essas comparações são diretas porque todos esses trabalhos também utilizaram o banco de dados *MIT-BIH Arrhythmia Database*.

Os resultados a seguir foram obtidos quando os sinais de entrada do algoritmo de compressão foram pré-processados pelo *de-noising*. Para cada tipo de sinal (EMG isométrico, EMG isotônico e ECG) foi usado o limiar de amplitude ótimo, obtido na seção 5.1.1.

A Figura 5.15 e a Figura 5.16 mostram os resultados do algoritmo de compressão quando os sinais de entrada de EMG (isométricos e isotônicos) são submetidos ao pré-processamento de *de-noising* com limiar de amplitude ótimo.

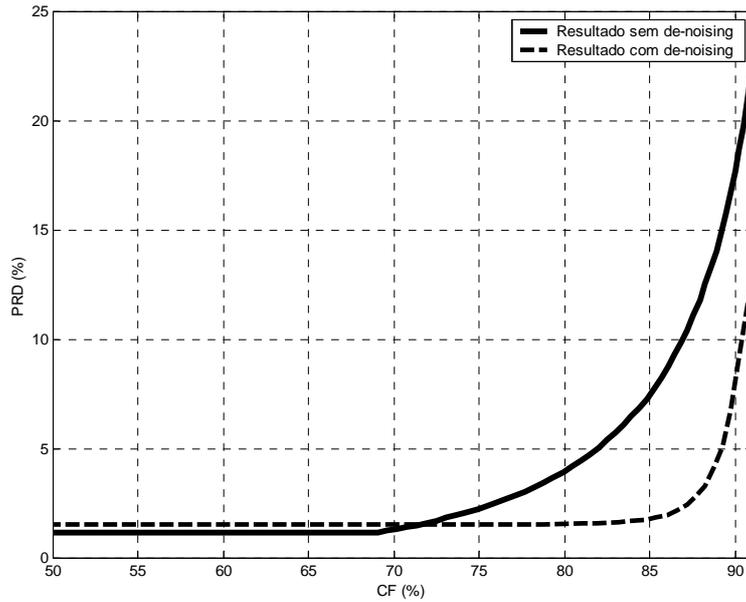


Figura 5.15 – Comparação dos resultado do algoritmo de compressão para sinais de EMG isotônicos, com e sem *de-noising*.

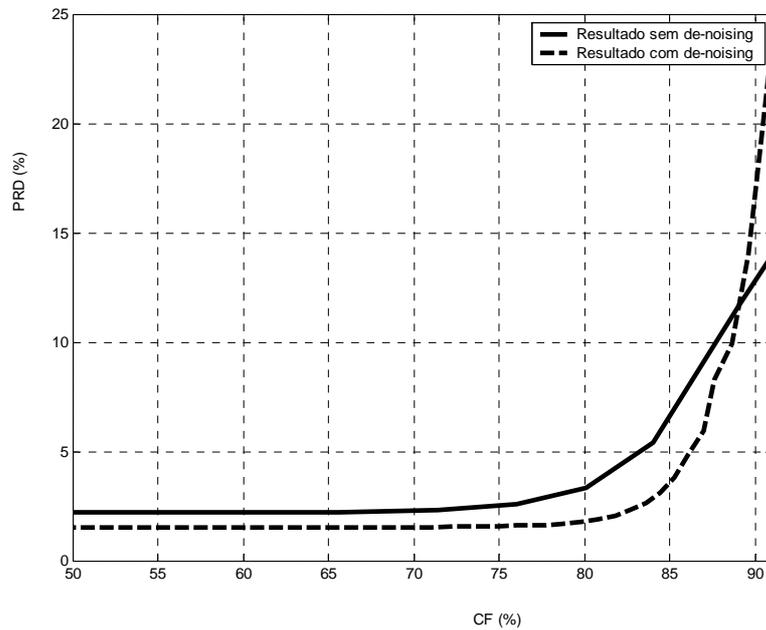


Figura 5.16 – Comparação dos resultados do algoritmo de compressão para sinais de EMG isotônicos, com e sem *de-noising*.

Pode-se observar uma melhora de desempenho considerável com relação aos resultados da Figura 5.9 e da Figura 5.10. Isto acontece porque, nos resultados anteriores, a forma de onda a

ser codificada continha coeficientes transformados ruidosos. Dessa forma, gastavam-se bits para representar informação irrelevante.

A Figura 5.17 mostra o resultado (média de todos os resultados do banco de sinais) do algoritmo de compressão para os sinais de entrada são sinais de ECG do banco de dados *MIT-BIH Arrhythmia Database*, quando esses são submetidos ao pré-processamento com *de-noising* utilizando o limiar de amplitude ótimo. Esse resultado é comparado com o resultado anterior, quando se utilizaram sinais não pré-processados com o algoritmo de *de-noising*.

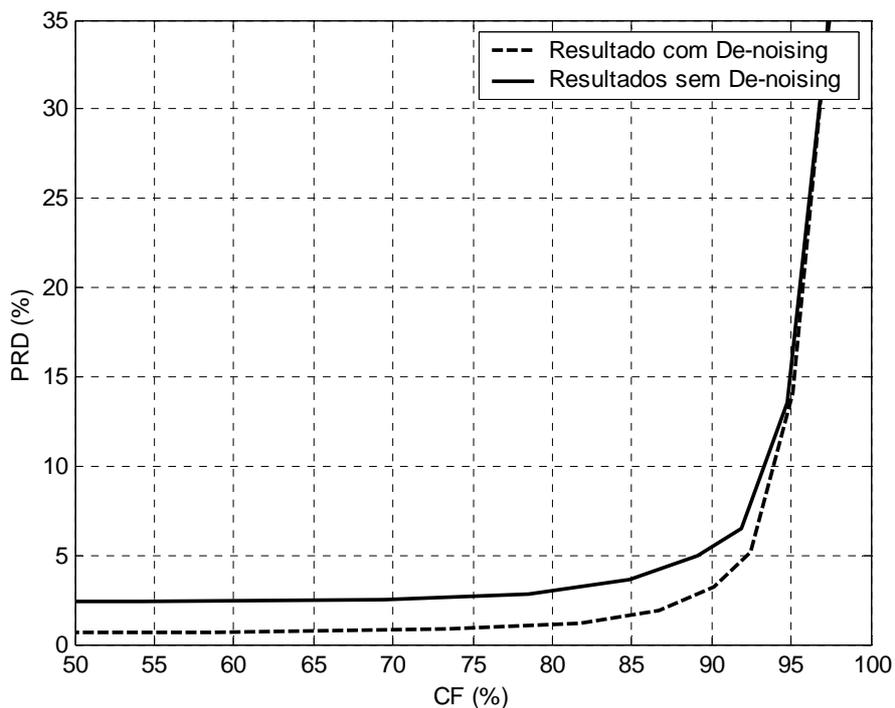


Figura 5.17 – Comparação dos resultados da compressão de sinais de ECG do banco de dados *MIT-BIH Arrhythmia Database*, com e sem *de-noising*.

Na Figura 5.18 é apresentado o resultado (média de todos os resultados do banco de sinais) do algoritmo de compressão para os sinais de entrada de ECG do banco de dados *MIT-BIH Compression Test Database*, quando esses são submetidos ao pré-processamento com *de-noising* utilizando o limiar de amplitude ótimo, comparado com o resultado anterior, em que esses sinais não foram pré-processados com o *de-noising*.

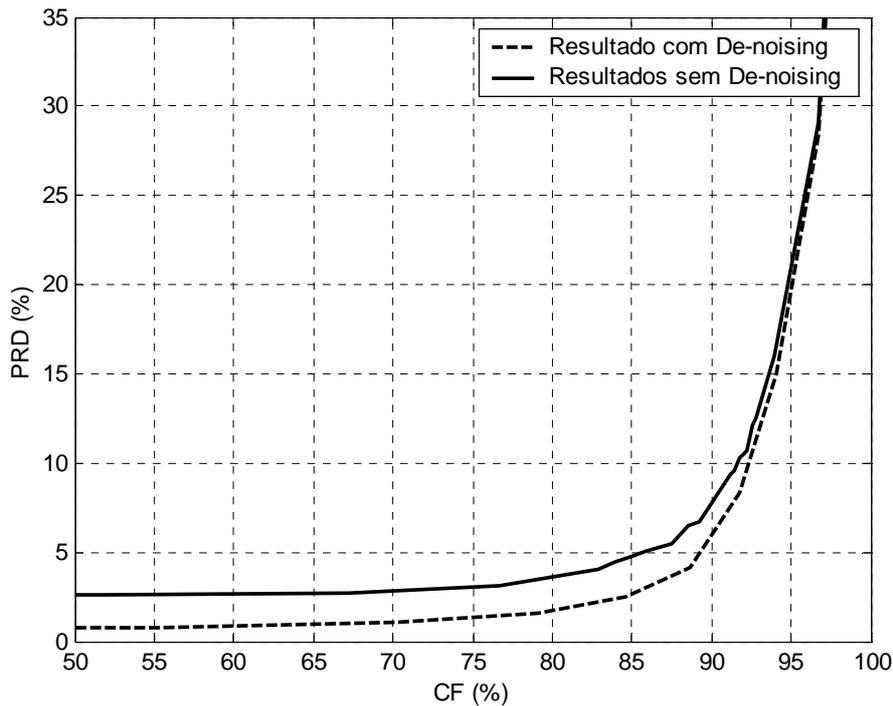


Figura 5.18 – Comparação dos resultados da compressão de sinais de ECG do banco de dados *MIT-BIH Compression Test Database*, com e sem *de-noising*

## 5.2 - RESULTADOS DO ALGORITMO DE CODIFICAÇÃO DE VOZ DE BANDA LARGA

Os resultados apresentados para o algoritmo de codificação de voz de banda larga foram obtidos com um conjunto de sinais de voz adquiridos de 10 indivíduos (5 homens e 5 mulheres) ao pronunciarem a frase “compressão de voz de banda larga”. Esses sinais de voz foram amostrados a uma taxa de 16kHz e quantizados com 16 bits.

Foram realizadas simulações do codificador descrito para três taxas de bits: 32, 24 e 16 kbps. A Tabela 5.3 apresenta a estratégia de codificação dos expoentes e a divisão, entre expoentes e mantissas, dos bits disponíveis por quadro.

Os sinais codificados com as taxas de 32 e 24 kbps têm qualidade subjetiva equivalente. É possível notar um leve abafamento, provavelmente devido a falhas na reprodução dos componentes de alta frequência. Nota-se também um ruído de baixa intensidade sibilado, que é devido aos componentes cujas mantissas não foram transmitidas e que foram substituídas por valores aleatórios. Contudo, o sinal com esse ruído é preferível ao sinal em cuja

reconstrução fazem-se iguais a zero as mantissas não transmitidas. Nesse último sinal é bem perceptível a ausência dos componentes de alta frequência. Os sinais codificados com 16 kbps apresentam uma degradação bem mais perceptível. Nota-se nesses sinais um ruído mais intenso e de espectro amplo, enquanto que no ruído presente nos sinais codificados com 32 ou 24 kbps predominam os componentes de alta frequência. Medidas objetivas também detectam essa maior degradação.

A Tabela 5.1 mostra os resultados das medidas PESQ para as diferentes taxas de bits testadas. A Tabela 5.2 – Escala MOS. mostra a equivalência entre essas medidas e a escala MOS. A redução da taxa de bits de 32 kbps para 24 kbps praticamente não alterou o valor PESQ, mantendo-se na faixa MOS de “reconstrução boa”. Já para 16 kbps a medida PESQ caiu cerca de 0,6 pontos, passando para a faixa de opinião MOS de “reconstrução boa”. Esses valores PESQ são a media dos valores para todo o conjunto de sinais testados.

Tabela 5.1 – Resultados PESQ para diferentes taxas de bits.

Taxa de bits	32 kbps	24 kbps	16 kbps
PESQ	4,1317	4,124	3,4787

Tabela 5.2 – Escala MOS.

MOS	Definição
5	Excelente
4	Boa
3	Regular
2	Pobre
1	Ruim

A Figura 5.19 mostra uma comparação entre um segmento de sinal de voz original e um segmento de sinal de voz decodificado para uma taxa de 32 kbps.

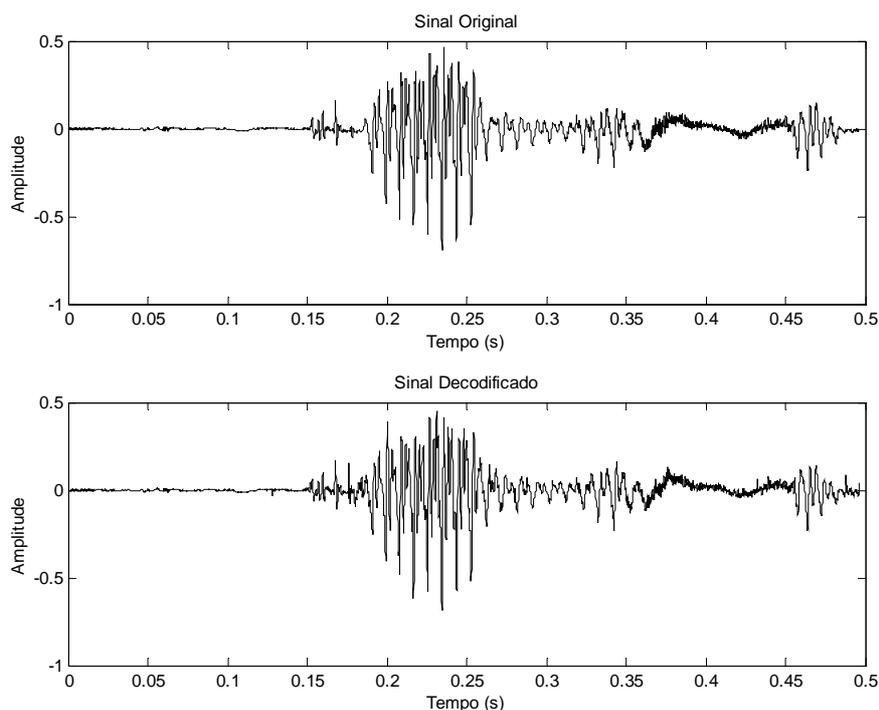


Figura 5.19 – Comparação entre um segmento de sinal de voz original e um segmento de sinal de voz decodificado para uma taxa de 32 kbps

A Tabela 5.4 mostra exemplos de valores medidos de razão sinal-ruído para sinais codificados com as três taxas de bits. Enquanto a redução à taxa de bits de 32 para 24 kbps levou a uma queda de 1,33 dB na SNR, a redução de 24 para 16 kbps levou a uma queda de 2,48 dB na SNR.

Tabela 5.3 – Estratégia de codificação e a alocação de bits para as taxas de 32, 24 e 16 kbps.

Taxa de bits global		32 kbps	24 kbps	16 kbps
Agrupamento dos expoentes		AU	AU	ANU
Reutilização dos expoentes		Não	Sim	Sim
Número médio de bits gasto por quadro (16 ms)	Expoentes	264	$264 \div 2 = 132$	$159 \div 2 = 79,5$
	Mantissas	248	252	176

Tabela 5.4 – Desempenho do codificador, em termos da razão sinal-ruído.

Taxa de bits	32 kbps	24 kbps	16 kbps
SNR	13,12 dB	11,79 dB	9,31 dB

A Figura 5.20 mostra a quantidade de ciclos de busca pela alocação de bits ótima na codificação dos sinais acima, comparando o padrão original com o padrão sugerido neste teste usando redes neurais artificiais do tipo MLP. O eixo das abscissas representa o índice do bloco, e o eixo das ordenadas representa o número de ciclos de busca da Figura 4.8. Este resultado é a média dos resultados para todo o conjunto de sinais testados e foi alcançado utilizando-se 16 neurônios na camada intermediária da rede MLP descrita na seção 39.

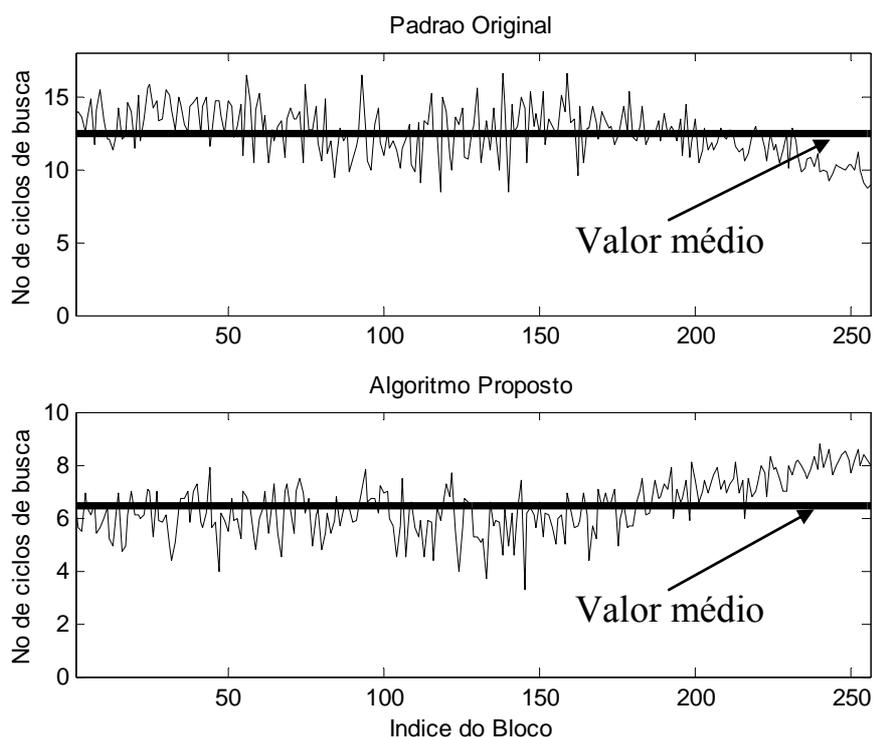


Figura 5.20 – Quantidade de ciclos na busca da alocação de bits ótima – comparação entre o padrão original e o algoritmo proposto.

O resultado da Figura 5.20 mostra que o desempenho médio do algoritmo aqui proposto é melhor, pois ele consegue atingir a alocação de bits desejada com quase a metade do número de ciclos de busca do padrão original.

Um outro tipo de teste realizado foi a simulação de perda de blocos de coeficientes MDCTs e a substituição desses blocos por versões obtidas pela interpolação dos coeficientes MDCTs dos blocos anterior e posterior ao bloco (ou blocos) perdido(s). Com o esquema que resultou em taxa de 32 kbps, a perda de até 10% dos blocos causa degradação quase imperceptível, e com perda de 20% dos pacotes a qualidade é ainda razoável, sendo que a inteligibilidade ainda é muito boa. Com os outros dois esquemas há uma deterioração maior, mas com perda

de até 10% dos pacotes se tem ainda uma qualidade razoável e boa inteligibilidade. A razão para a maior sensibilidade desses dois esquemas é o fato de eles reutilizarem os expoentes codificados. Nesse caso, quando um bloco que contém expoentes é perdido, o próximo bloco também fica perdido.

Tabela 5.5 – Medida PESQ para todos os sinais de voz do conjunto de teste para uma simulação de perda de 10% dos bloco de coeficientes MDCT.

Sinal de Voz	PESQ
1	3.904
2	3.121
3	3.155
4	3.203
5	3.394
6	4.128
7	3.048
8	3.977
9	3.423
10	4.255
Média	3.5608

Para alguns dos sinais de voz testados a medida PESQ ficou próxima de 4 (faixa de opinião MOS de “reconstrução boa”), mas a média dos resultados apresenta uma performance um pouco pior, ficando com PESQ em torno de 3.5 (faixa de opinião MOS de “reconstrução regular”).

## 6 - CONCLUSÕES

Este trabalho colaborou com o desenvolvimento de tecnologias de codificação de sinais biológicos (sinais eletrofisiológicos e sinais de voz) através da criação, avaliação e, principalmente, da otimização de codificadores utilizando redes neurais artificiais.

Apresentou-se um algoritmo de compressão de sinais eletrofisiológicos (sinais de eletromiografia isométricos, sinais de eletromiografia isotônicos e sinais de eletrocardiografia) baseado na técnica de compressão de sinais por transformadas ortogonais, mais especificamente a transformada *wavelet* discreta. A sua alocação de bits é dinâmica e utiliza uma rede neural artificial para otimizar a alocação de bits para adequar-se aos diferentes tipos de sinais e suas características. A rede neural artificial utilizada é uma camada de Kohonen. Os sinais eletrofisiológicos foram pré-processados por um algoritmo de *de-noising* invariante ao deslocamento utilizando a transformada *wavelet* discreta. Os resultados mostraram que o *de-noising* melhora significativamente o desempenho do algoritmo de compressão.

Na compressão de sinais de eletrocardiografia (EMG) isométricos, o algoritmo proposto conseguiu alcançar um fator de compressão (CF) médio na faixa de 50% a 85% com uma diferença percentual residual (PRD) média na faixa de 2,4% a 7,3%. Para sinais de EMG isotônicos, com a mesma variação de CF, o algoritmo proposto nesta tese apresentou a PRD média entre 0,4% e 7,2%. Demonstramos que, para sinais de EMG isométricos, o processo de compressão aqui descrito possui desempenho melhor do que os algoritmos baseados na transformada *wavelet* padrão descritos na literatura, conseguindo resultados que são comparáveis aos codificadores que utilizam algoritmos baseados na EZW (do inglês, *Embedded Zero-tree Wavelet*). Para os sinais de EMG isotônicos, o algoritmo proposto tem um desempenho que é melhor até do que os algoritmos baseados na EZW, apresentando uma diminuição de pelo menos 3,6% na PRD, para o mesmo fator de compressão.

Outra observação importante é que o algoritmo desenvolvido nesta tese tem performances semelhantes para os sinais de EMG isométricos e isotônicos, enquanto que os algoritmos descritos na literatura têm performances que são bem melhores para os sinais de EMG isométricos. Isso acontece por causa da alocação dinâmica de bits que permite ao codificador adaptar-se às variações no tempo das características espectrais dos sinais de EMG isotônicos.

Os resultados da compressão dos sinais de ECG mostraram que o algoritmo proposto apresenta resultados comparáveis aos resultados apresentados em (Istepanian *et al.*, 2001). E com relação aos trabalhos de (Hilton, 1997; Lu *et al.*, 2000), pode-se notar uma melhora de desempenho, diminuição de pelo menos 2% na PRD para o mesmo fator de compressão. Estes trabalhos utilizaram o mesmo banco de dados de sinais de teste que esta tese (o banco de dados *MIT-BIH Arrhythmia Database*) e portanto a comparação de desempenho não sofre restrições. Além deste banco de dados, o algoritmo proposto, ainda foi testado com o banco de dados *MIT-BIH Compression Test Database*. Esse outro banco de dados foi desenvolvido pelo MIT para tornar a tarefa dos codificadores de sinais de ECG mais árdua. Entretanto, o algoritmo que foi desenvolvido teve um desempenho tão bom quanto da compressão dos sinais do primeiro banco de dados. A robustez na compressão dos diferentes conjuntos de sinais de ECG pode ser atribuída ao algoritmo de alocação dinâmica de bits utilizado.

Apresentou-se também, um algoritmo para codificação de sinais de voz de banda larga. O algoritmo usa a codificação baseada em modelo perceptivo e transformada MDCT (do inglês, *Modified Discrete Co-sine Transform*). O modelo perceptivo proposto é baseado na codificação de áudio AC-3 (Todd *et al.*, 1994; G. Davidson *et al.*, 2001; ATSC *Revision A*” Doc. A/52<sup>a</sup>, 2001), com algumas modificações importantes, para que o codificador proposto pudesse tornar-se eficiente na codificação de sinais de voz de banda larga. Os resultados mostraram que as modificações no padrão original foram suficientes para adequar o codificador à aplicação de sinais de voz de banda larga (50 – 7000 Hz). Uma modificação de grande importância foi a utilização de uma rede neural artificial do tipo MLP (do inglês, *Multi-Layer Perceptron*) na estratégia de busca da alocação ótima de bits. Os resultados mostraram que a busca pela melhor alocação de bits com a rede MLP foi pelo menos 40% mais eficiente em termos de número de ciclos de busca.

Os resultados na compressão de voz mostraram que com o esquema desenvolvido nesta tese é possível codificar sinais de voz de banda larga a taxas de 32 kbps e 24 kbps com qualidade PESQ (*Perceptual Evaluation of Speech Quality*) de aproximadamente 4 pontos e, para a taxa de 16 kbps, com aproximadamente 3,5 pontos. Isto é, de acordo com a escala de opinião MOS (*Mean Opinion Score*) foi possível codificar os sinais de voz a 32 kbps e 24 kbps com uma qualidade boa, enquanto para a taxa de 16 kbps alcançou-se uma qualidade regular.

O algoritmo proposto para codificação de sinais de voz de banda larga mostrou-se robusto na simulação de perda de blocos de coeficientes MDCTs). Para taxas de 32 kbps, a perda de até 10% dos blocos causa degradação quase imperceptível e, com perda de 20% dos pacotes a qualidade é ainda razoável e a inteligibilidade ainda é muito boa. Para taxas de 16 e 24 kbps, com perda de 20% dos pacotes há uma deterioração maior, mas com perda de até 10% dos pacotes se tem ainda uma qualidade razoável e boa inteligibilidade.

Por fim, os resultados aqui apresentados evidenciam que a introdução das redes neurais artificiais pode ser de grande valia para o desenvolvimento de algoritmos de codificação de sinais, uma vez que essas técnicas foram testadas tanto em aplicações em que se privilegia a qualidade de reconstrução da forma de onda em detrimento da taxa de bits por amostra, quanto em aplicações em que se exige o contrário.

Uma sugestão para o desenvolvimento de trabalhos futuros é utilizar a transformada EZW no algoritmo de compressão dos sinais eletrofisiológicos. Para o algoritmo de compressão de sinais de voz de banda larga, pesquisas futuras poderiam ser realizadas utilizando uma rede neural artificial do tipo MLP para realizar todo o processo de alocação de bits, substituindo inclusive o atual modelo perceptivo.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Abenstein J. e Tompkins W. (1982), “New data-reduction algorithm for real-time ECG analysis”, *IEEE Trans. Biomed. Engineering*, v. 29, p. 43-48.
- Antoniol G. L. and Tonella P., 1997, “EEG data compression technique”, *IEEE Trans. Biomed. Engineering*, v. 44, n.2, p. 105-114.
- Arsenault H. and Genestar B., 1971, “Deconvolution of experimental data”, *Canadian Journal of Physics*, v. 49, p. 1865-1868.
- Basmajian J. V. and de Luca C. J., 1985, “Muscles Alive: Their Functions Revealed by Electromyography”, *Williams & Wilkins*.
- Berger P. A., Nascimento F. A. O., Carmo J. C. e Rocha A. F., 2006, “Compression of EMG Signals with Wavelet Transform and Artificial Neural Networks”, *Physiological Measurement*, v. 27, p. 457-465.
- Berger P. A., Nascimento F. A. O., Carmo J. C., Rocha A. F. and dos Santos I., 2003, “Algorithm for compression of EMG signals”, *Proc. 23th annual International Conference of the IEEE Engineering in Medicine Biology Society*, Cancun, p. 1299–1302.
- Berger, P. A., Nascimento F. A. O., Rocha A. F., Costa J. C., Carmo J. C., 2002, “Compressão de Sinais Eletromiográficos Usando Transformada de Wavelet Discreta. *Anais do XVIII Congresso Brasileiro de Engenharia Biomédica, São José dos Campos*.
- Berger, P. A., Nascimento F. A. O., Rocha A. F., Costa J. C., Carmo J. C., 2003, “Algoritmo para Compressão de EMG”, *Anais do X Congresso Brasileiro de Biomecânica, Ouro Preto, MG*, v. 1, p. 94-97.
- Berger, P. A., Nascimento F. A. O., Rocha A. F., Costa J. C., Carmo J. C., 2004, “Algoritmo para Compressão de Sinais de EMG Isotônicos Utilizando Redes Neurais Artificiais”, *Anais do III Congresso Latino-Americano de Engenharia Biomédica, João Pessoa*, p. 1103-1106.
- Berger, P. A., Nascimento F. A. O., Silva L. M., 2005, “Codificação Perceptiva de Sinais de Voz de Banda Larga”. *Anais do III Congresso da Sociedade de Engenharia de Áudio – AES, São Paulo*, p. 1-5.
- Berger, P. A., Nascimento F. A. O., Rocha A. F., Costa J. C., Carmo J. C., 2002, “Modelo para Simulação de Sinais EMG Isométricos”, *Anais do XVIII Congresso Brasileiro de Engenharia Biomédica, São José dos Campos*.
- Burrus, S. C., Gopinath, R. A., Guo, H. (1998), “Introduction to Wavelets and Wavelets Transforms: a primer”, London, Prentice-Hall International.
- Cox J., Noelle F., Fozzard H. e Oliver G.. 1968, “AZTEC: A preprocessing program for real-time ECG rhythm analysis”, *IEEE Trans. Biomed. Engineering*, v. 15, p. 128-129.
- Davidson G., Fielder L. e Link B., 1994, “Parametric Bit Allocation in a Perceptual Audio Coder”, *97th Convention of the Audio Engineering Society*, preprint 3921.

de Luca C. J., 1979, "Physiology and mathematics of myoelectric signal", *IEEE Trans. Biomed. Engineering*, v.11, n.4, p. 251-279.

Digital Television Standard, United States Advanced Television Systems Committee (ATSC), 2001, "ATSC Standard: Digital Audio Compression (AC-3), Revision A", Doc. A/52A.

Dipersio D. A. e Barr R. C., 1985, "Evaluation of the FAN method of adaptive sampling on human eletrocardiograms", *Med. Biomed. Eng. Computing*, p. 401-410.

Gersho A. and Gray R. M., 1992, "Vector quantization and signal compression", Norwell, NJ, Kluwer Academic Publishers.

Guerrero A. and Maihes C., 1997, "On the choice of an electromyogram data compression method", *Proc. 19th annual International Conference of the IEEE Engineering in Medicine Biology Society*, p. 1558–1561.

Haugland D., Heber J. e Hunsay J., 1997, "Optimization algorithms for ECG data compression" *Med. Biomed. Eng. Computing*, v. 35, p. 420-424.

Hebb, D., 1949, "The Organization of Behavior", New York, Wiley.

Helal J. and Duchene J., 1989, "A Pseudoperiodic Model for Myoelectric Signal During Dynamic Exercise", *IEEE Trans. Biomed. Engineering*, v. 36, n. 11, p. 1092-1097.

Hilton M. L., 1997, "Wavelet and wavelet packet compression of electrocardiograms", *IEEE Trans. Biomed. Engineering*, v. 44, p. 394-402.

Istepanian R. S. H., Hadjileontiadis L., Panas S., 2001, "ECG Data Compression using Wavelets and higher order Statistics Methods for Telemedical Applications", *IEEE Transactions on Information Technology in Biomedicine*, v. 5, n. 2, p. 108-115.

Jalaleddine S. M., Hutchens C. G., Satrattan R. D. and Coberly W. A., 1990, "ECG data compression techniques – the unified approach", *IEEE Trans. Biomed. Engineering*, v. 37, n. 4, p. 329-343.

Kohonen, T., 1988, "Self-Organization and Associative Memory", New York, Springer-Verlag.

Koilkpillai R.D. e Vaidyanathan P.P., 1992, "Cosine-Modulated FIR Banks Satisfying Perfect Reconstruction", *IEEE Trans. Signal Processing*, v. SP-40, p. 770-783.

Kovács, Z. L., 1997, "Redes Neurais Artificiais: Fundamentos e Aplicações", São Paulo, Collegium Cognitio.

Lu Z., Kim D. Y., Pearlman W. A., 2000, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees (SPIHT) algorithm", *IEEE Trans. Signal Processing*, v. 47, p. 849-856.

Mallat, S. 1999, "A Wavelet Tour of Signal Processing", London, Academic Press.

Malvar H., 1990, "Lapped Transforms for Efficient Transform/Subband Coding", *IEEE Trans. Acoust., Speech, Signal Processing*, v. ASSP-38, p. 969-978.

- McCulloch, W. e Pitts, W., 1943, "A logical Calculus of The Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics*, v. 5, p. 115-133.
- Merletti R. and Parker P., 2004, "Electromyography: Engineering and Noninvasive Applications", Hoboken, NJ, John Wiley & Sons.
- Mueller W., 1978, "Arrhythmia detection program for an ambulatory ECG monitor", *IEEE Trans. Biomed. Engineering*, v. 14, p. 81-85.
- Nave G. e Cohen A., 1993, "ECG compression using long-term prediction", *IEEE Trans. Biomed. Engineering*, v. 40, p. 877-885.
- N-Azghandi H. e Kabal P., 1997, "Perceptual Coding of Narrowband Audio Signals at 8 kbit/s", *Proc. IEEE Workshop on Speech Coding for Telecom.*, p. 109-110.
- Norris J. A., Englehart K. and Lovely D., 2001, "Steady-state and dynamic myoelectric signal compression using embedded zero-tree wavelets", *Proc. 23th annual International Conference of the IEEE Engineering in Medicine Biology Society*, p. 1879-1882.
- Norris J. F. and Lovely D. F., 1995, "Real-time compression of myoelectric data utilizing adaptive differential pulse code modulation", *Med. Biol. Eng. Comput.*, v. 33, p. 629-635.
- Nygaard R. e Haugland D., 1998, "Compressing ECG signals by piecewise polynomial approximation", *Proceedings of the Int. Conf. Acoust. Speech Signal Processing*, Seattle, p. 420-424.
- Painter T. e Spanias A., 2000, "Perceptual Coding of Digital Audio", *Proceedings of the IEEE*, v. 88, n. 4, p. 451-513.
- Princen J.P. e Bradley A.B., 1986, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation", *IEEE Trans. Acoust., Speech, Signal Processing*, v. ASSP-34, p. 1153-1161.
- Roseblatt, M., 1958, "The Perceptrons: A probabilistic model for information storage and organization in the brain", *Psychological Review*, v. 65, p. 386-408.
- Rumelhart, D. E. e Zipser, D., 1985, "Feature discovery by competitive learning", *Cognitive Science*, v. 9, p. 75-112.
- Rumelhart, D. E., Hinton, G. E. e Williams, R. J., 1986, "Learning internal representation by error propagation", Cambridge, MIT Press.
- Shahein C.F. e Abbas H.M., 1994, "ECG data-compression via cubic-splines and scan along polygonal approximation", *Signal Processing*, v. 35, n 3, p. 269-283.
- Tai S. C., 1991, "Slope – A real-time ECG data compressor", *J. Biomed. Eng. Computing*, v. 29, p. 175-179.
- Todd C., Davidson G., Davis M., Fielder L., Link B. e Vernon S., 1994, "AC-3: Flexible Perceptual Coding for Audio Transmission and Storage", *96th Convention of the Audio Engineering Society*, preprint 3796.

Wellig P., Zhenlan C., Semling M. and Moschytz G. S., 1998, "Electromyogram data compression using single-tree and modified zero-tree wavelet encoding", *Engineering in Medicine and Biology Society 23rd Annual International Conference of the IEEE*, p. 3 1303-1306.

Widrow, B., 1962, "Generalization and Information Storage in Networks of Adaptive Neurons", *Self-Organizing Systems*, p. 435-461.

Woods, W. A., 1986, "Important issues in knowledge representation", *Proceedings of the IEEE*, v. 74, p. 1322-1334.

Zaghetto A., Berger P. A., Nascimento, F. A. O., Rocha A. F., "Aplicação de Redes Neurais Artificiais na Alocação Dinâmica de Bits em Algoritmo de Compressão de Sinais de Eletromiografia", *Anais do III Congresso Brasileiro de Computação*, Itajaí, SC, p. 2330-2341.

## **APÊNDICES**

## A - RESULTADOS DO *DE-NOISING* DE SINAIS ELETROFISIOLÓGICOS

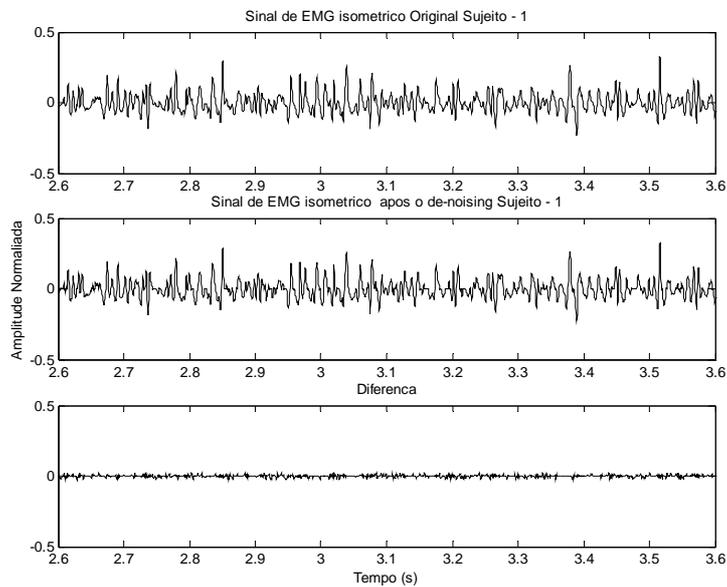


Figura A.1 – Resultado do *de-noising* de sinais eletrofisiológicos n° 1.

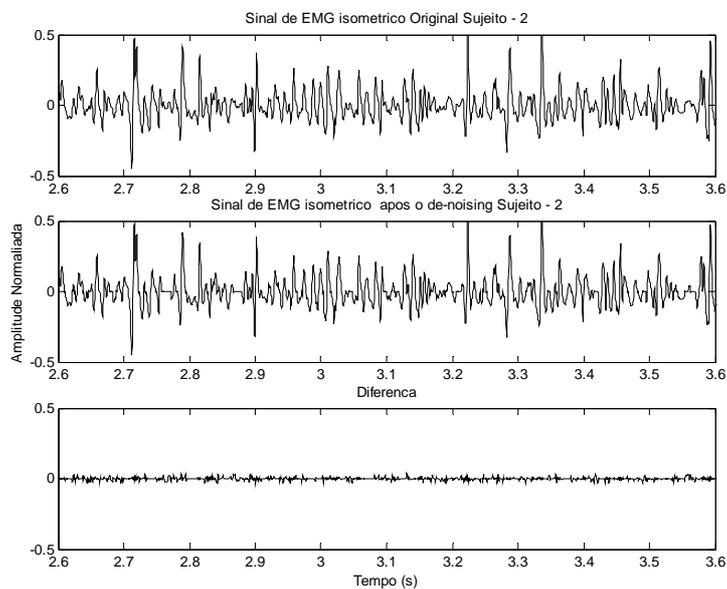


Figura A.2 – Resultado do *de-noising* de sinais eletrofisiológicos n° 2.

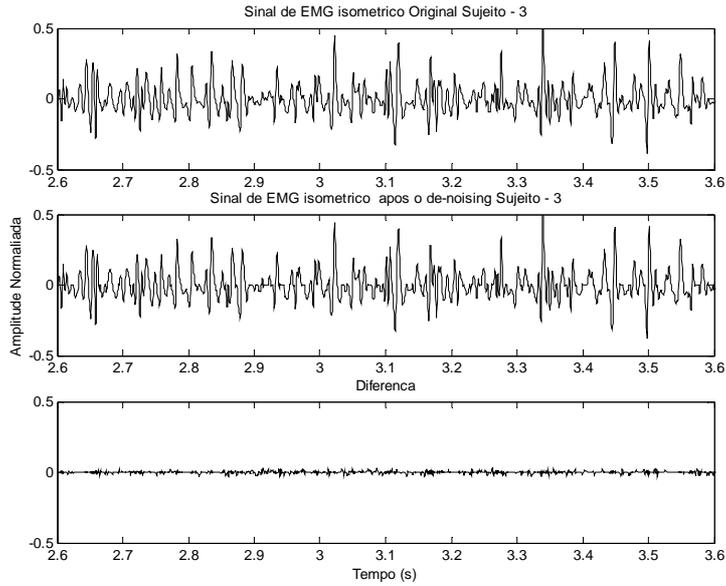


Figura A.3 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 3.

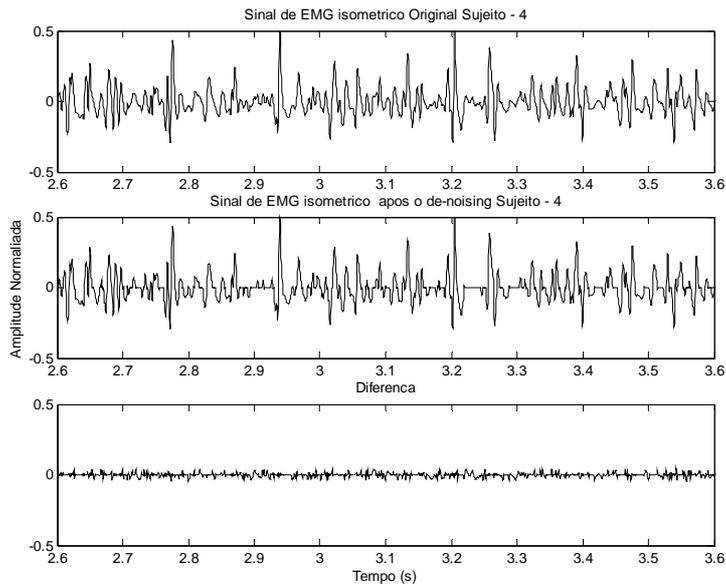


Figura A.4 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 4.

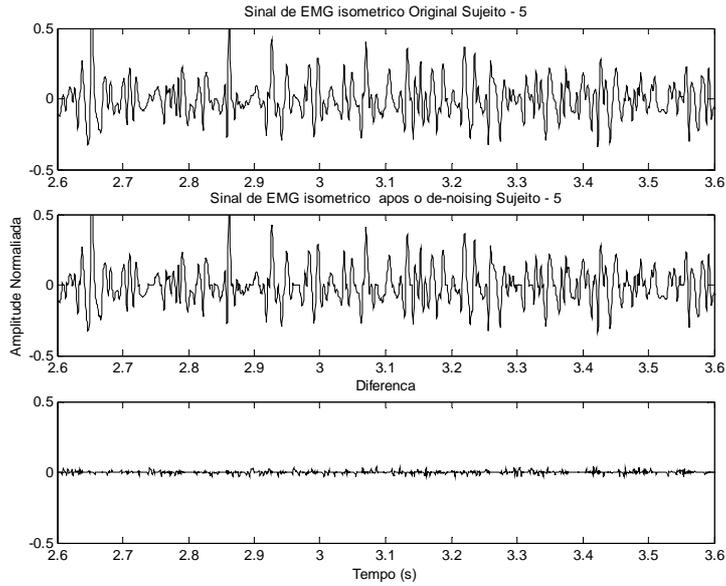


Figura A.5 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 5.

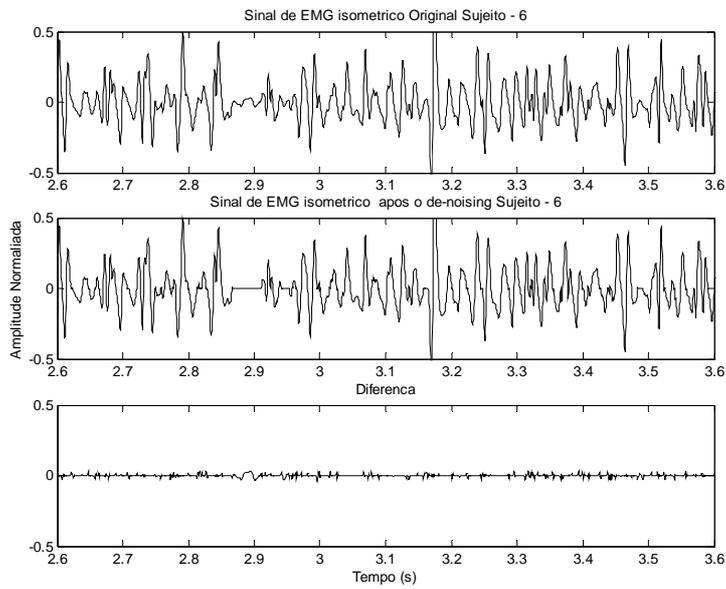


Figura A.6 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 6.

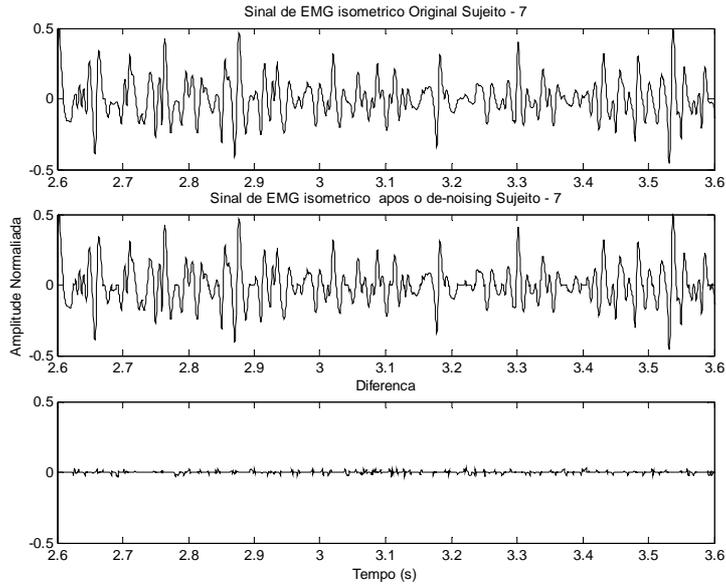


Figura A.7 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 7.

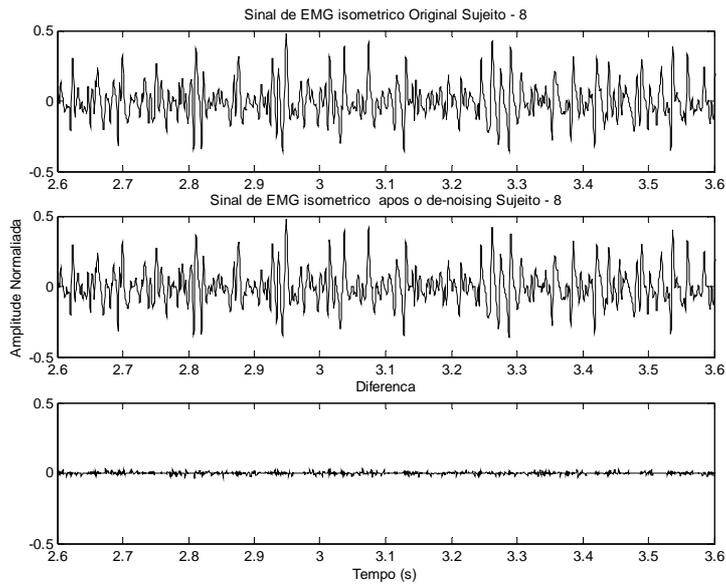


Figura A.8 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 8.

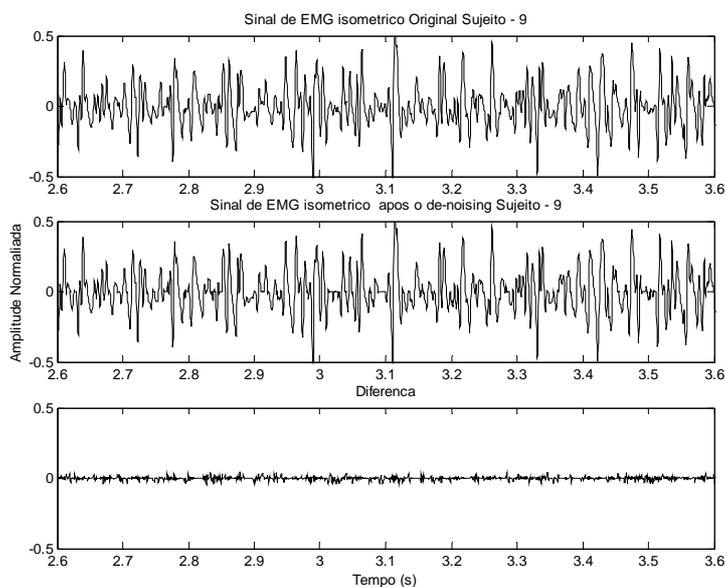


Figura A.9 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 9.

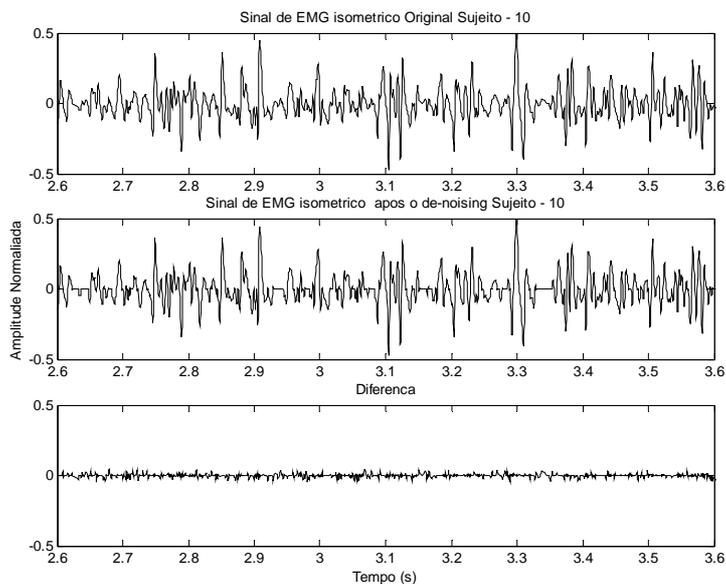


Figura A.10 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 10.

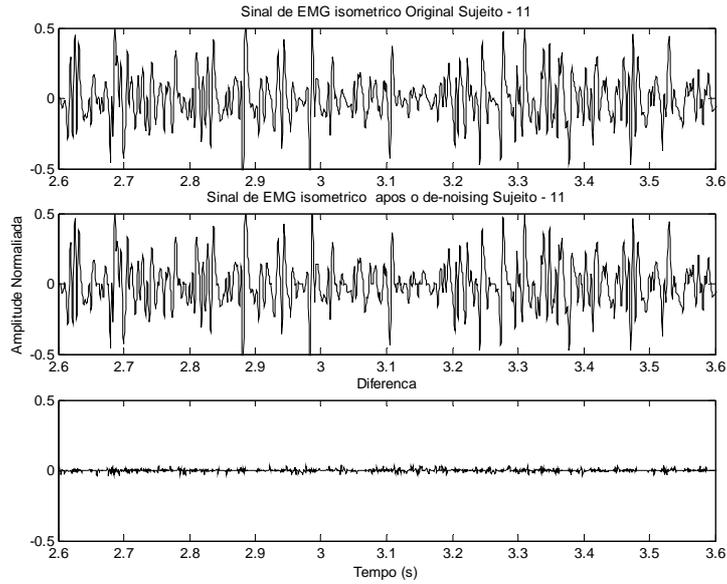


Figura A.11 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 11.

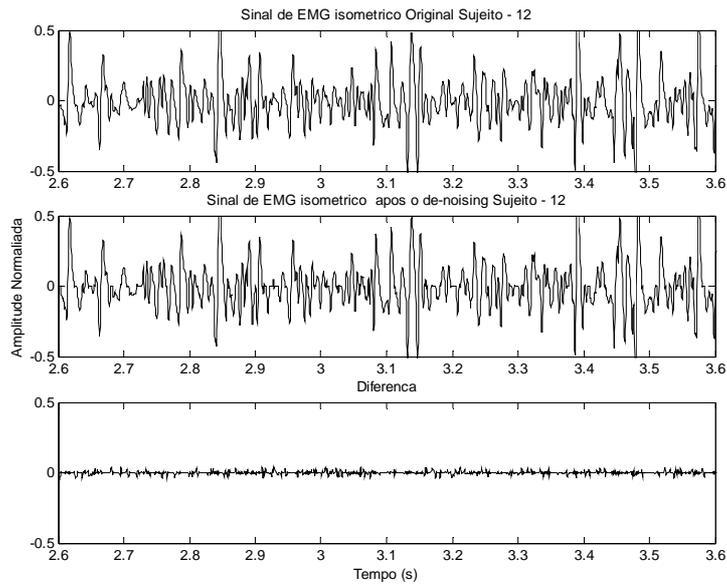


Figura A.12 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 12.

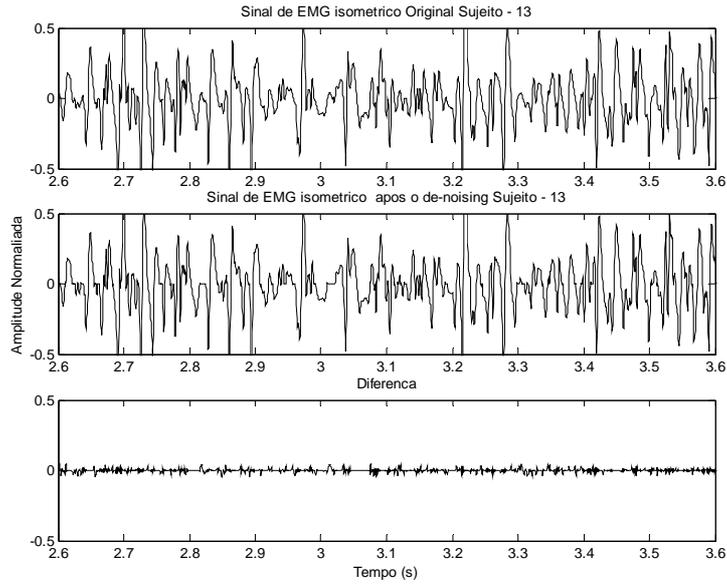


Figura A.13 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 13.

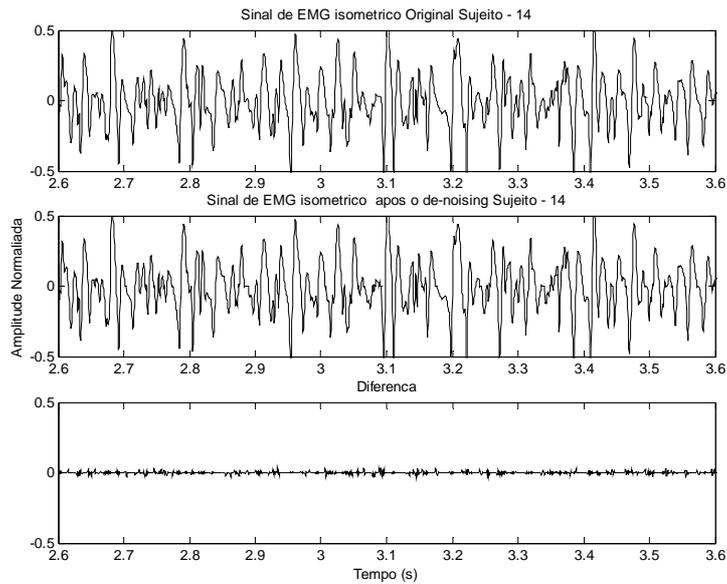


Figura A.14 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 14.

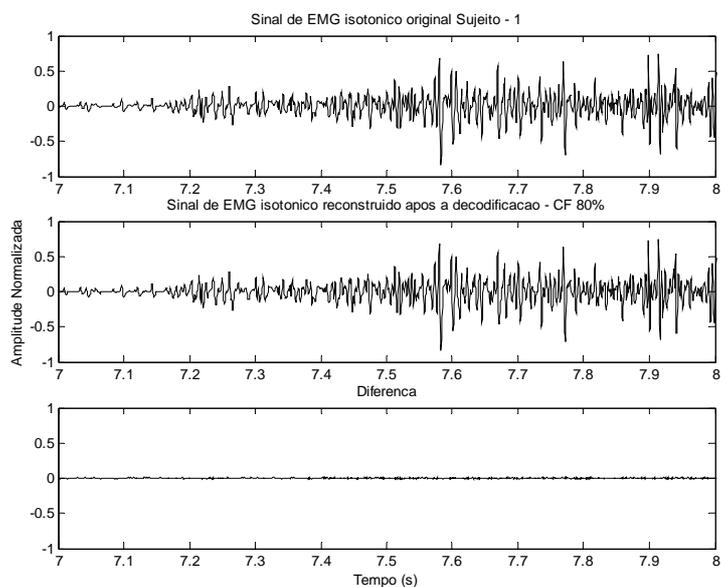


Figura A.15 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 15.

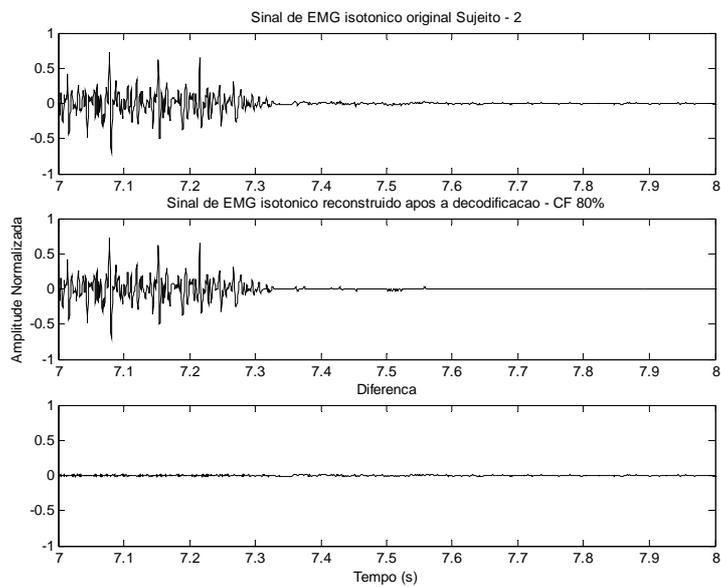


Figura A.16 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 16.

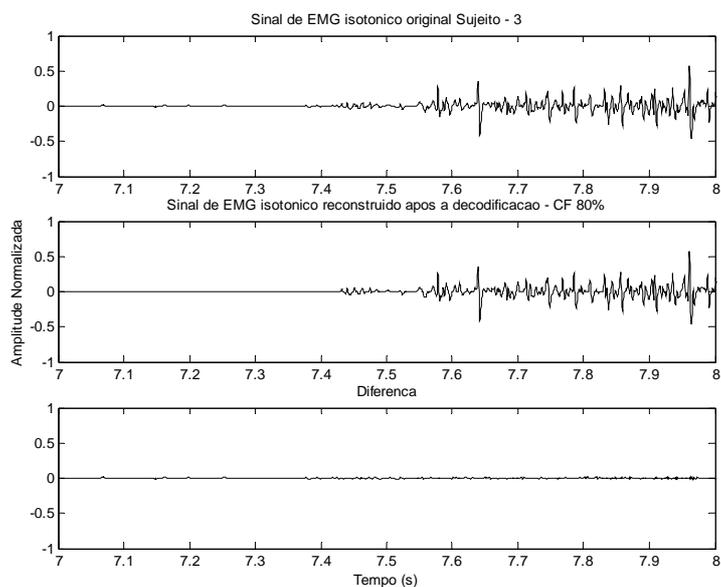


Figura A.17 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 17.

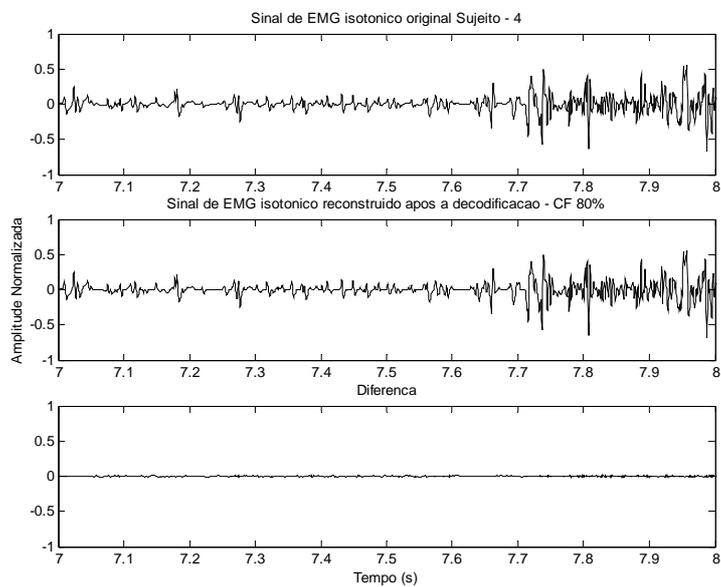


Figura A.18 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 18.

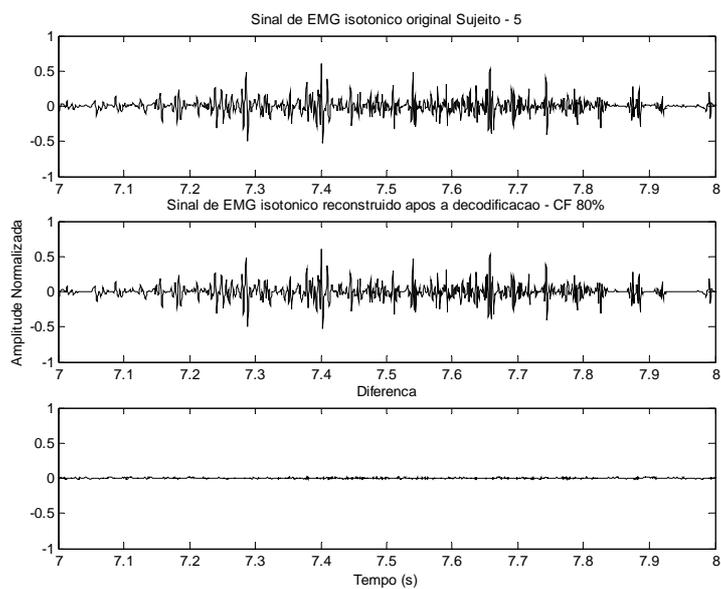


Figura A.19 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 19.

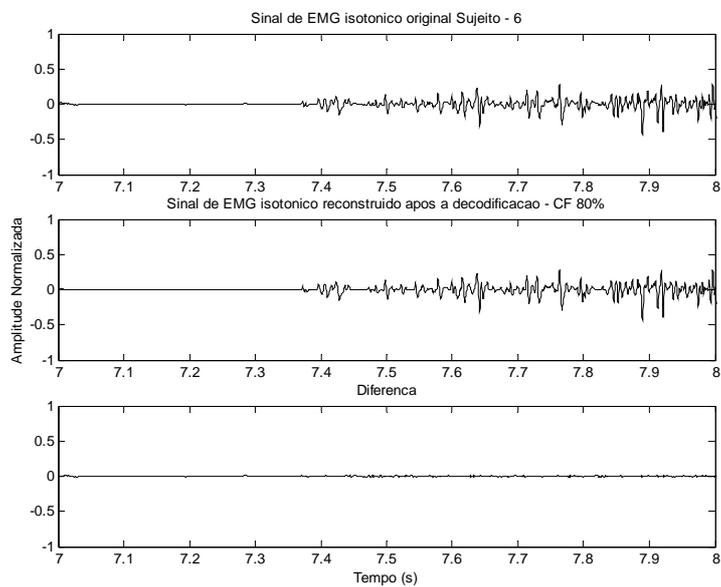


Figura A.20 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 20.

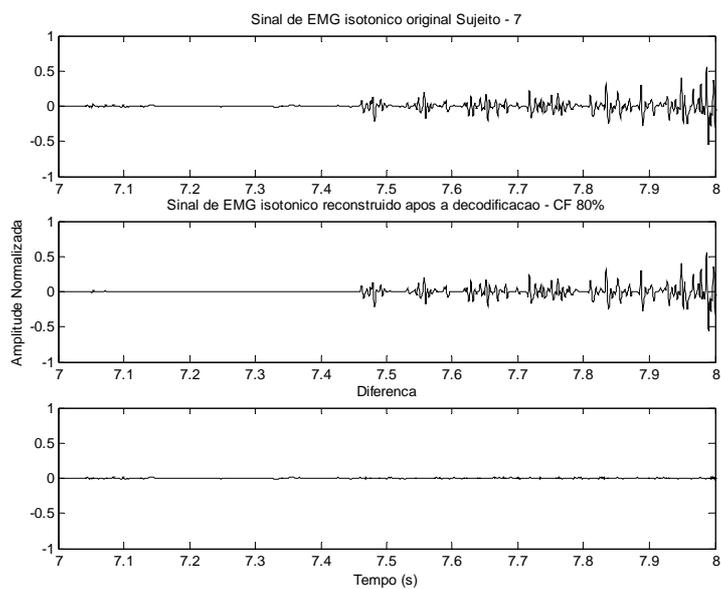


Figura A.21 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 21.

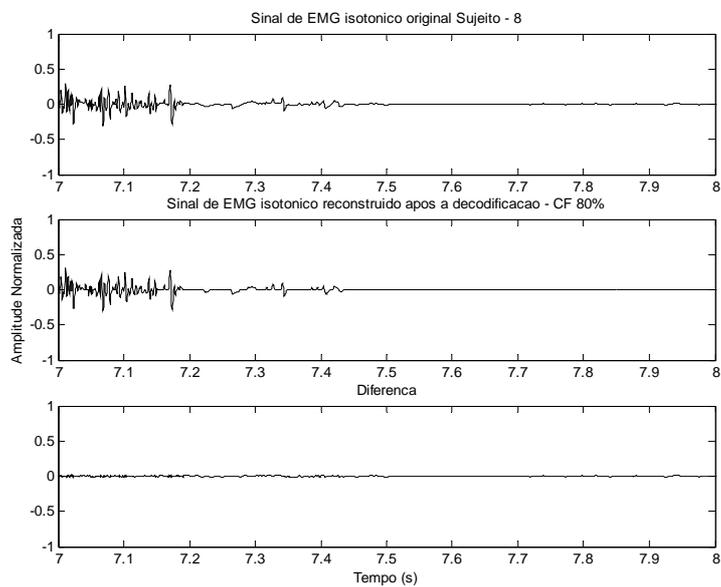


Figura A.22 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 22.

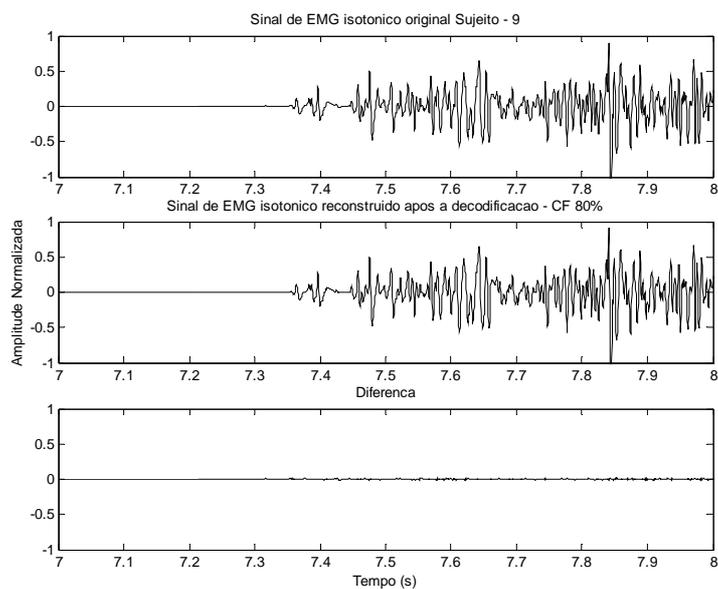


Figura A.23 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 23.

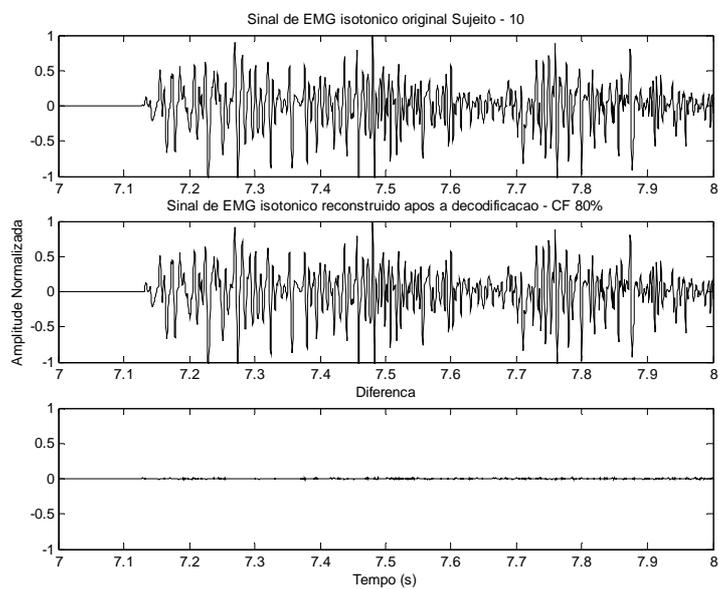


Figura A.24 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 24.

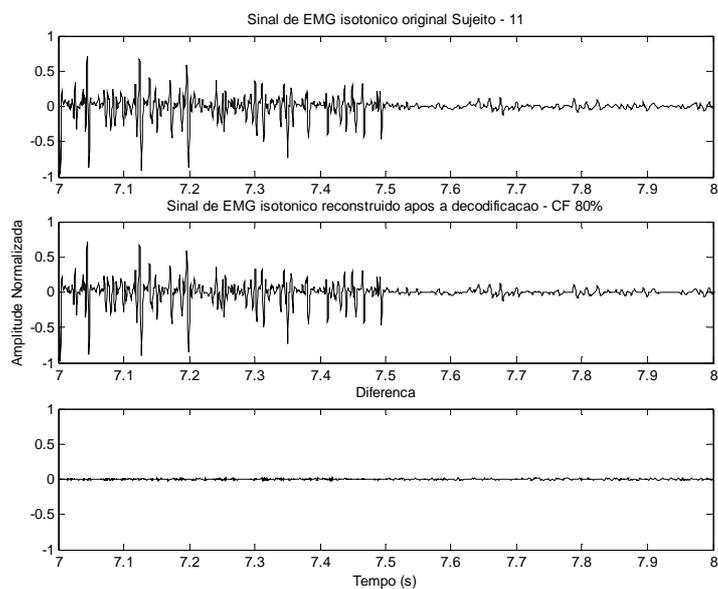


Figura A.25 – Resultado do *de-noising* de sinais eletrofisiológicos n° 25.

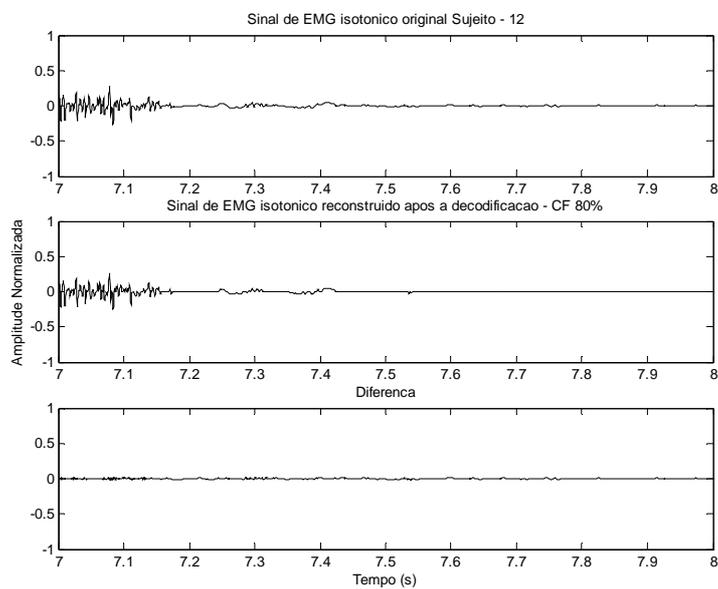


Figura A.26 – Resultado do *de-noising* de sinais eletrofisiológicos n° 26.

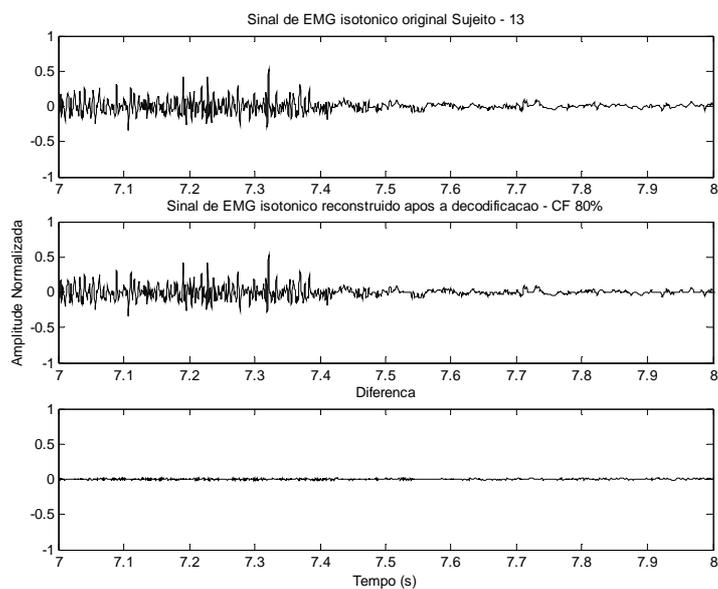


Figura A.27 – Resultado do *de-noising* de sinais eletrofisiológicos n° 27.

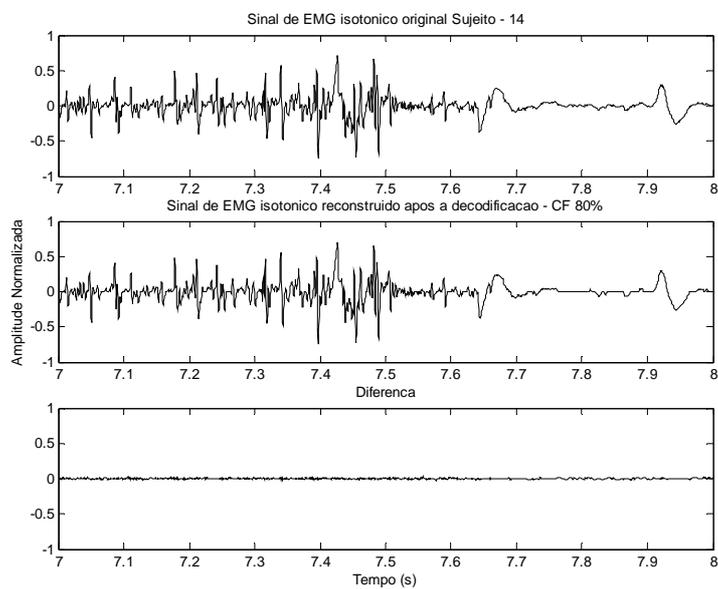


Figura A.28 – Resultado do *de-noising* de sinais eletrofisiológicos n° 28.

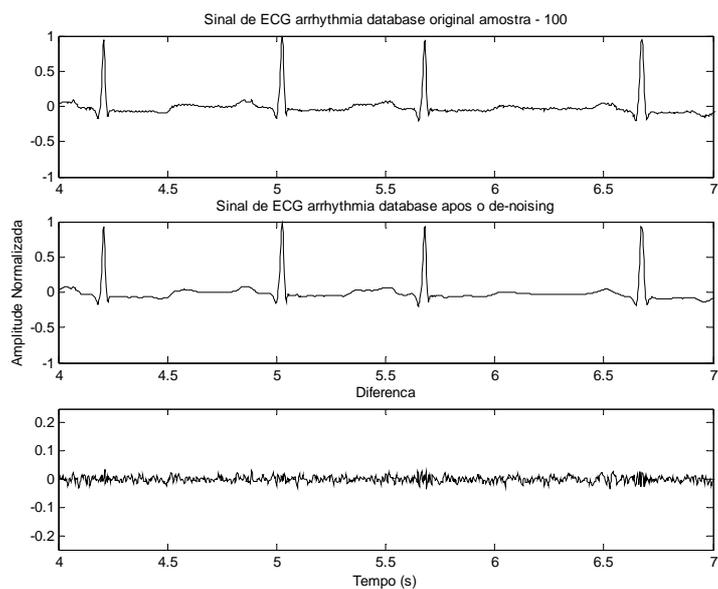


Figura A.29 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 29.

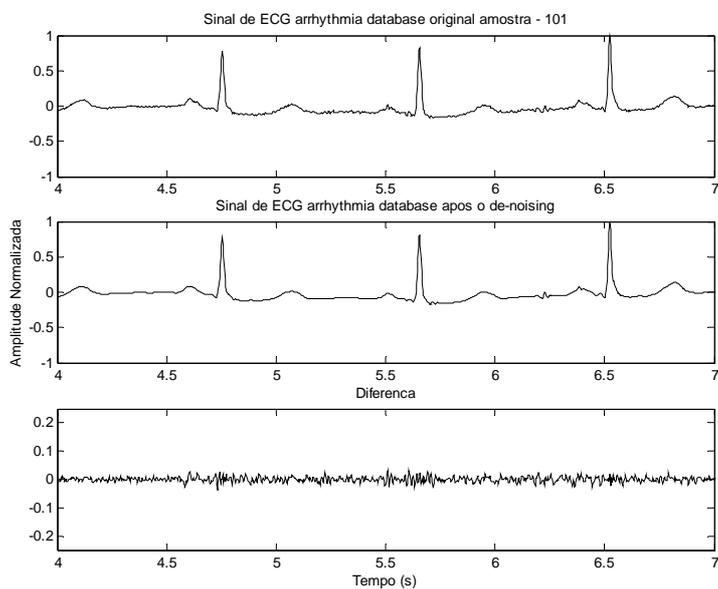


Figura A.30 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 30.

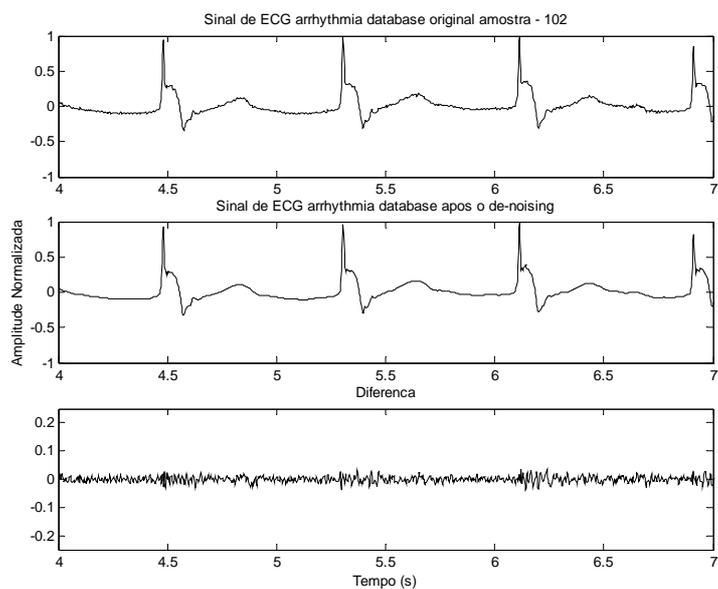


Figura A.31 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 31.

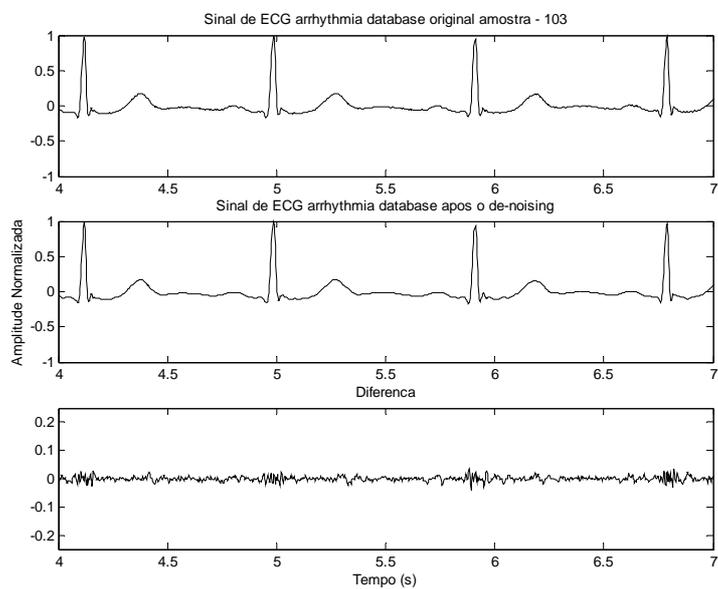


Figura A.32 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 32.

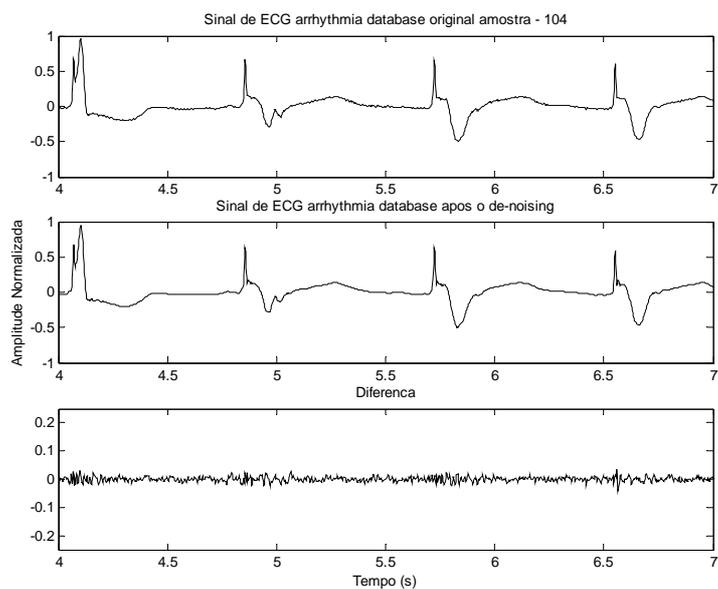


Figura A.33 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 33.

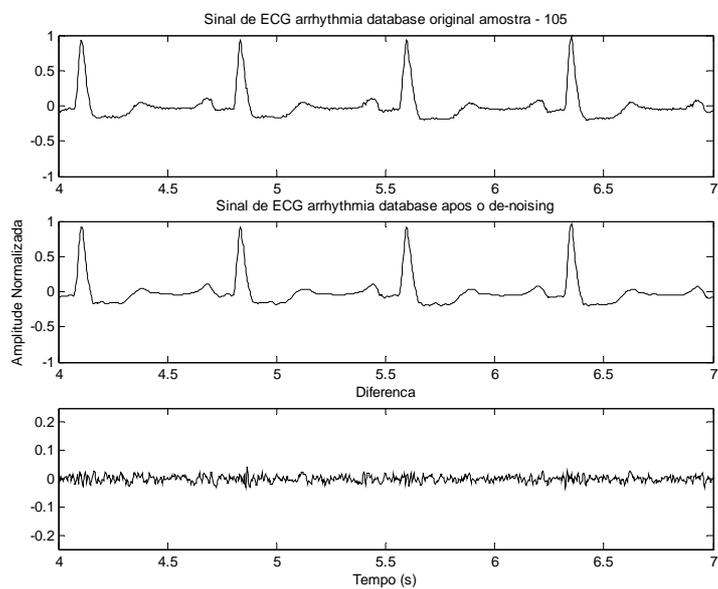


Figura A.34 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 34.

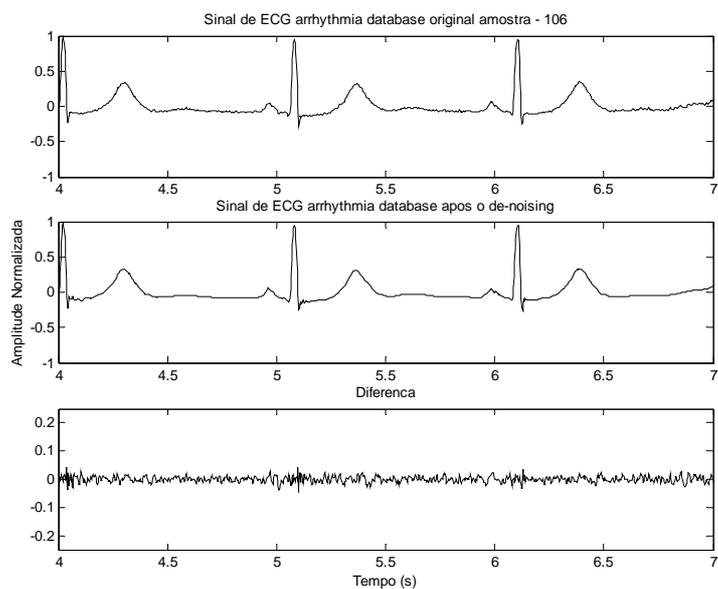


Figura A.35 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 35.

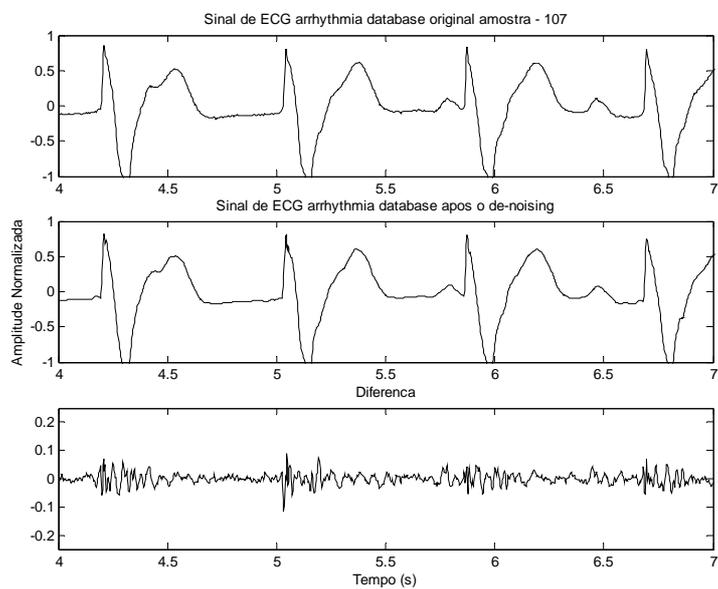


Figura A.36 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 36.

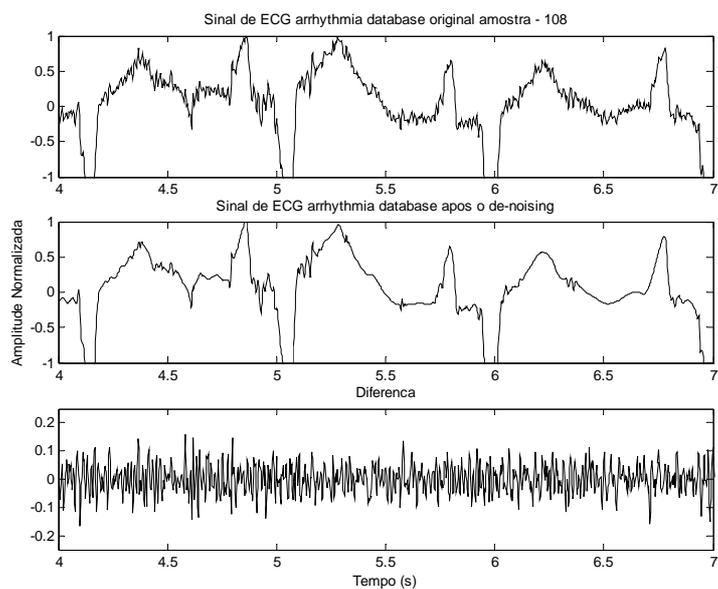


Figura A.37 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 37.

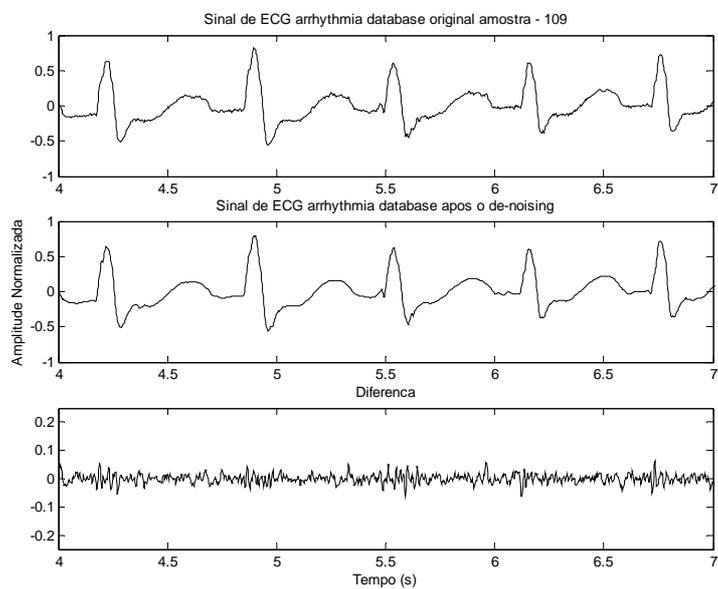


Figura A.38 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 38.

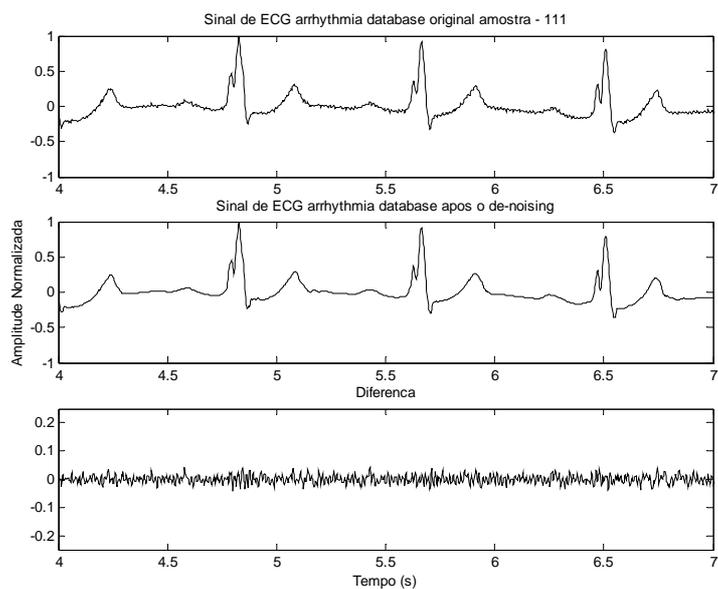


Figura A.39 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 39.

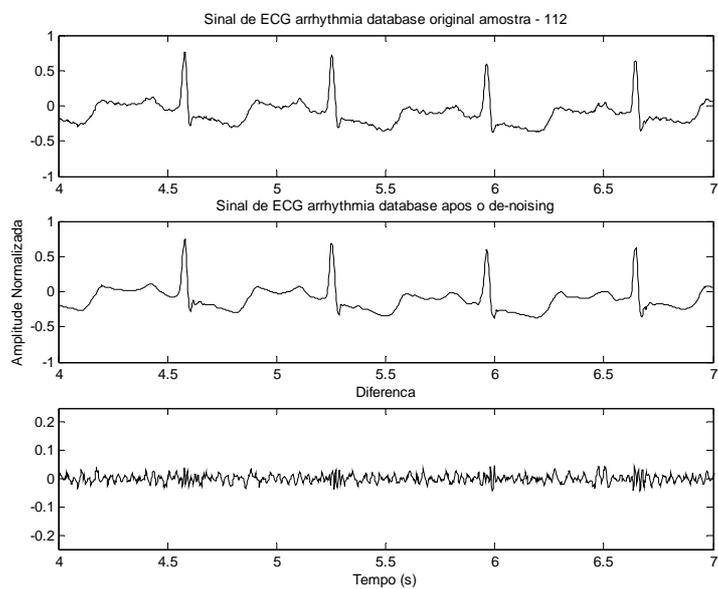


Figura A.40 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 40.

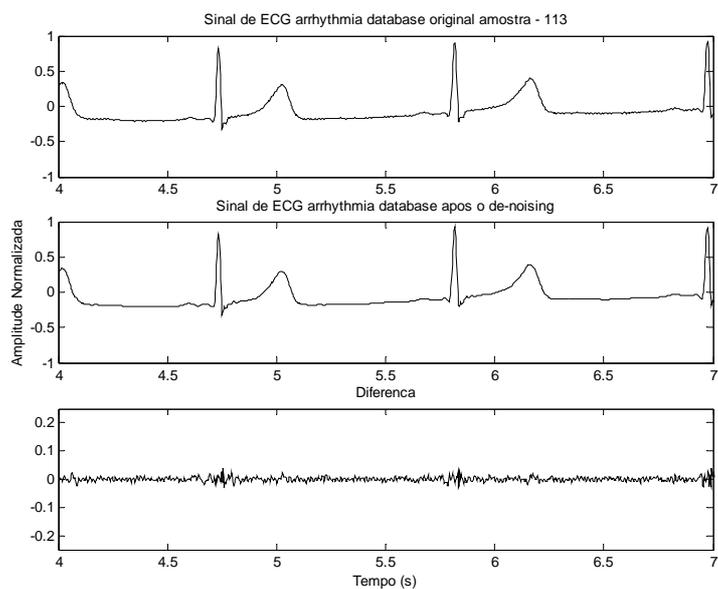


Figura A.41 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 41.

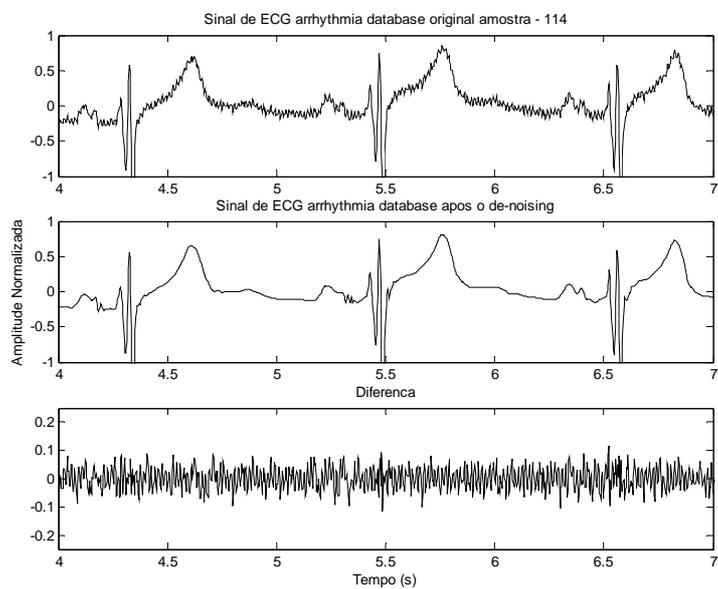


Figura A.42 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 42.

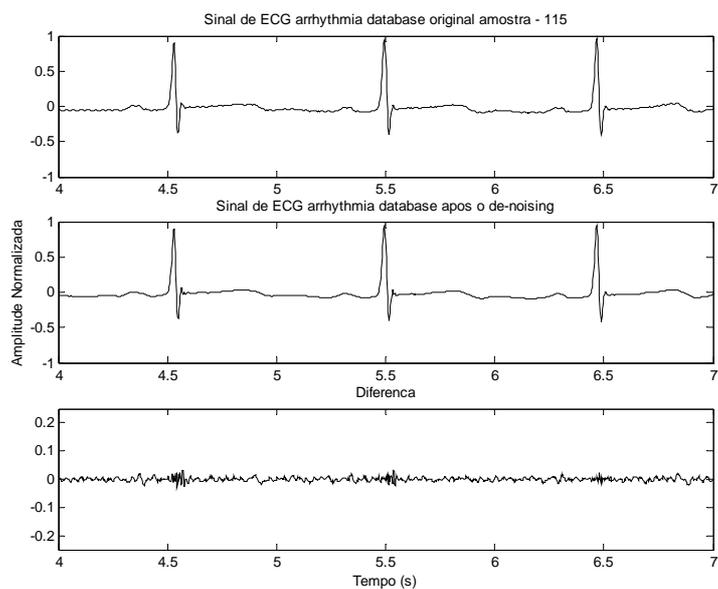


Figura A.43 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 43.

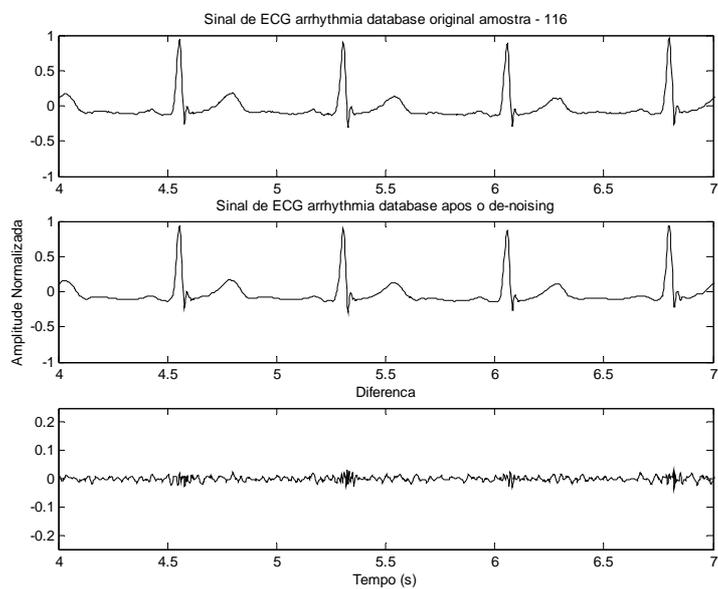


Figura A.44 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 44.

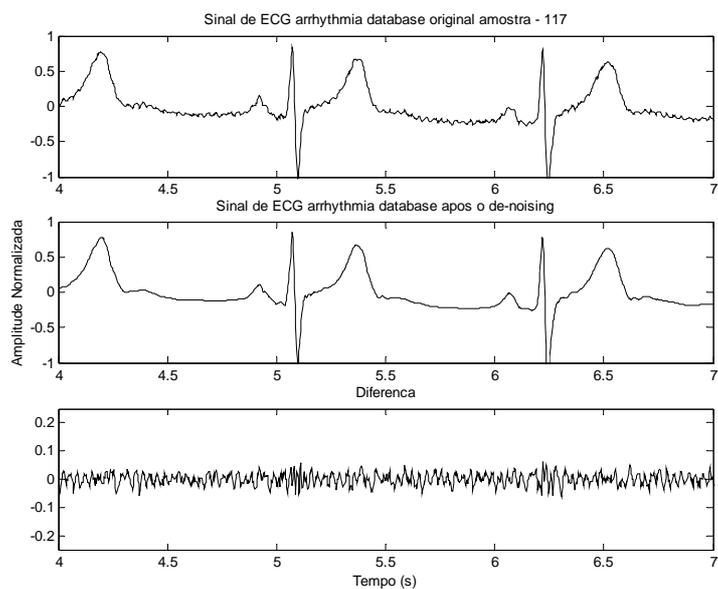


Figura A.45 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 45.

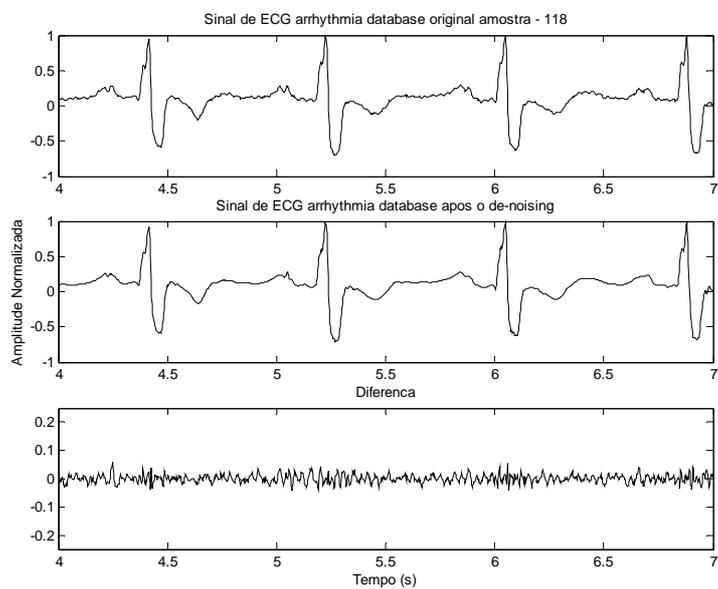


Figura A.46 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 46.

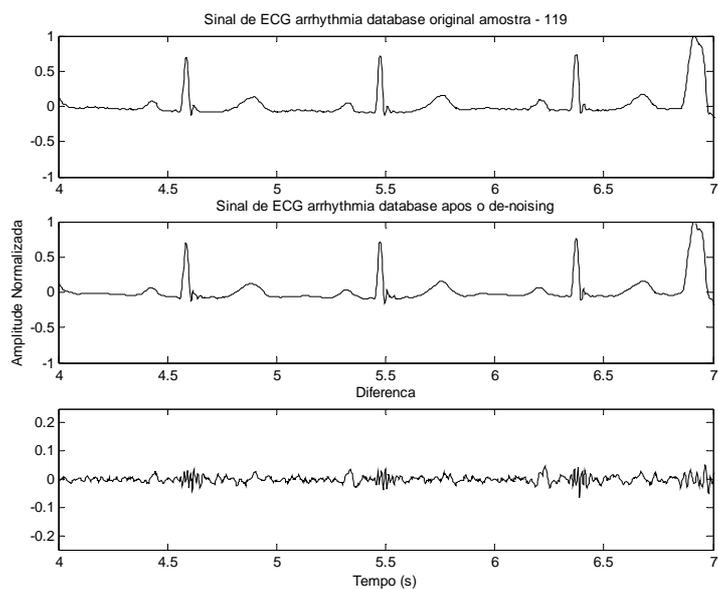


Figura A.47 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 47.

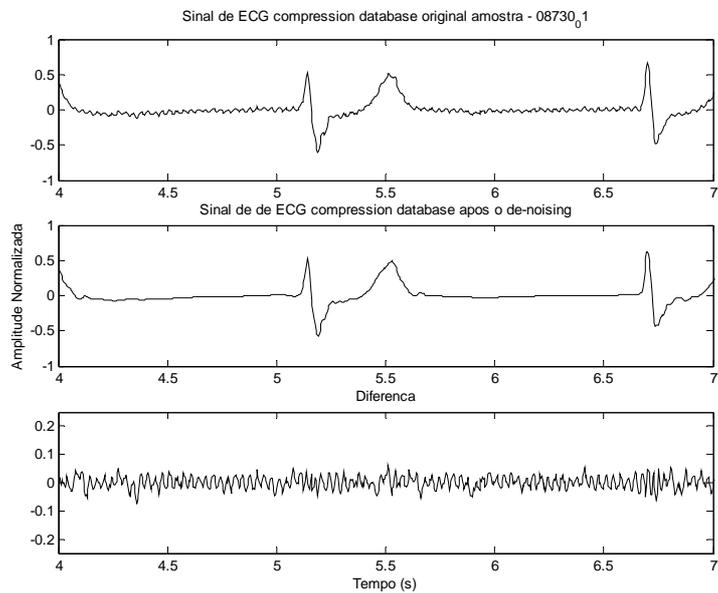


Figura A.48 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 48.

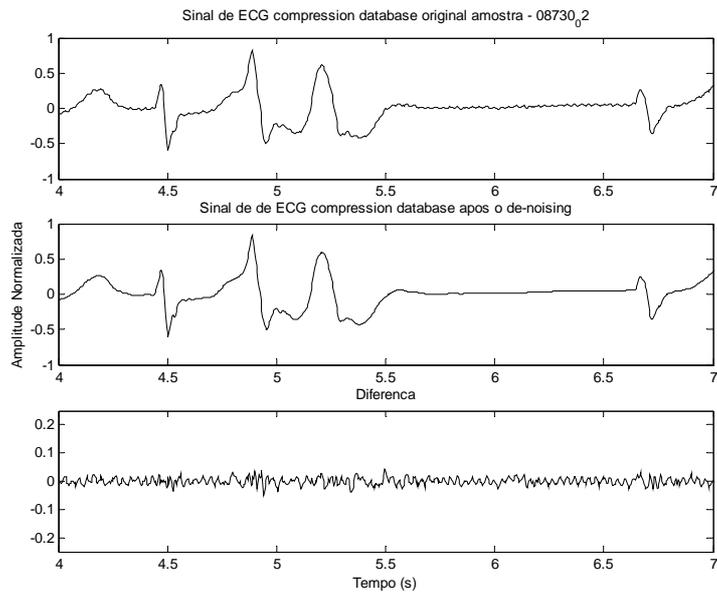


Figura A.49 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 49.

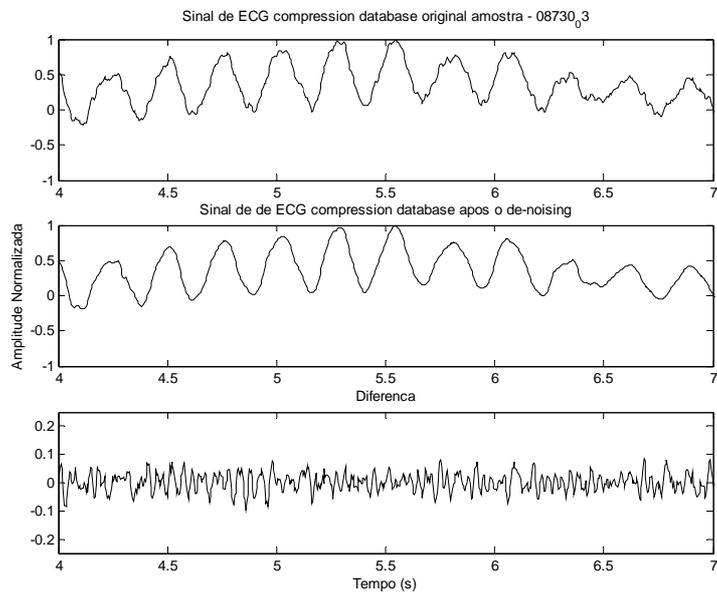


Figura A.50 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 50.

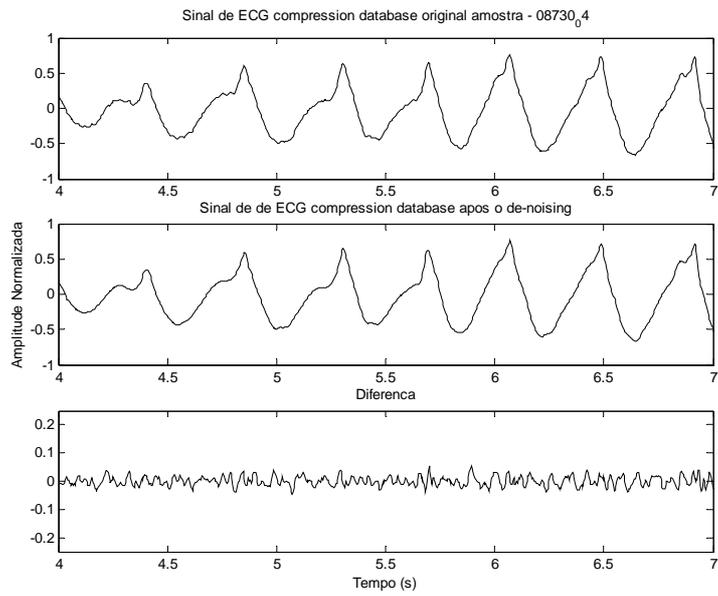


Figura A.51 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 51.

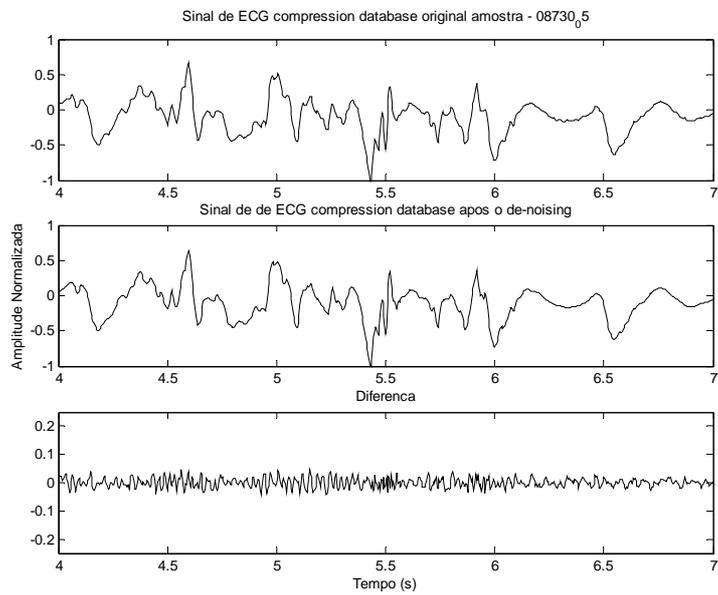


Figura A.52 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 52.

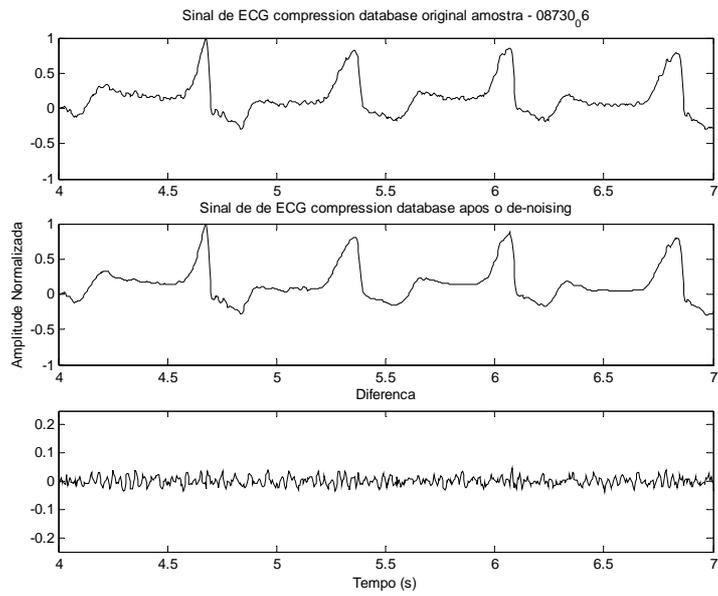


Figura A.53 – Resultado do *de-noising* de sinais eletrofisiológicos n<sup>o</sup> 53.

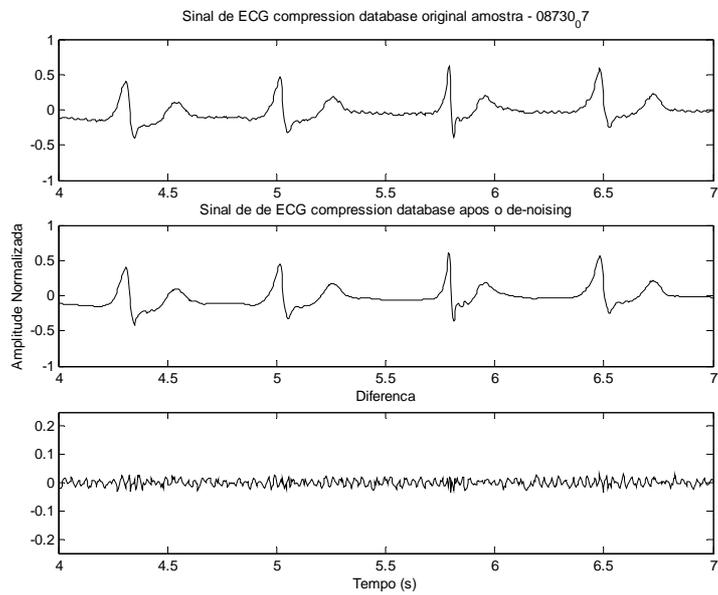


Figura A.54 – Resultado do *de-noising* de sinais eletrofisiológicos n<sup>o</sup> 54.

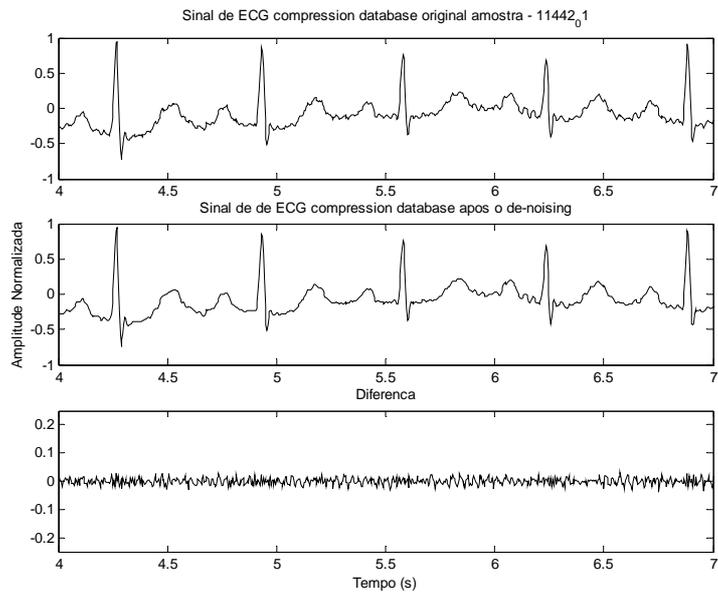


Figura A.55 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 55.

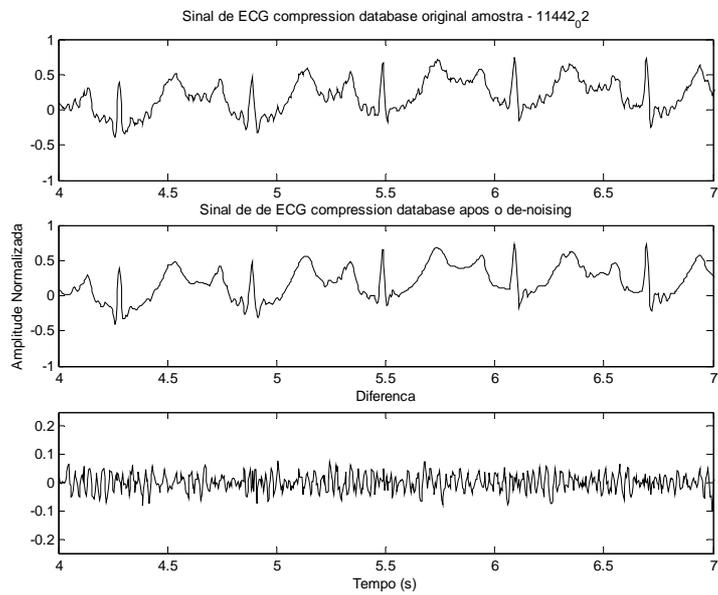


Figura A.56 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 56.

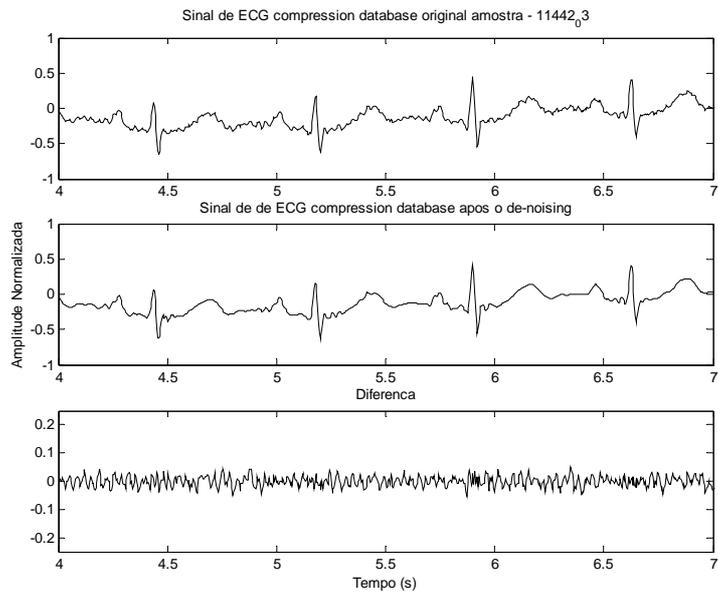


Figura A.57 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 57.

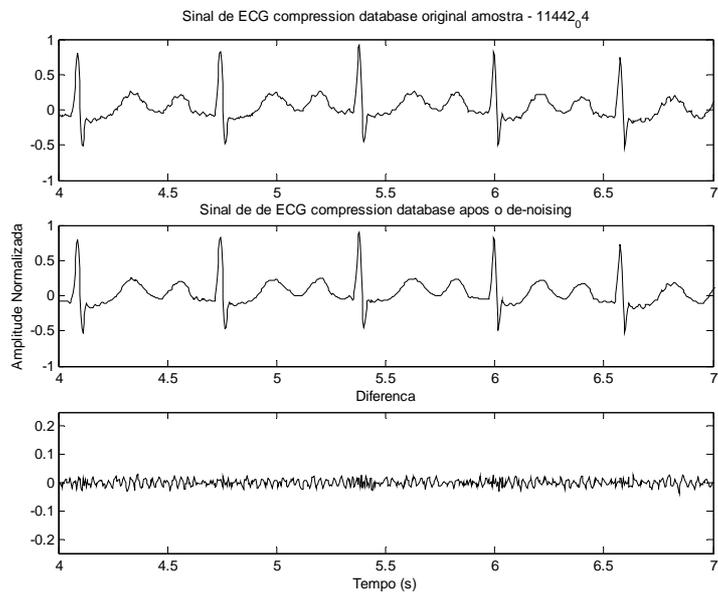


Figura A.58 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 58.

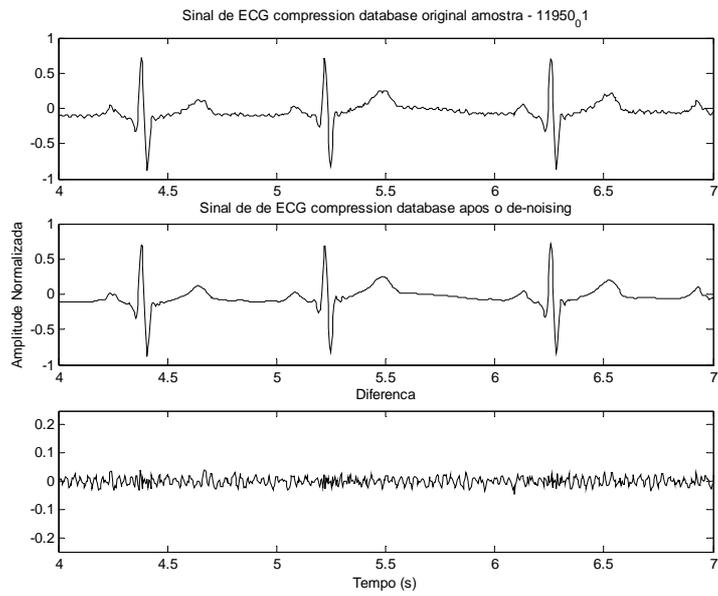


Figura A.59 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 59.

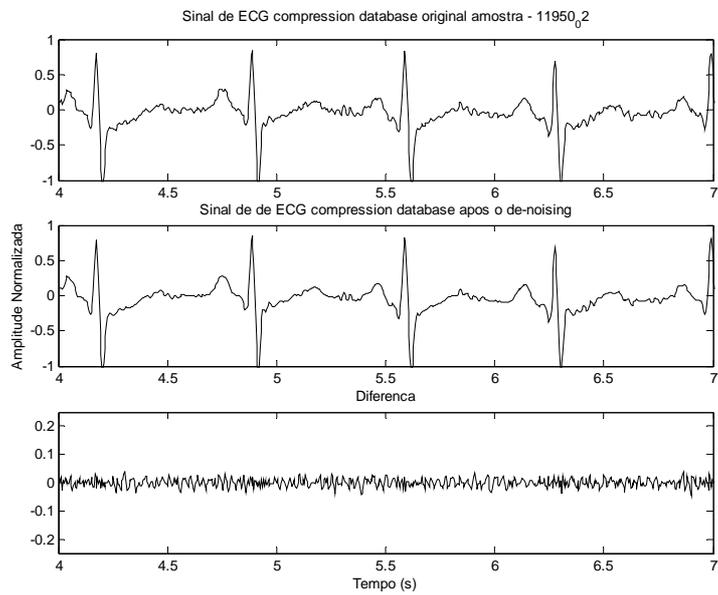


Figura A.60 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 60.

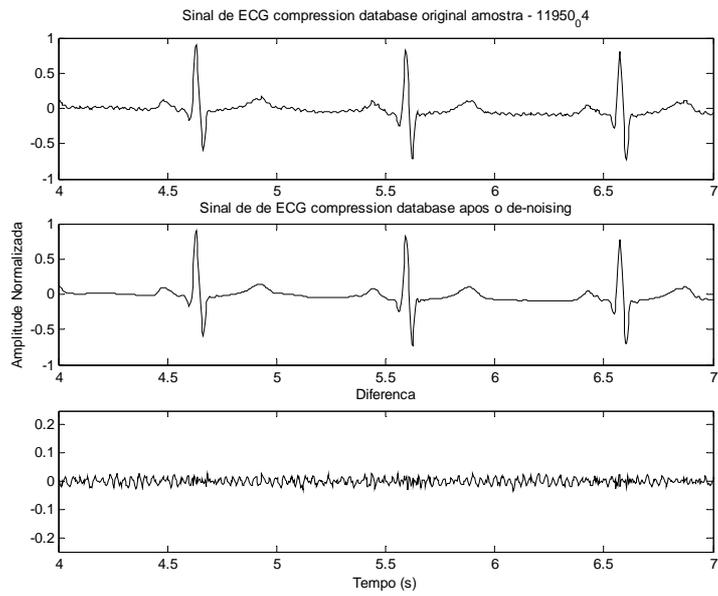


Figura A.61 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 61.

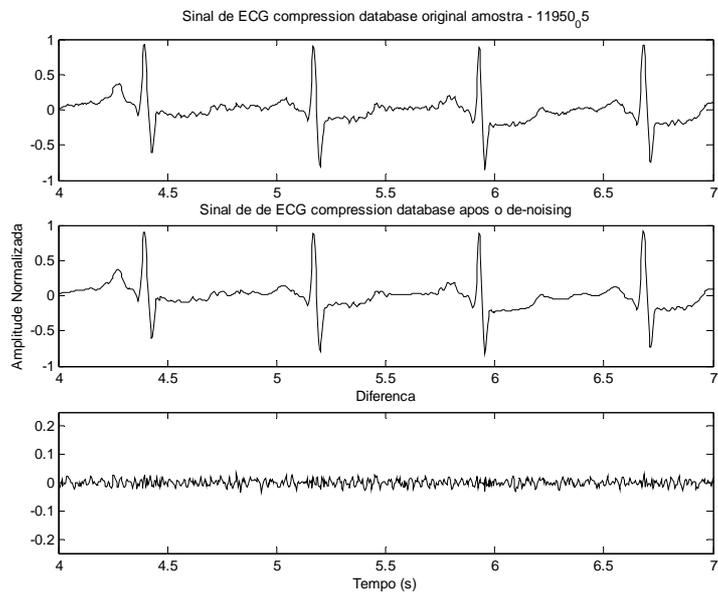


Figura A.62 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 62.

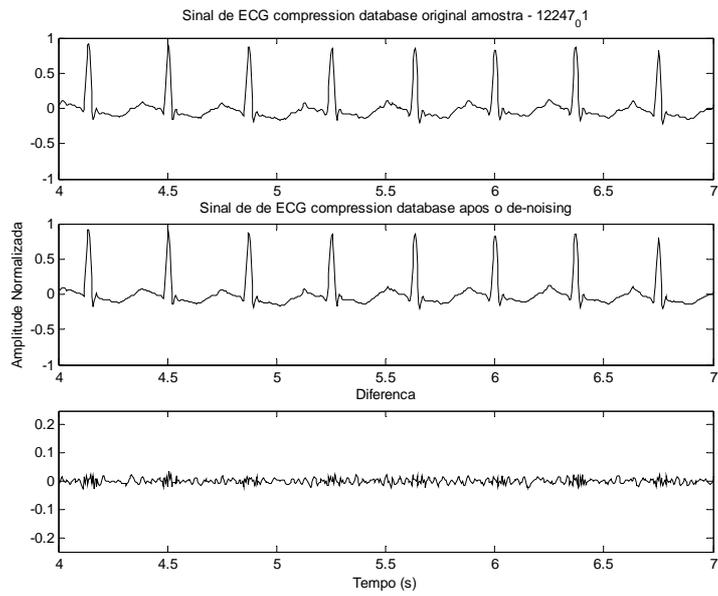


Figura A.63 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 63.

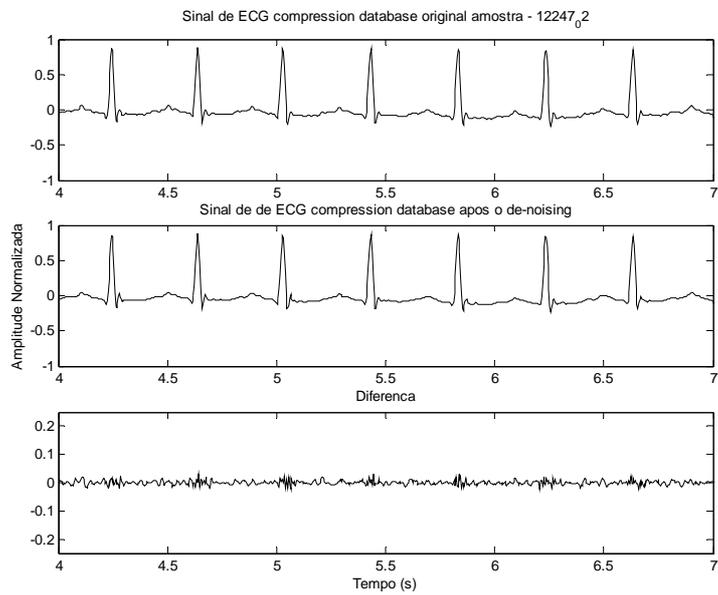


Figura A.64 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 64.

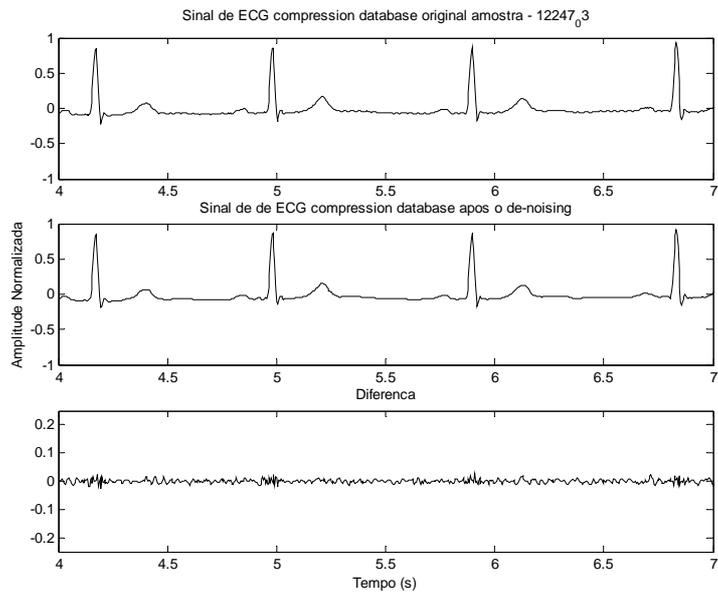


Figura A.65 – Resultado do *de-noising* de sinais eletrofisiológicos n<sup>o</sup> 65.

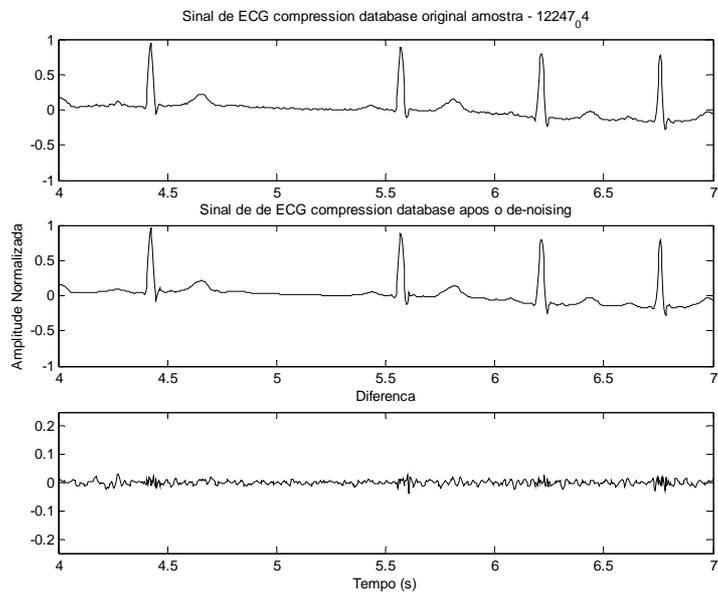


Figura A.66 – Resultado do *de-noising* de sinais eletrofisiológicos n<sup>o</sup> 66.

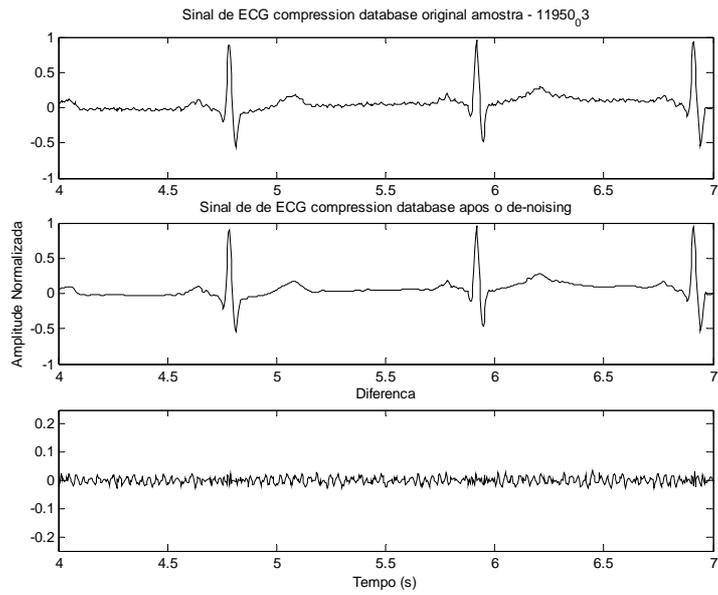


Figura A.67 – Resultado do *de-noising* de sinais eletrofisiológicos n ° 67.

## B - RESULTADOS DA COMPRESSÃO DE SINAIS ELETROFISIOLÓGICOS SEM PRÉ-PROCESSAMENTO COM *DE-NOISING*

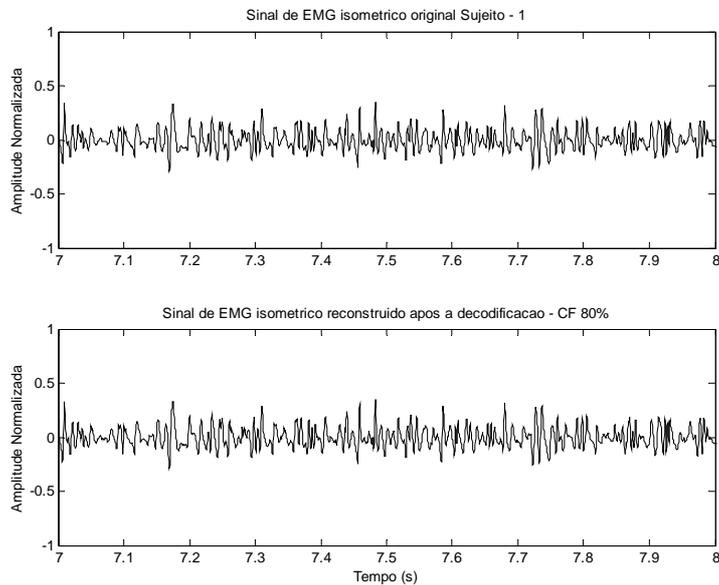


Figura B.1 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 1.

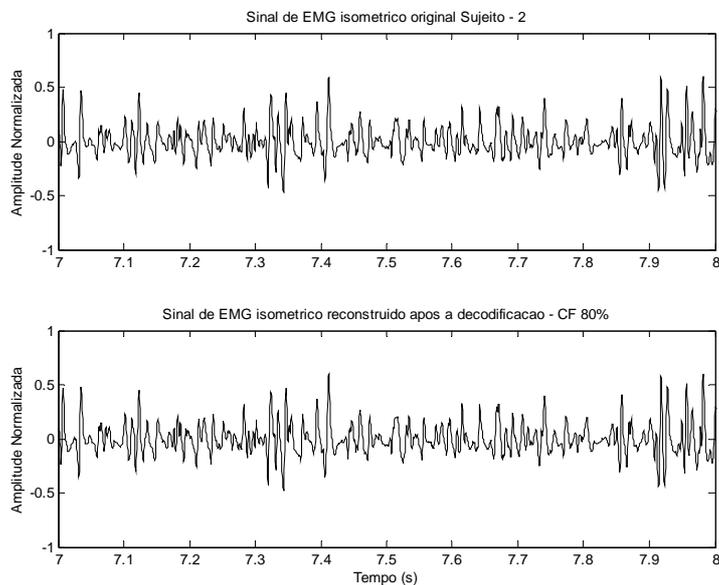


Figura B.2 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 2.

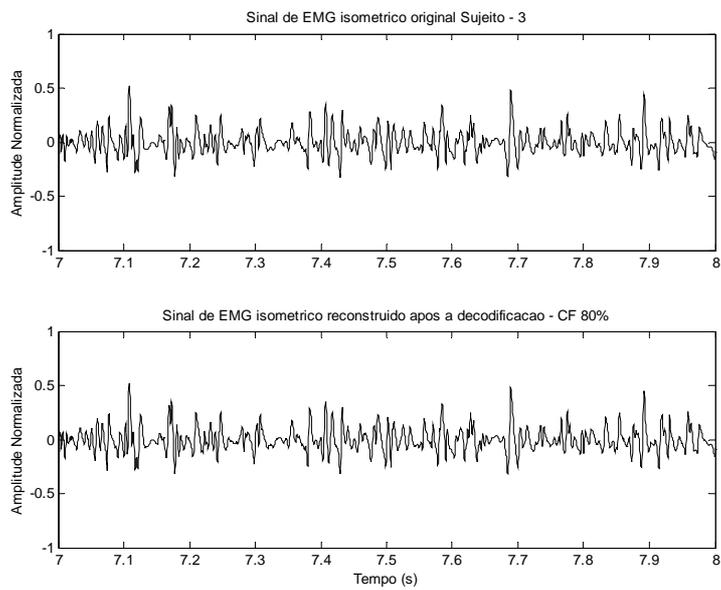


Figura B.3 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 3.

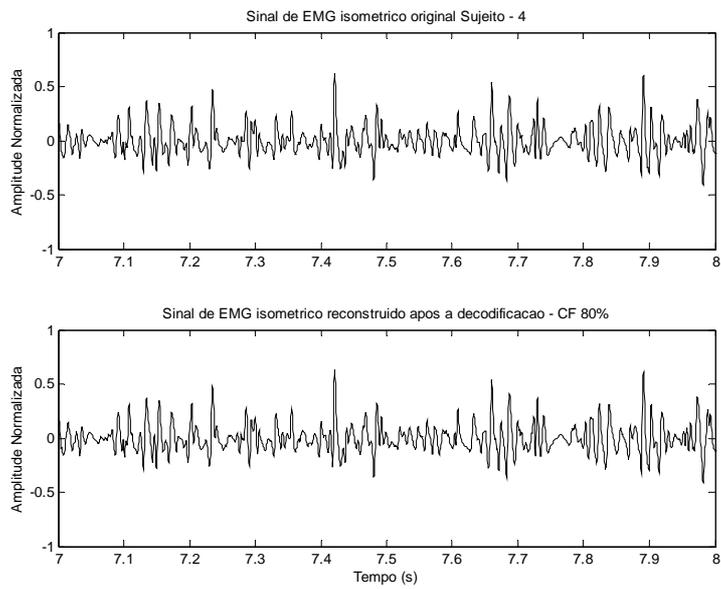


Figura B.4 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 4.

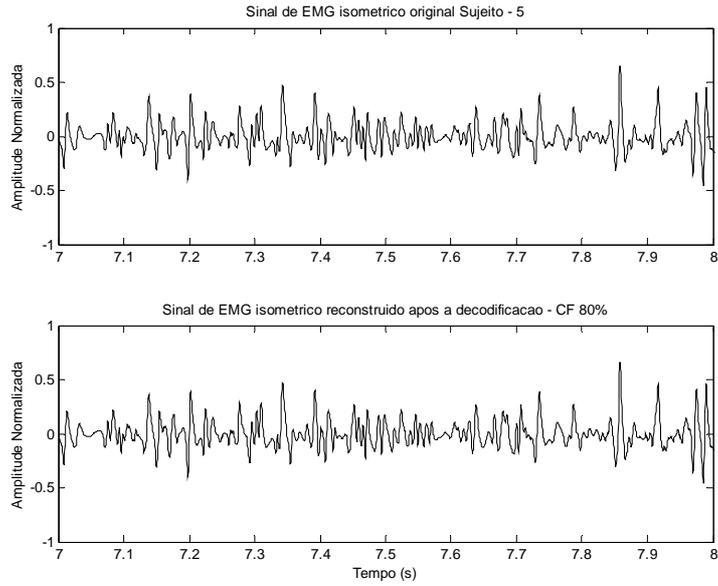


Figura B.5 – Resultado da compressão de sinais de EMG isométricos n ° 5.

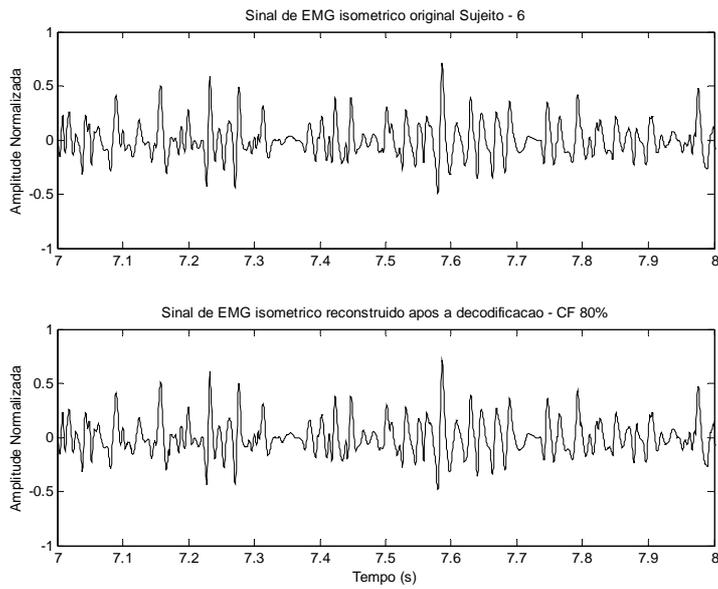


Figura B.6 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 6.

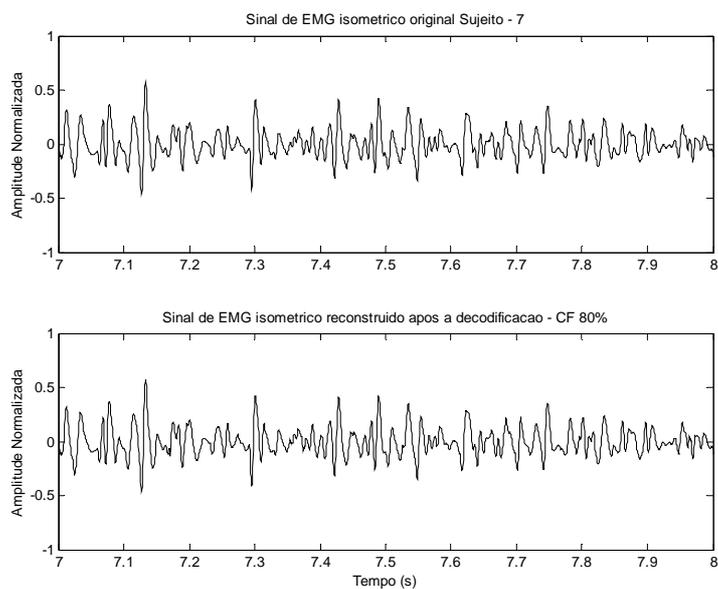


Figura B.7 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 7.

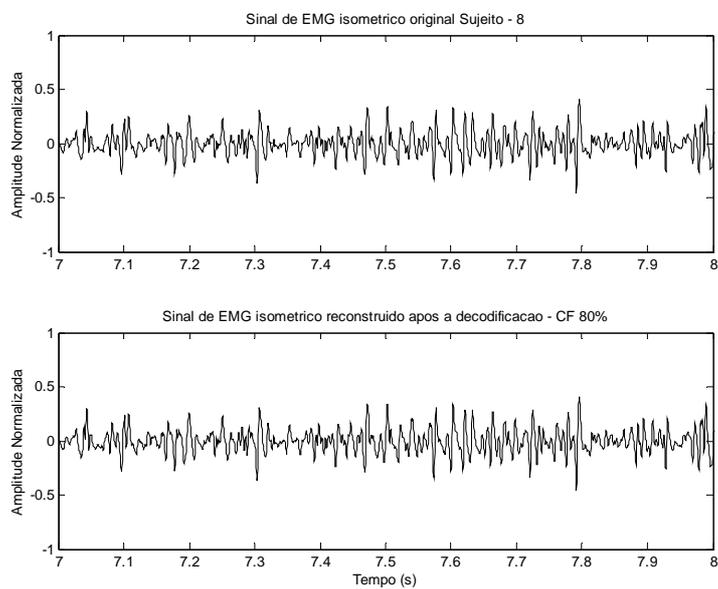


Figura B.8 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 8.

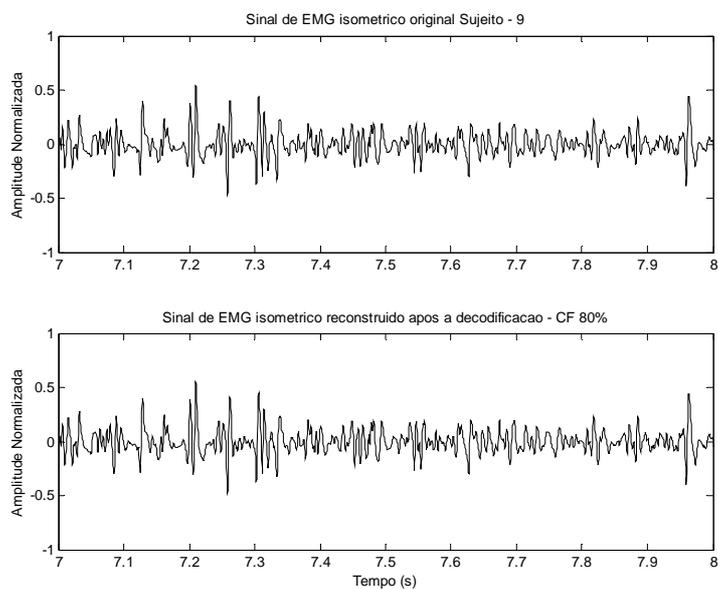


Figura B.9 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 9.

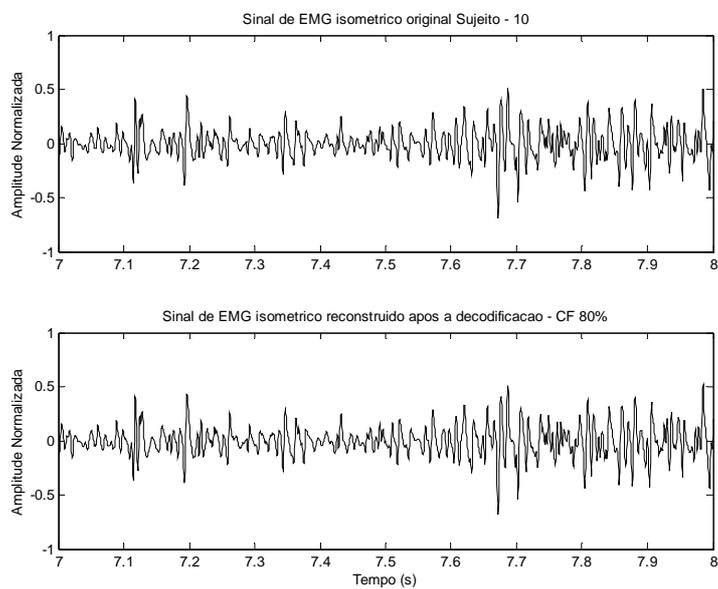


Figura B.10 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 10.

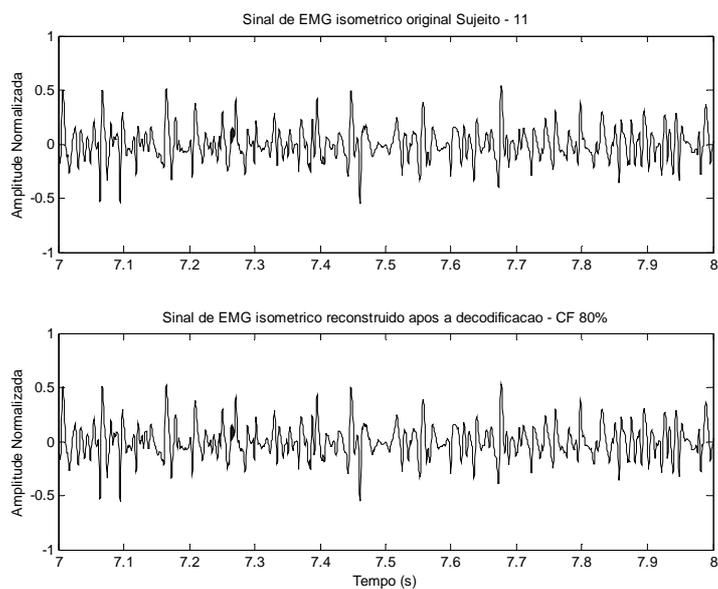


Figura B.11 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 11.

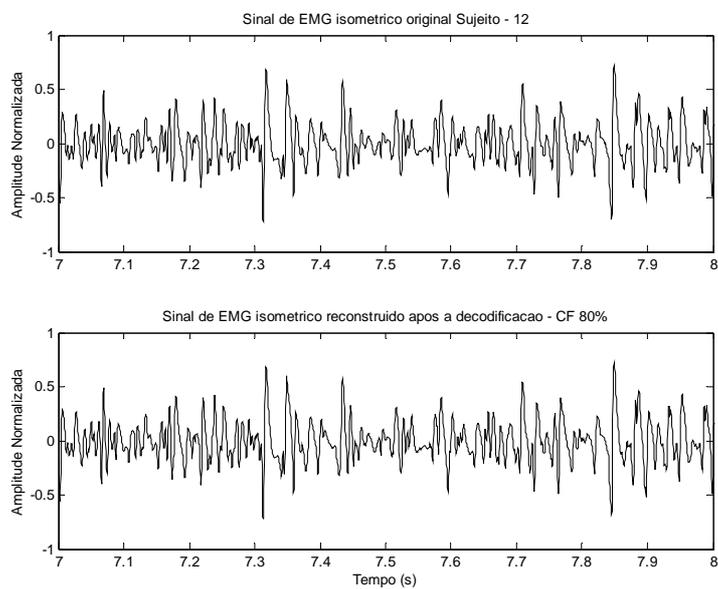


Figura B.12 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 12.

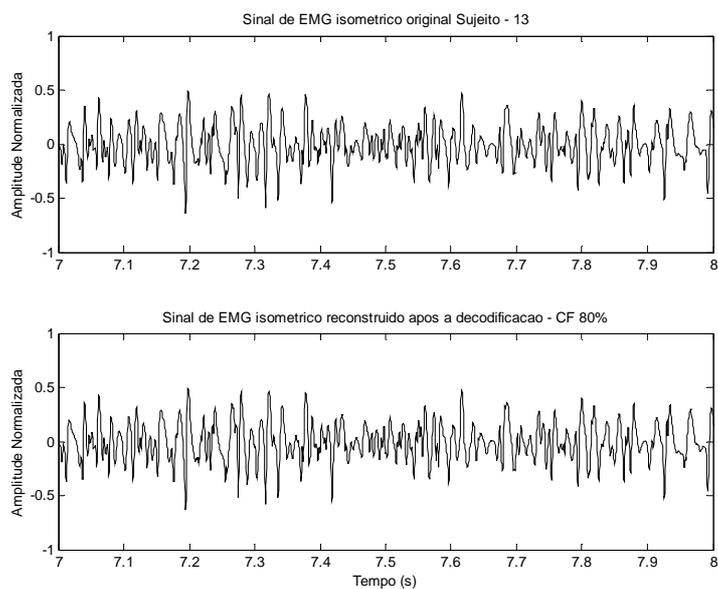


Figura B.13 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 13.

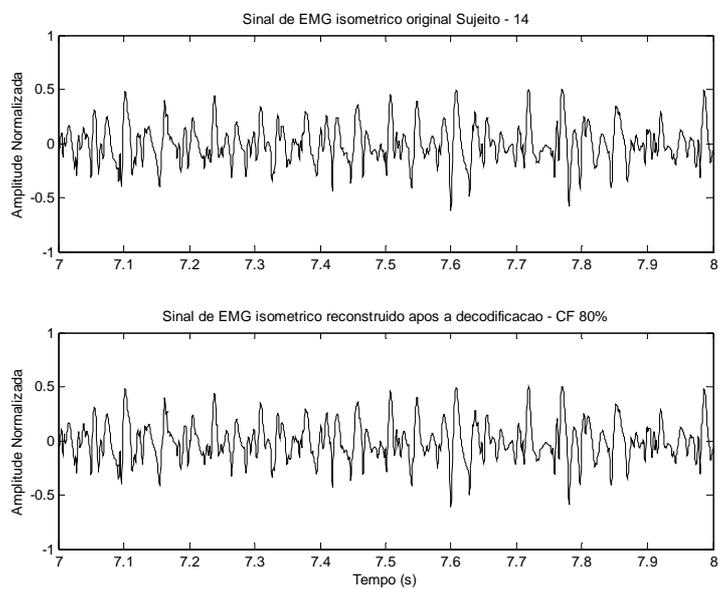


Figura B.14 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 14.

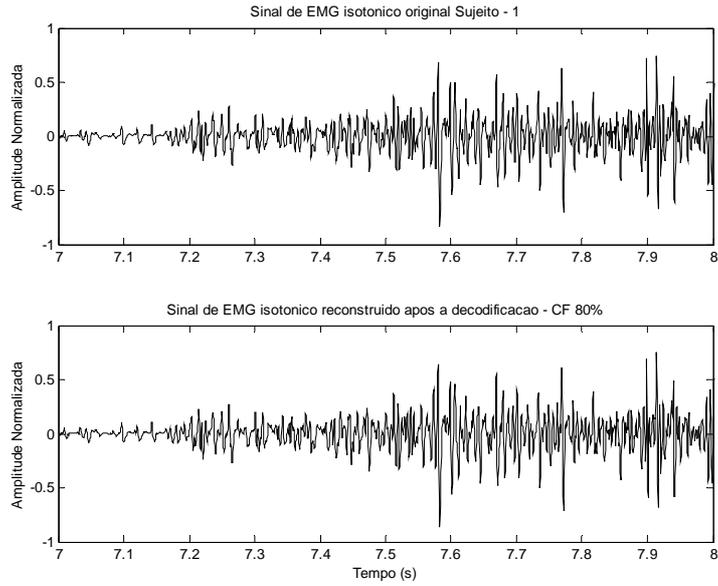


Figura B.15 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 15.

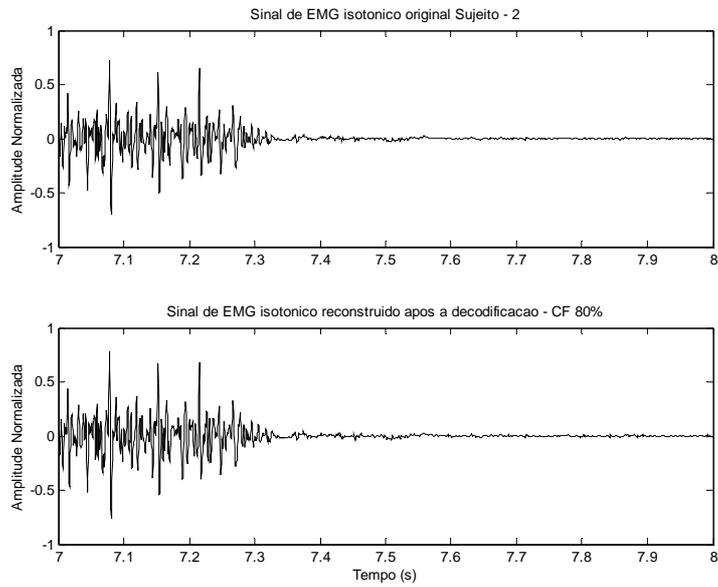


Figura B.16 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 16.

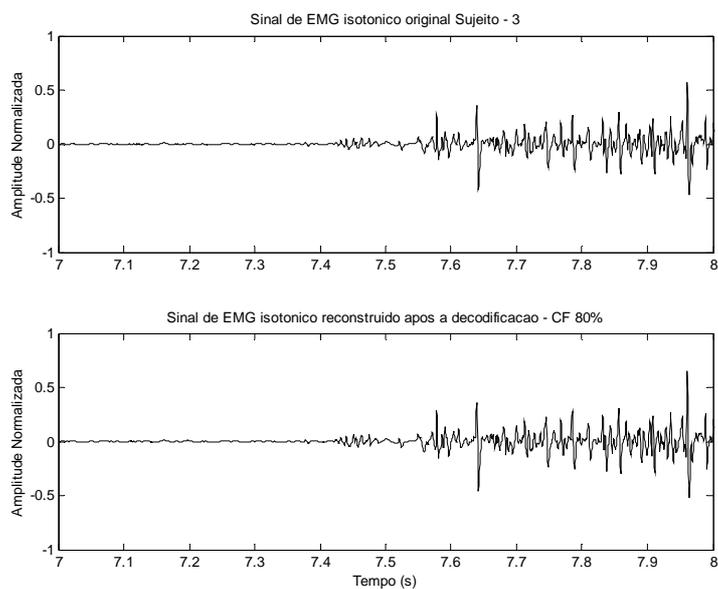


Figura B.17 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 17.

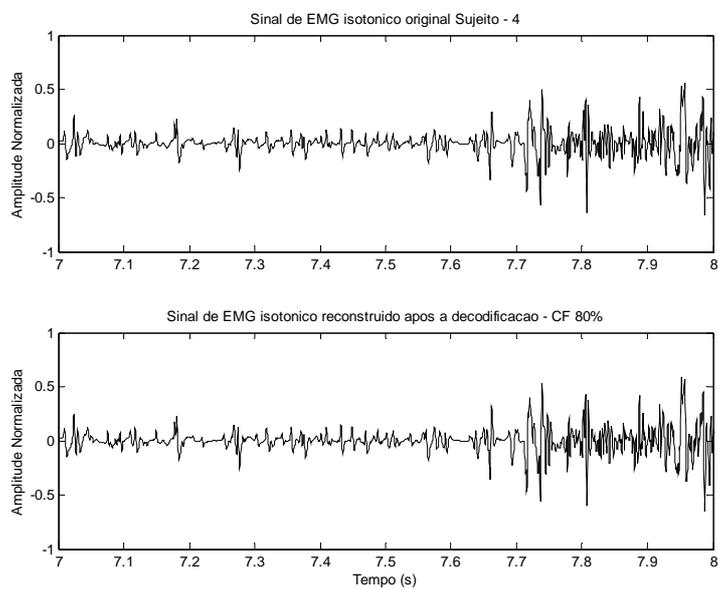


Figura B.18 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 18.

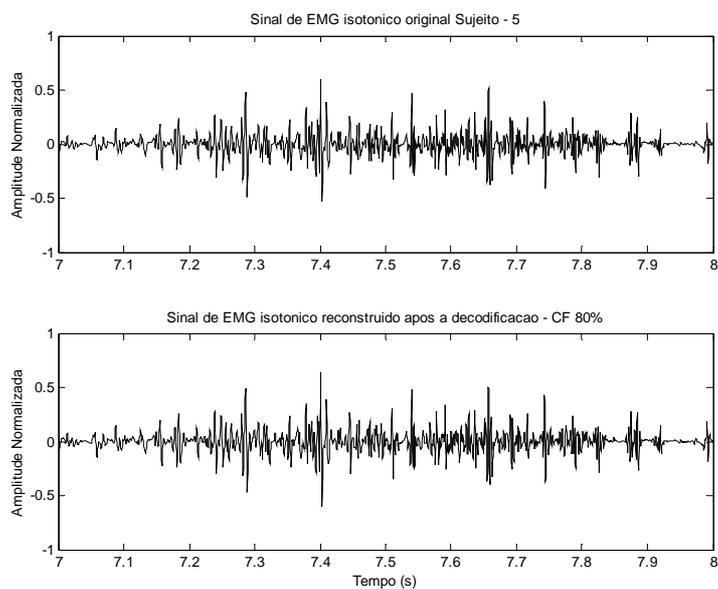


Figura B.19 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 19.

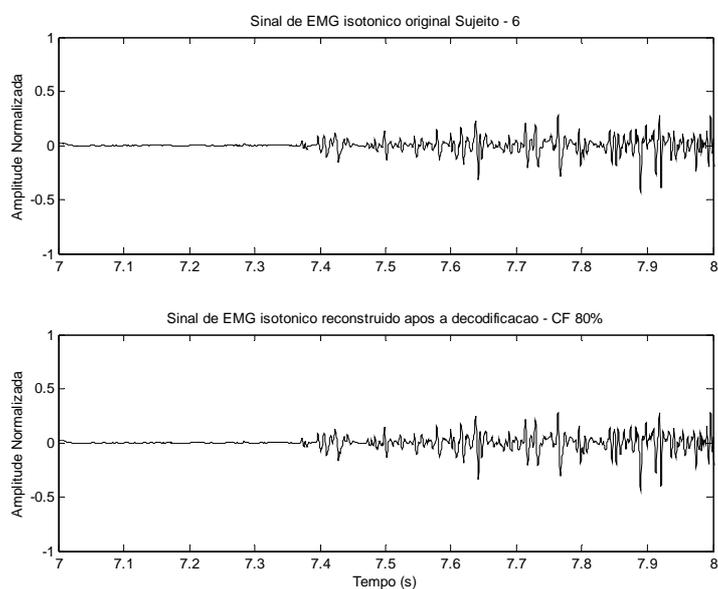


Figura B.20 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 20.

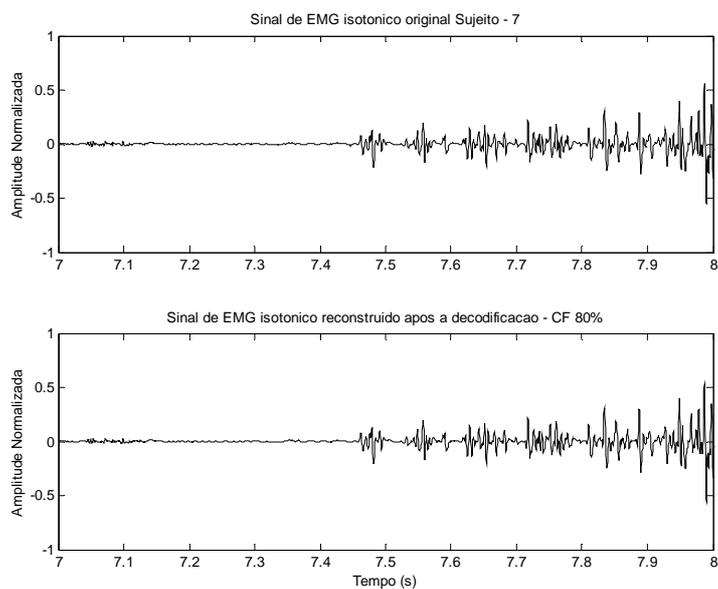


Figura B.21 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 21.

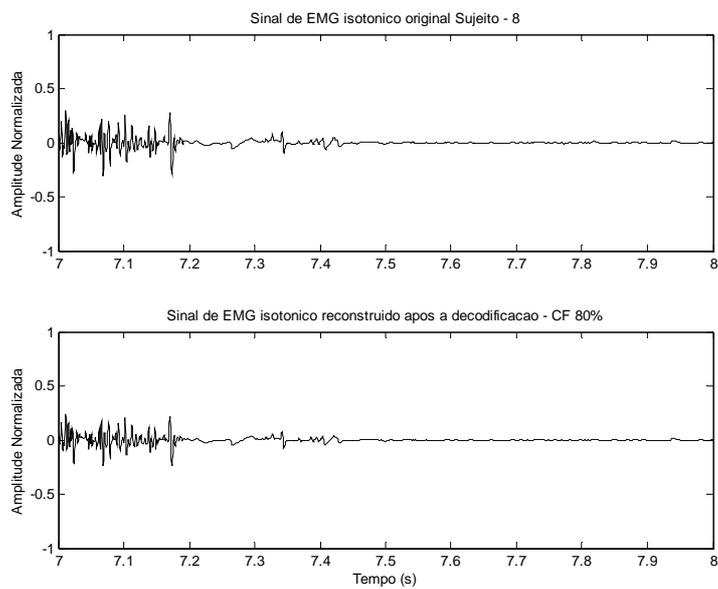


Figura B.22 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 22.

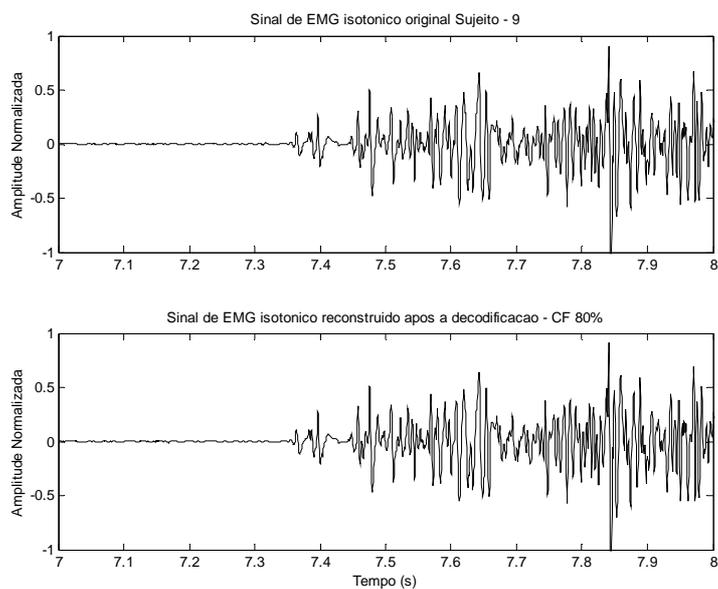


Figura B.23 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 23.

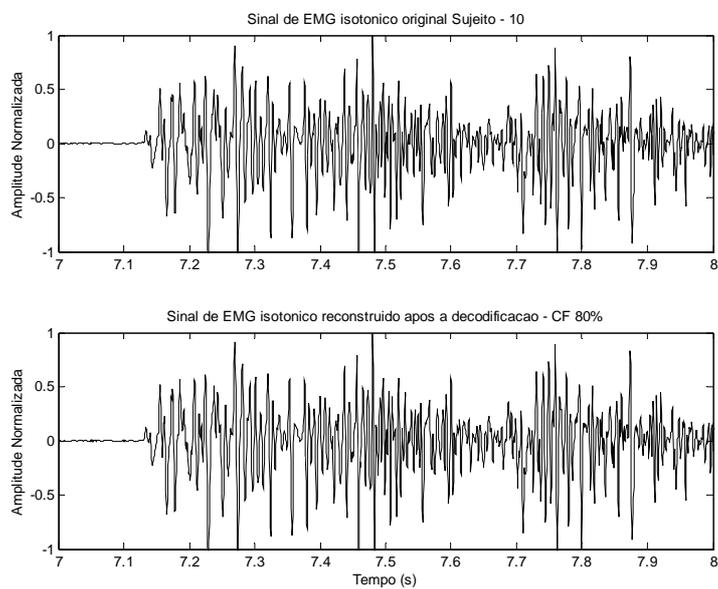


Figura B.24 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 24.

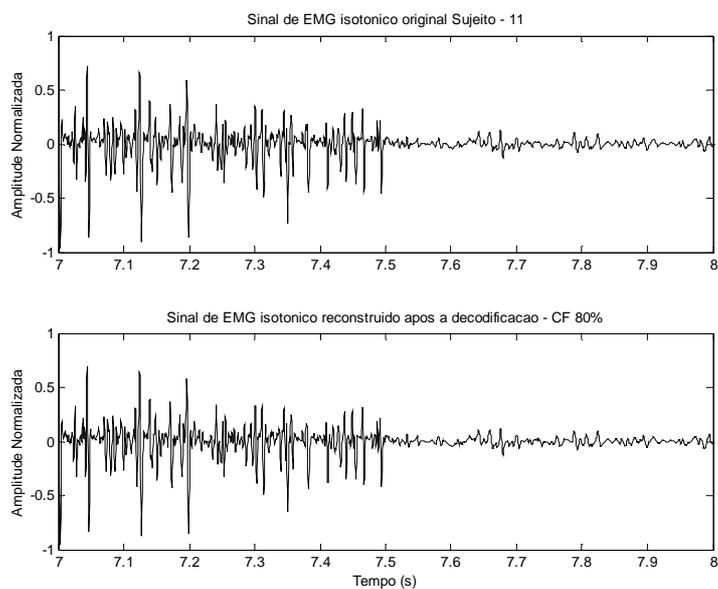


Figura B.25 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 25.

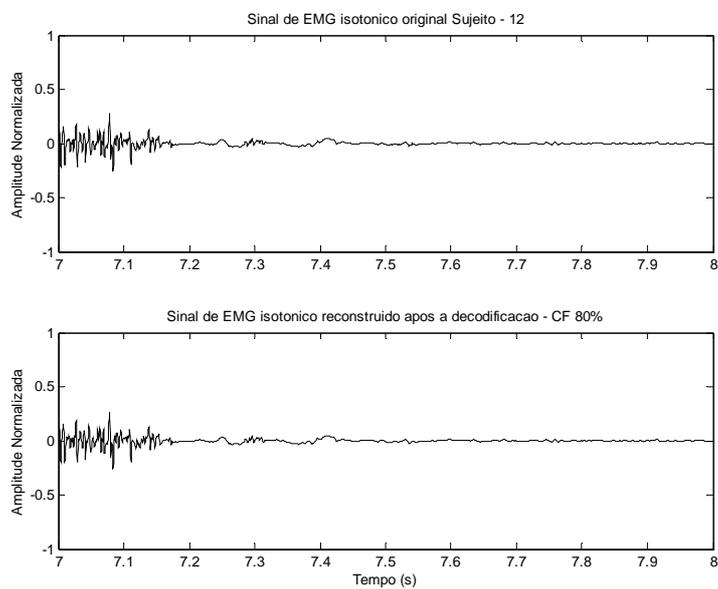


Figura B.26 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 26.

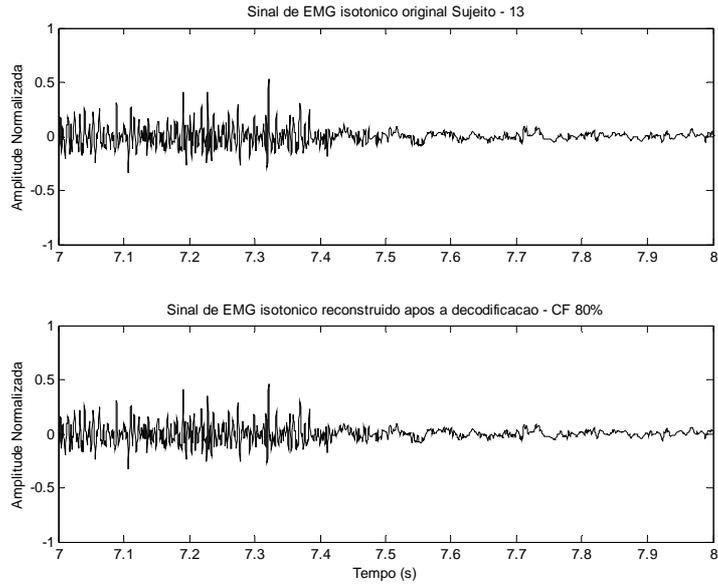


Figura B.27 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 27.

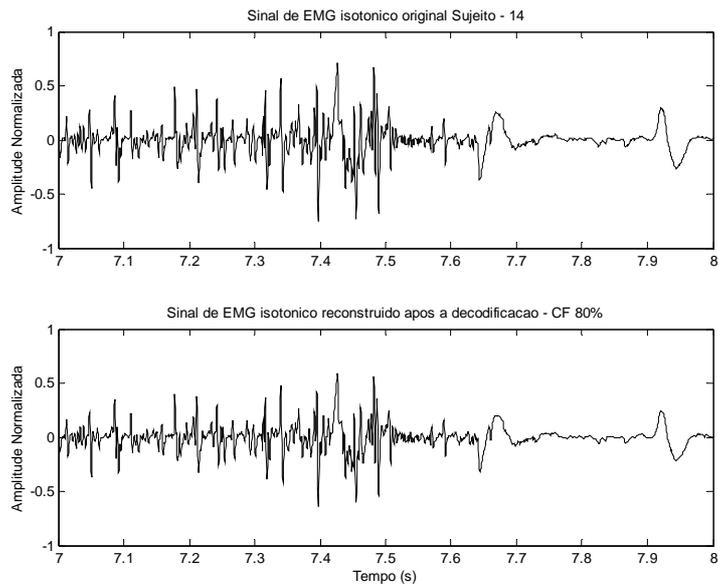


Figura B.28 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 28.

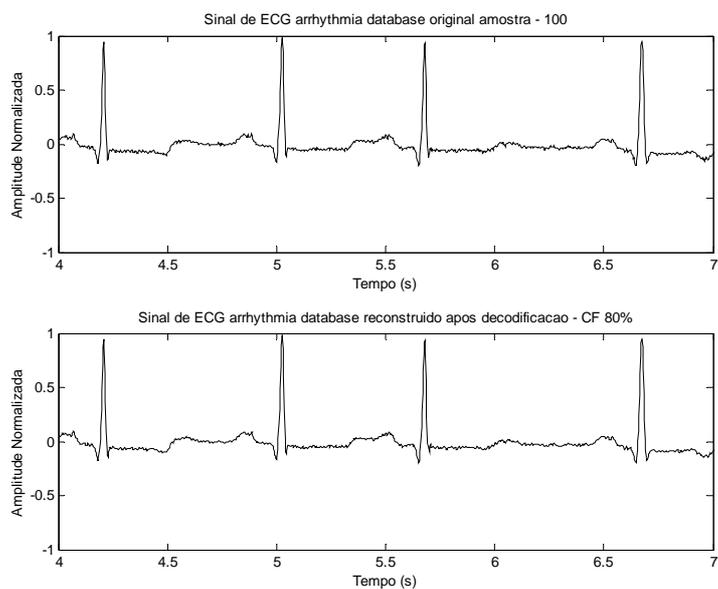


Figura B.29 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 29.

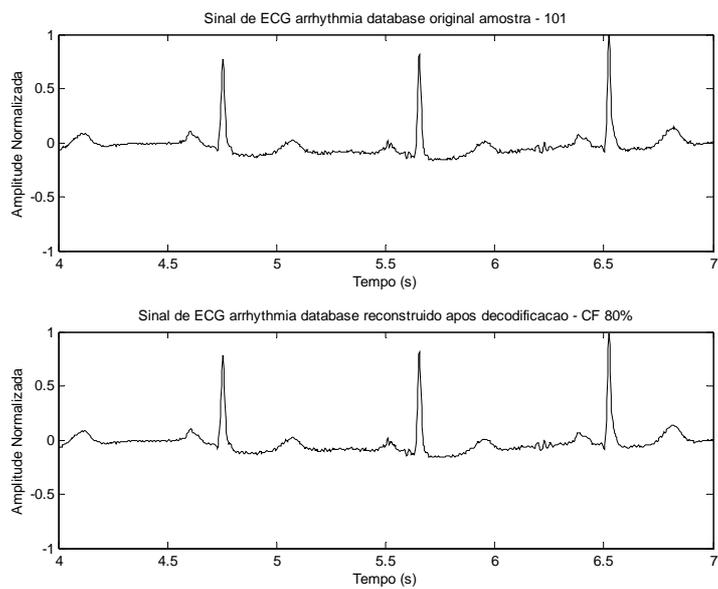


Figura B.30 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 30.

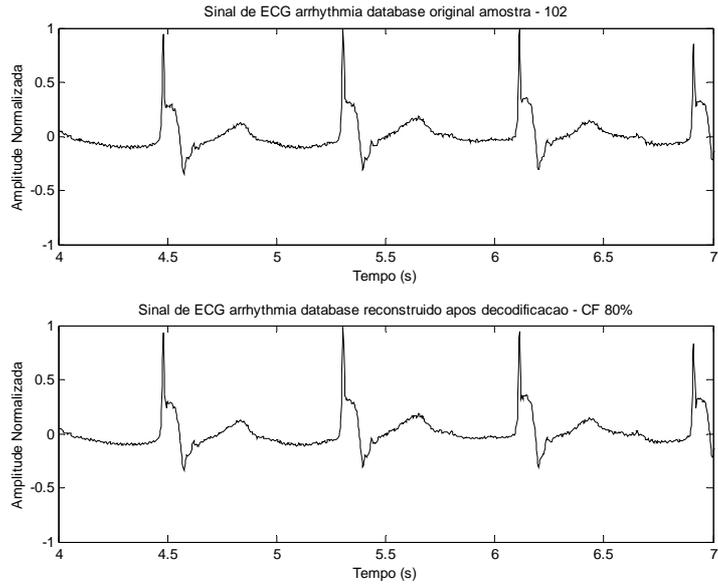


Figura B.31 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 31.

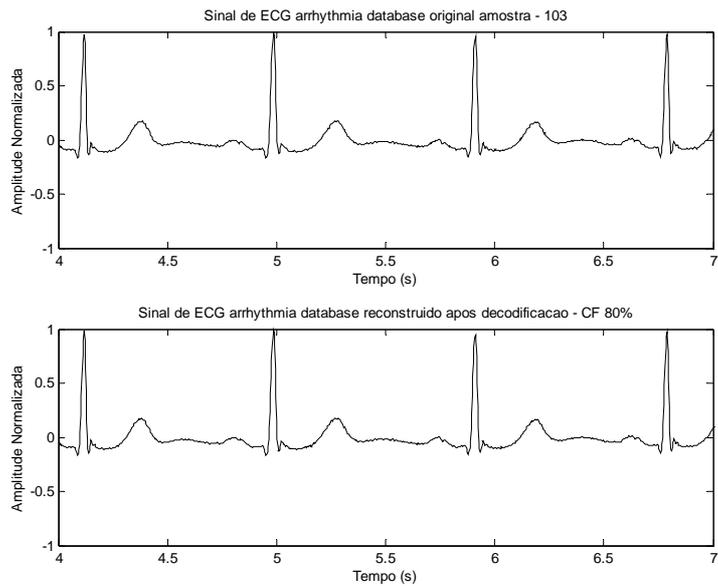


Figura B.32 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 32.

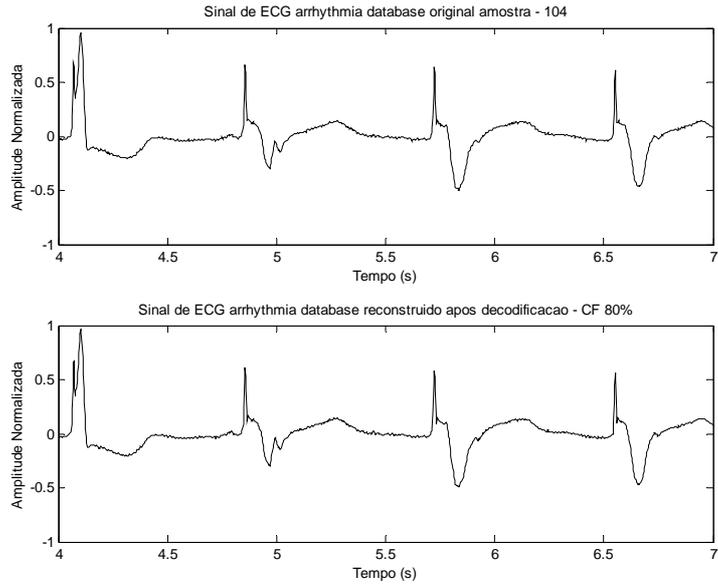


Figura B.33 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 33.

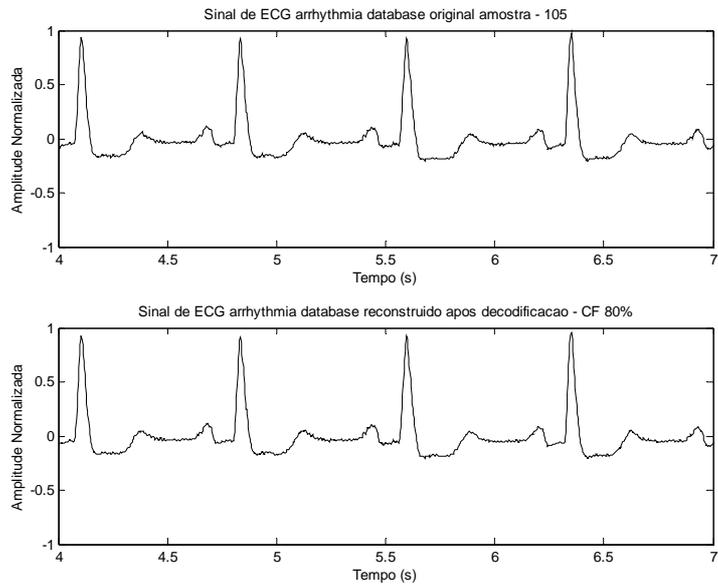


Figura B.34 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 34.

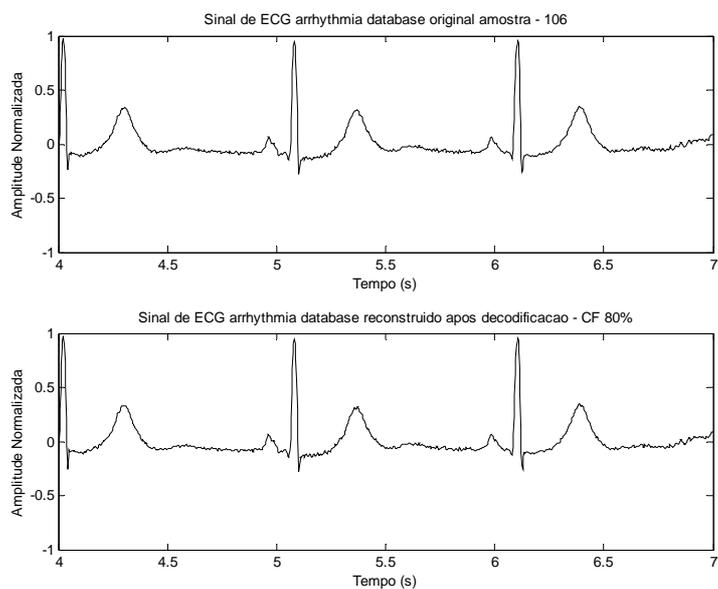


Figura B.35 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 35.

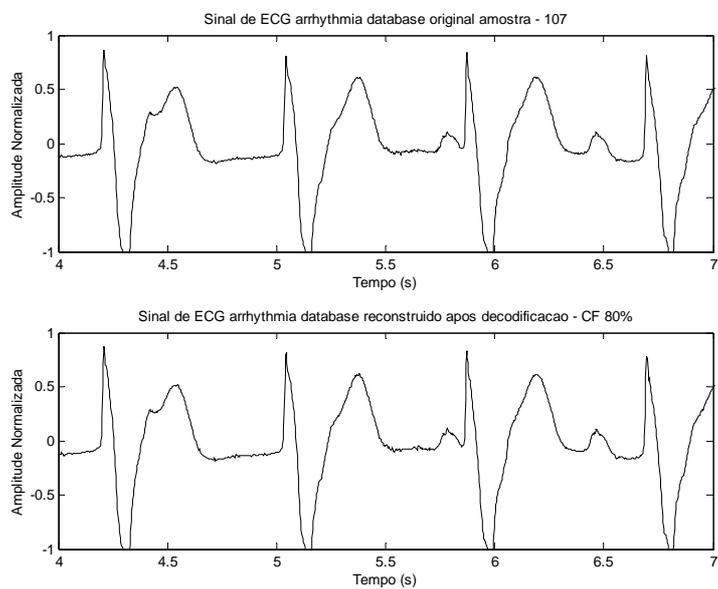


Figura B.36 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 36.

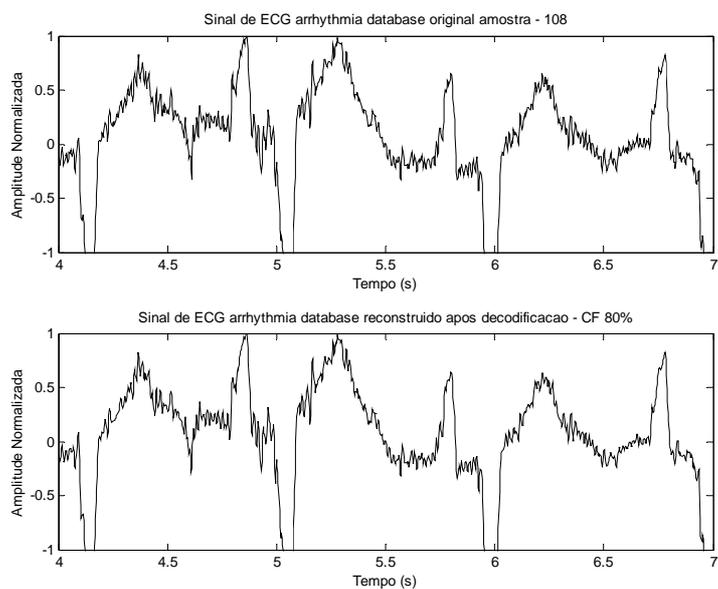


Figura B.37 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 37.

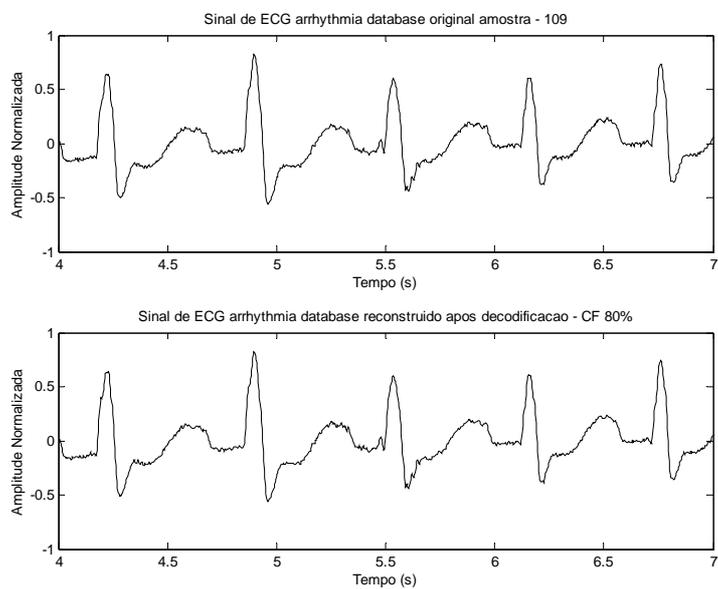


Figura B.38 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 38.

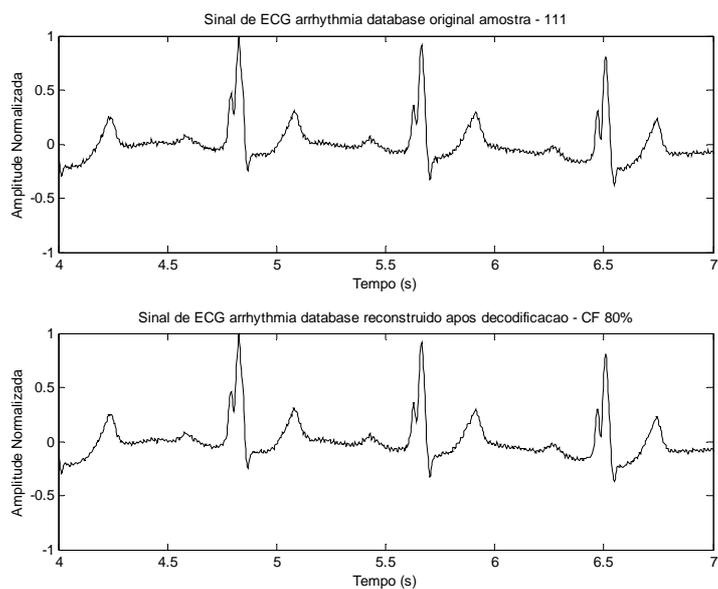


Figura B.39 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 39.

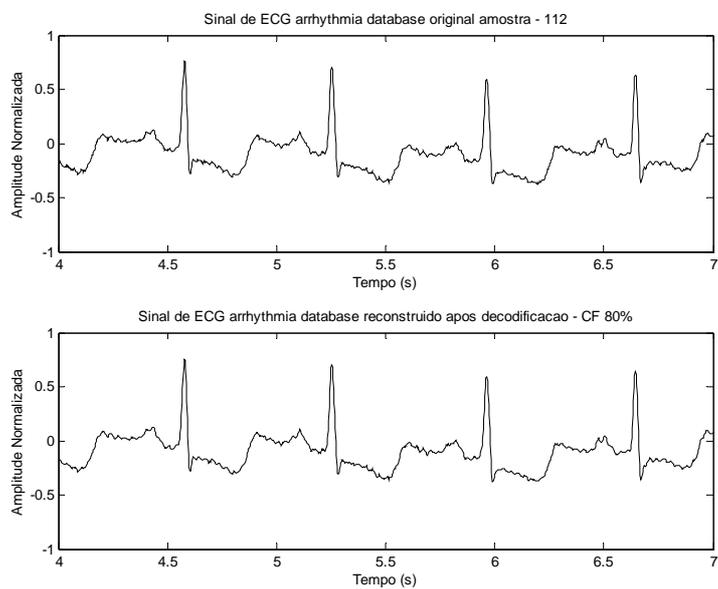


Figura B.40 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 40.

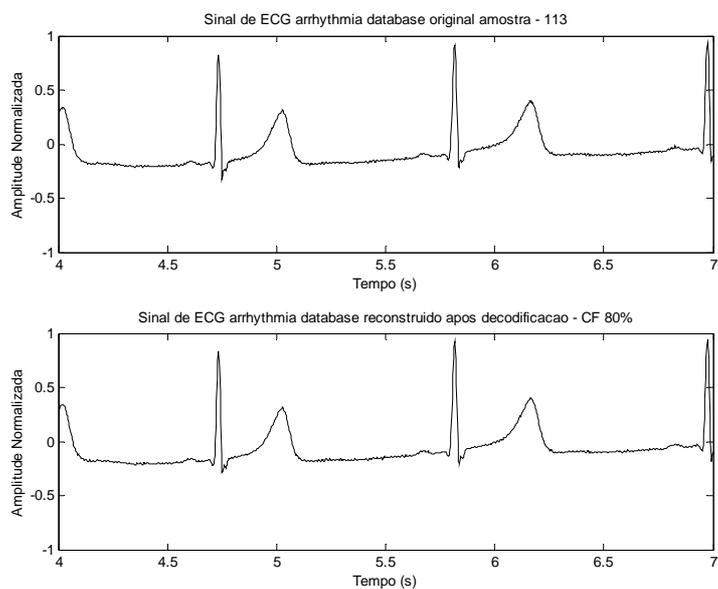


Figura B.41 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 41.

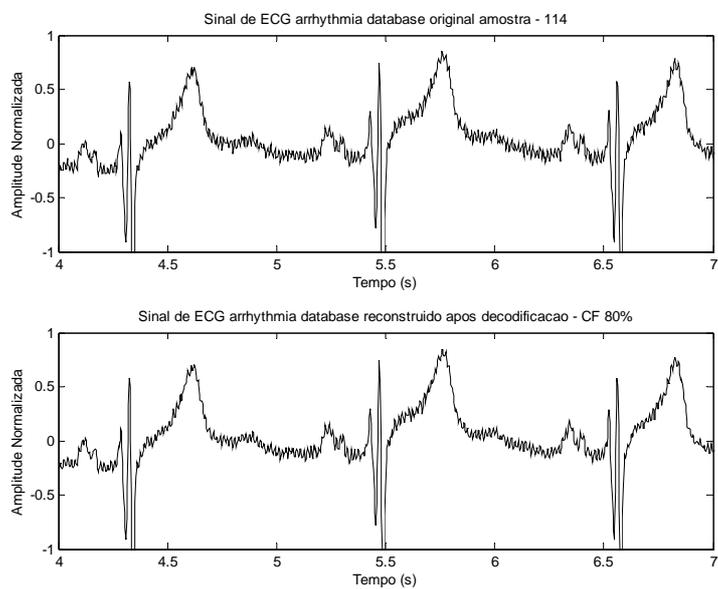


Figura B.42 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 42.

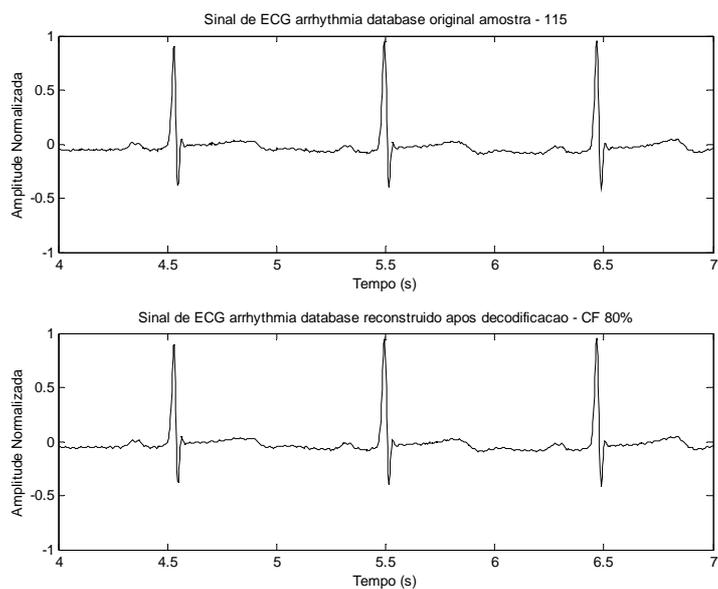


Figura B.43 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n<sup>o</sup> 43.

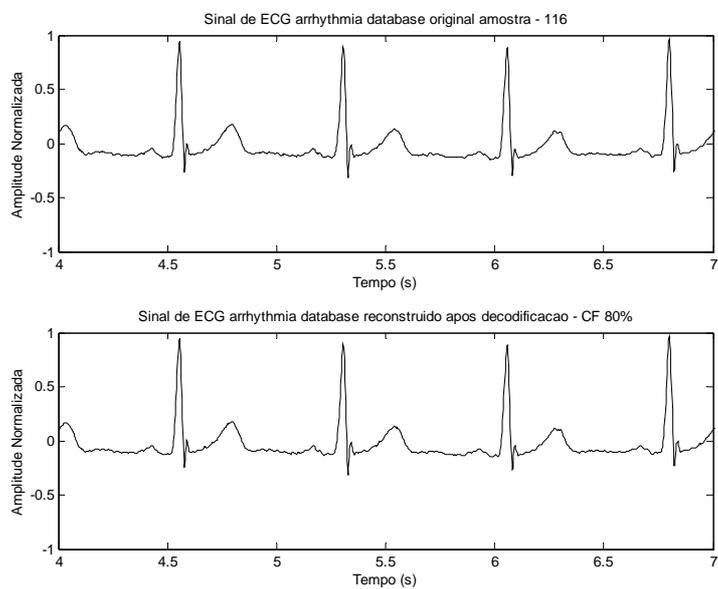


Figura B.44 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n<sup>o</sup> 44.

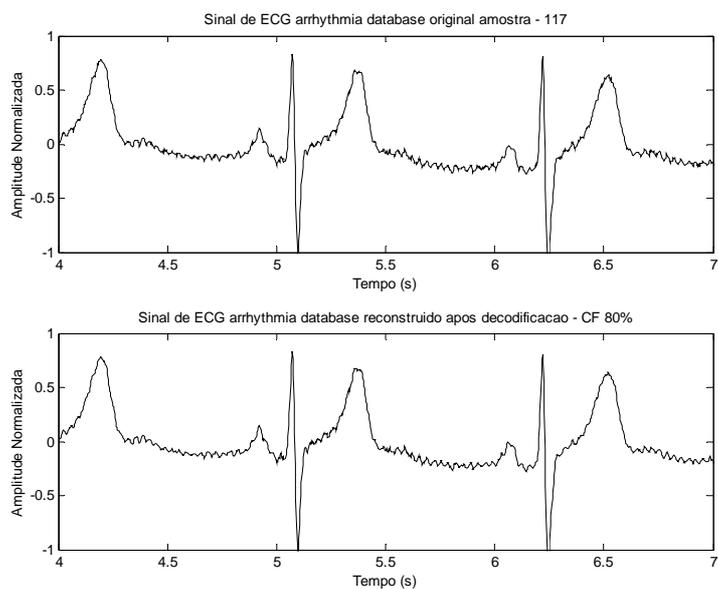


Figura B.45 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 45.

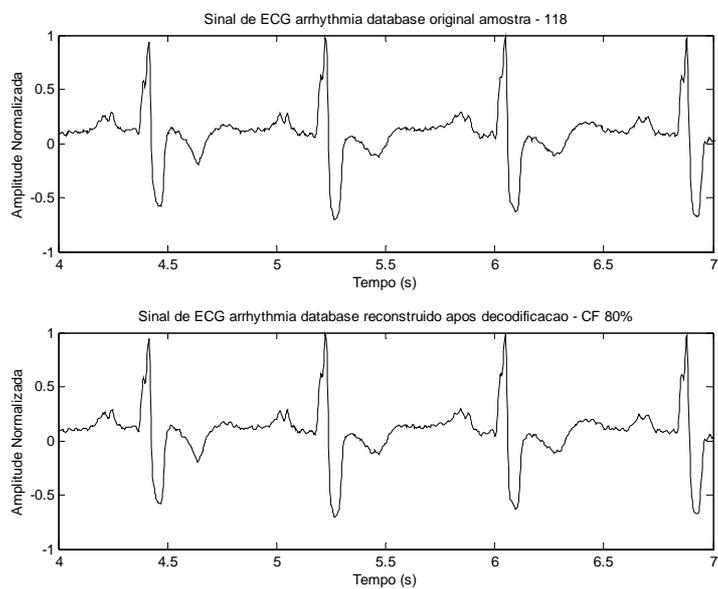


Figura B.46 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 46.

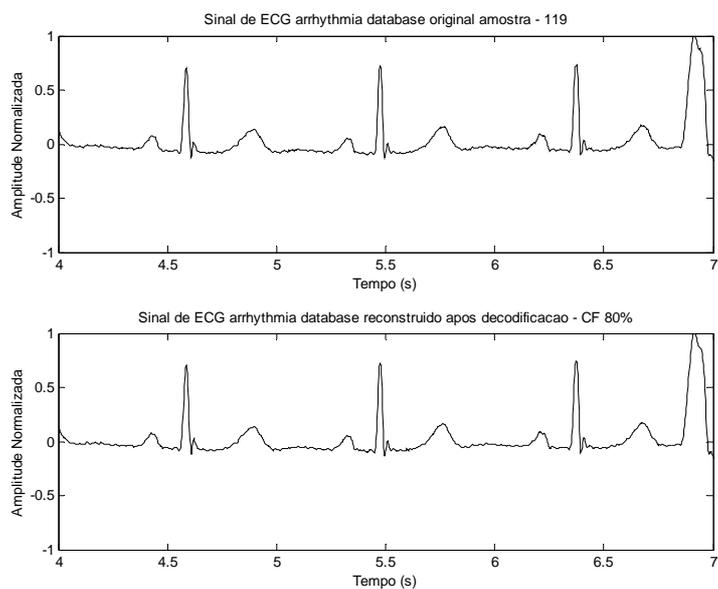


Figura B.47 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 47.

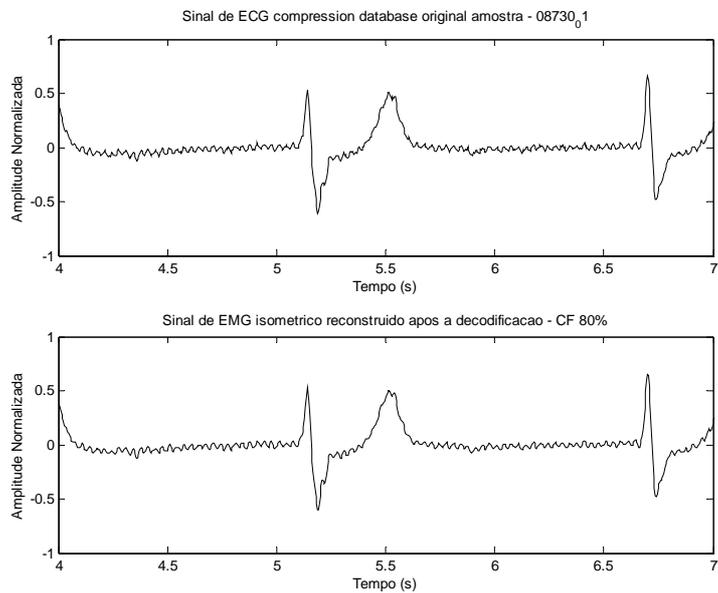


Figura B.48 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 48.

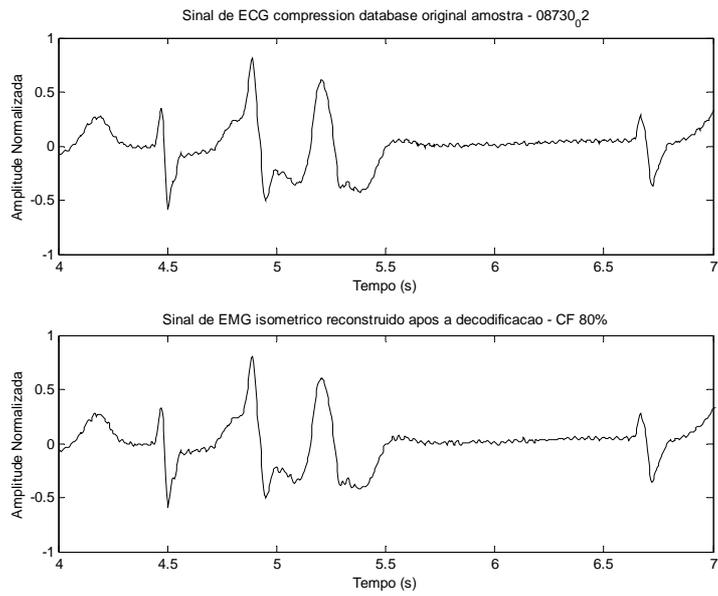


Figura B.49 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 49.

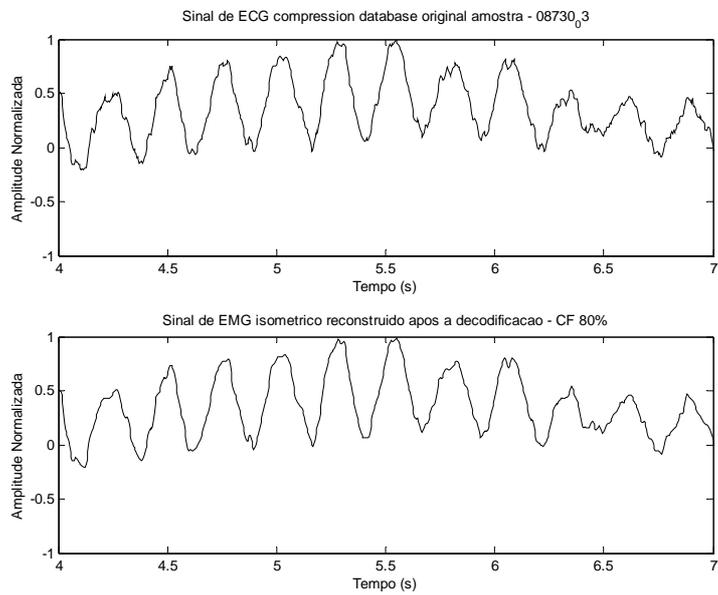


Figura B.50 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 50.

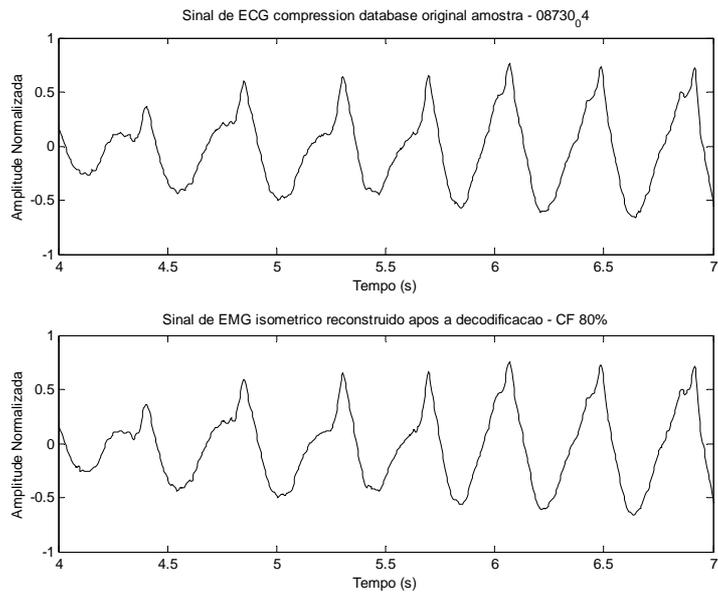


Figura B.51 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 51.

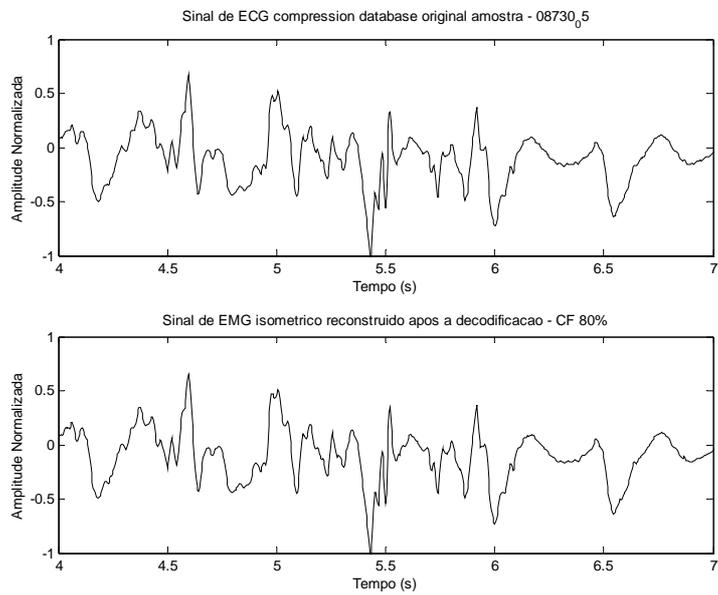


Figura B.52 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 52.

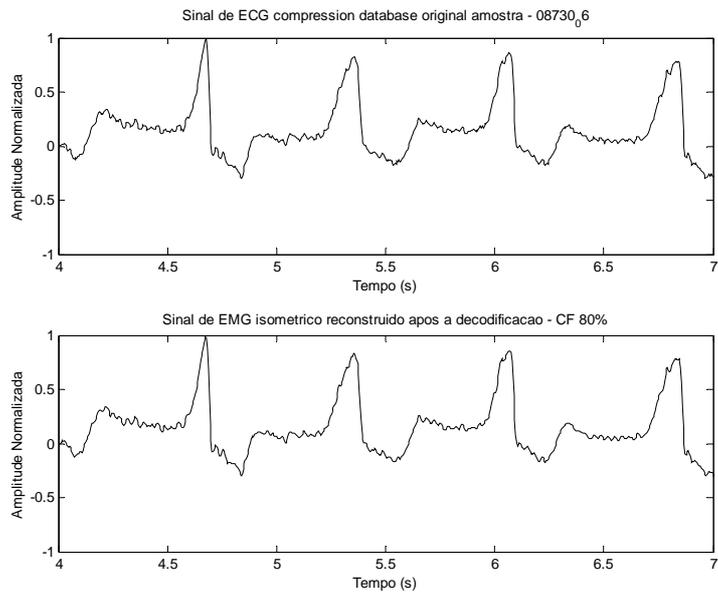


Figura B.53 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 53.

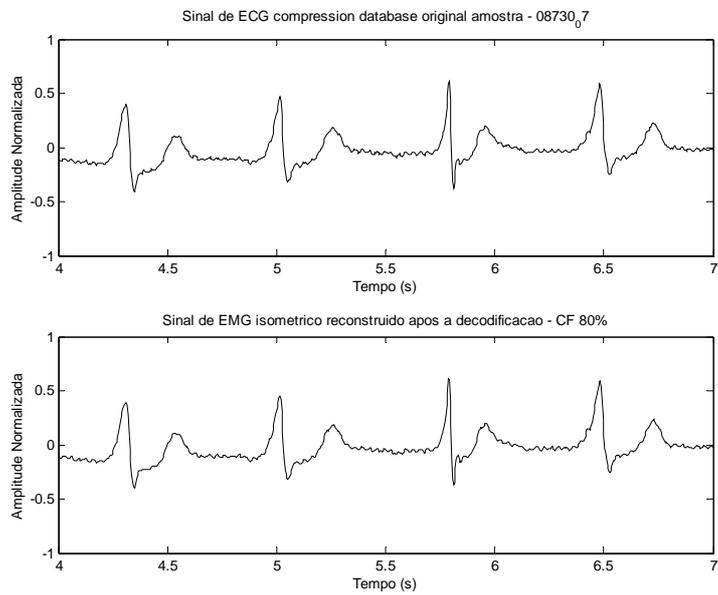


Figura B.54 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 54.

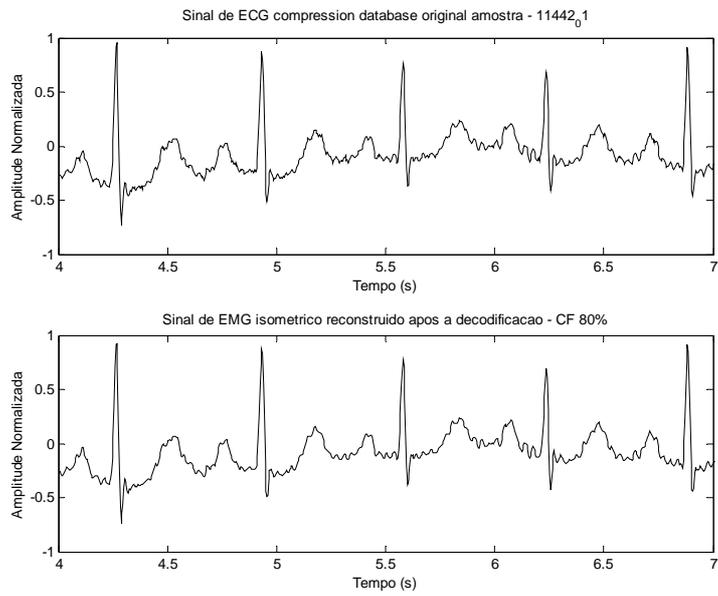


Figura B.55 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 55.

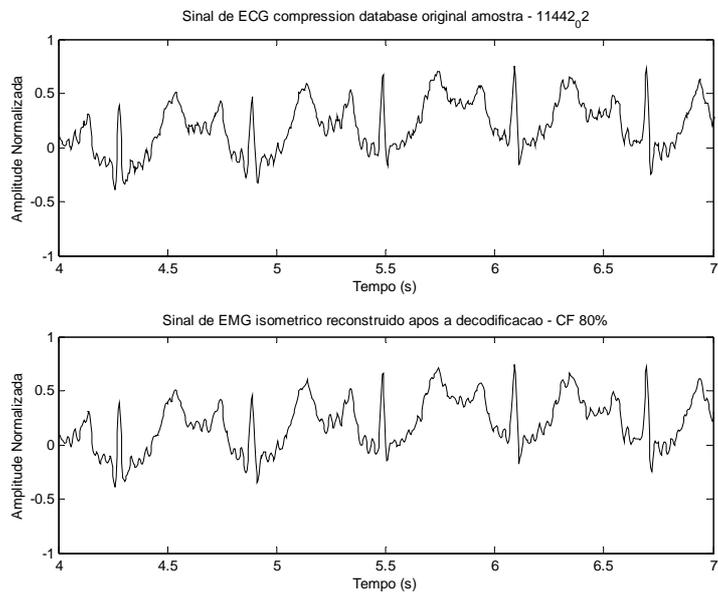


Figura B.56 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 56.

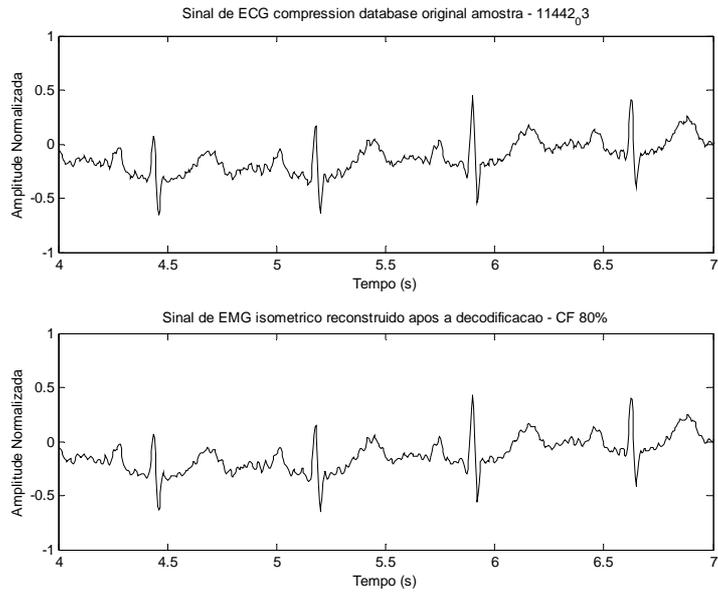


Figura B.57 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 57.

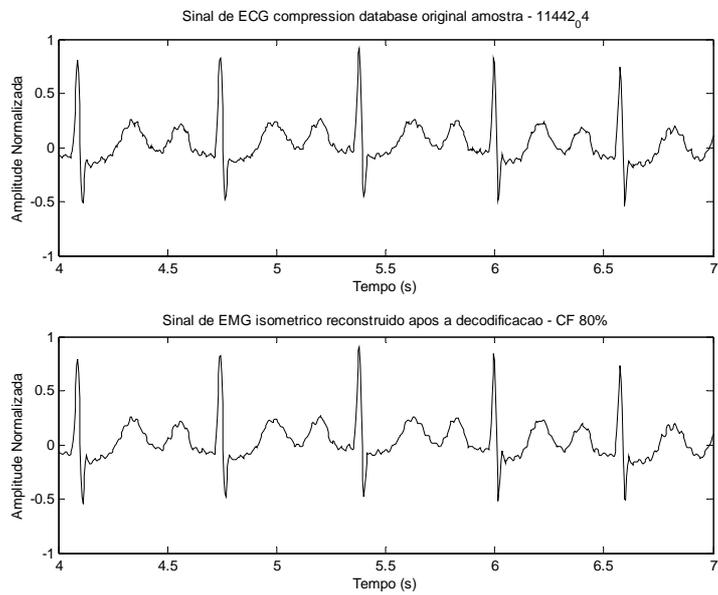


Figura B.58 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 58.

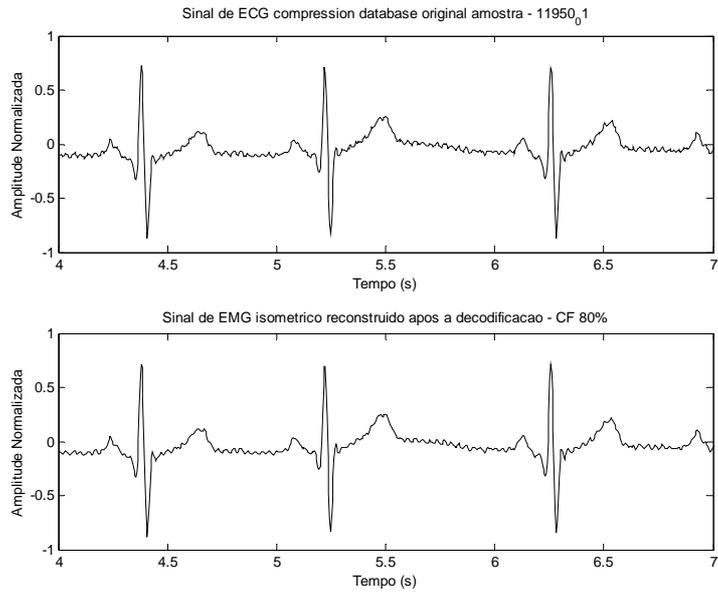


Figura B.59 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 59.

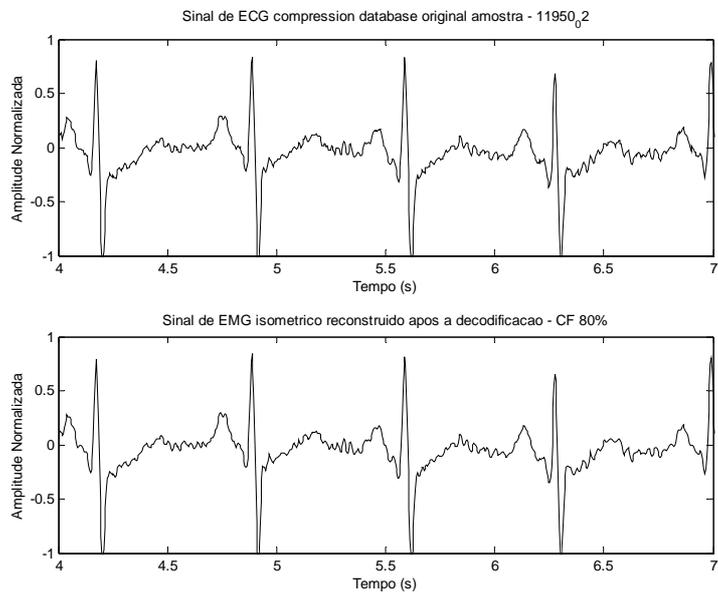


Figura B.60 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n ° 60.

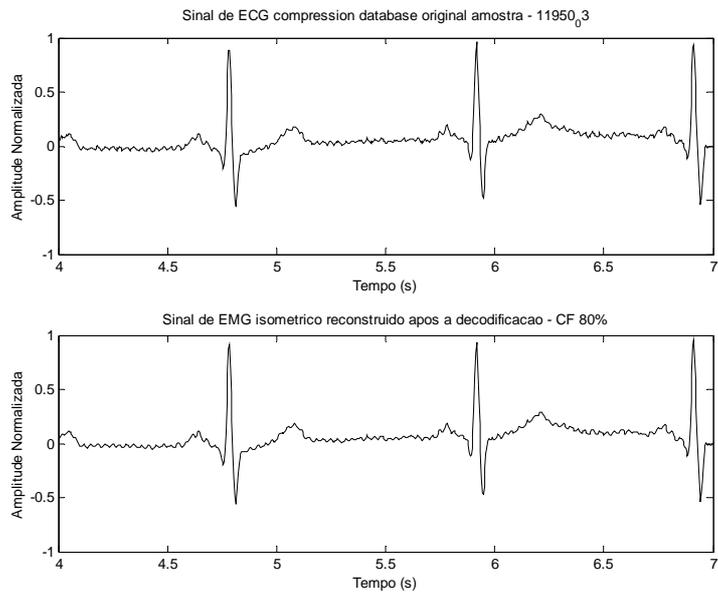


Figura B.61 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 61.

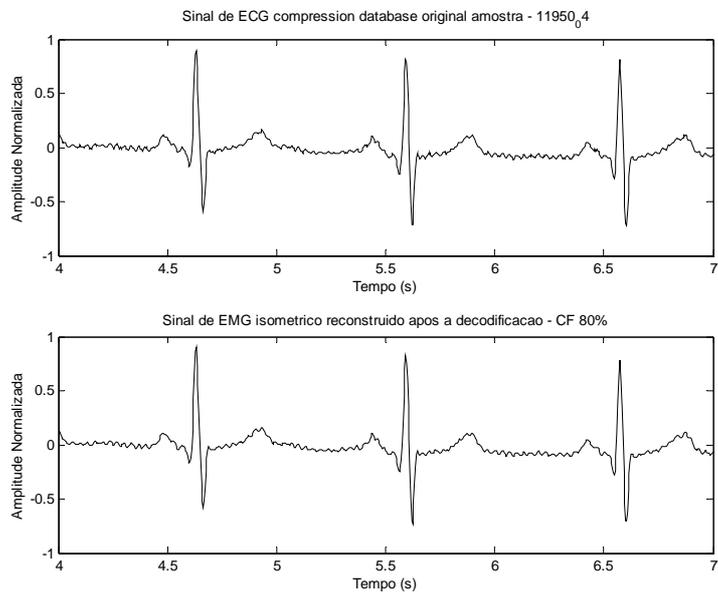


Figura B.62 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 62.

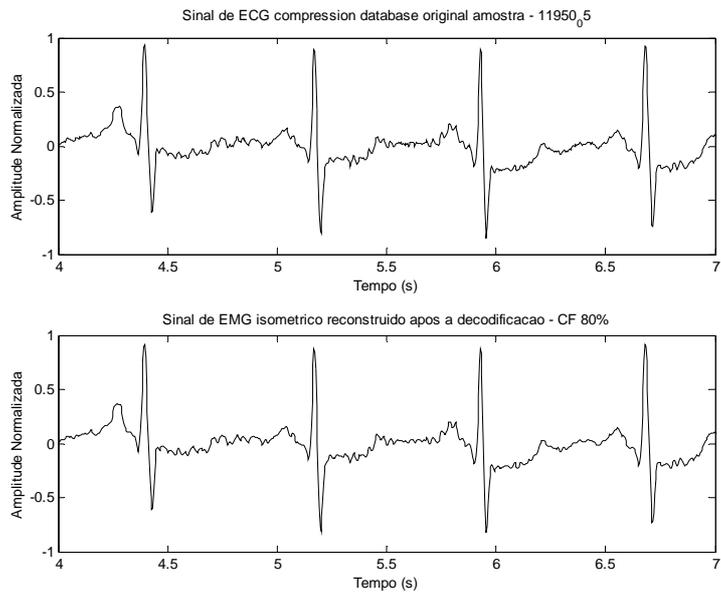


Figura B.63 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 63.

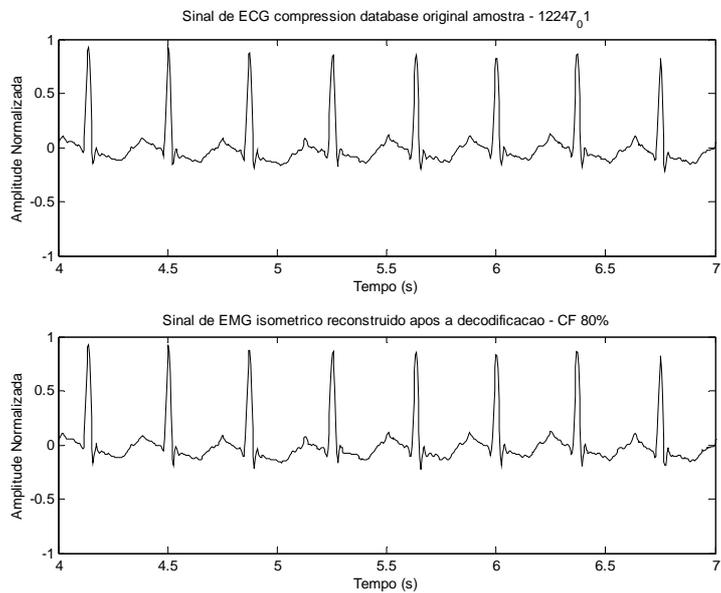


Figura B.64 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 64.

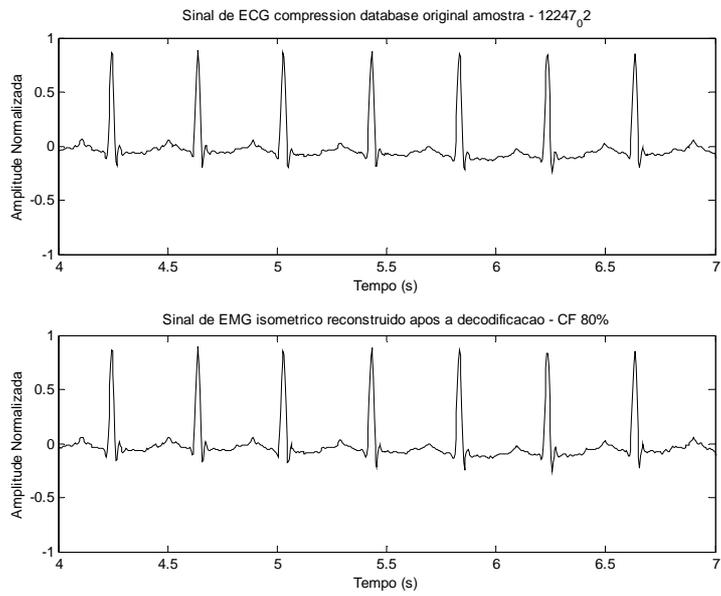


Figura B.65 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n<sup>o</sup> 65.

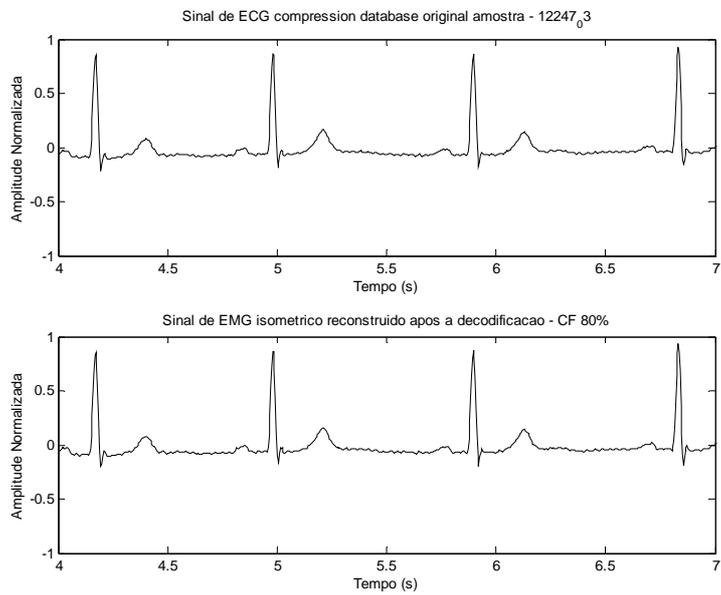


Figura B.66 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n<sup>o</sup> 66.

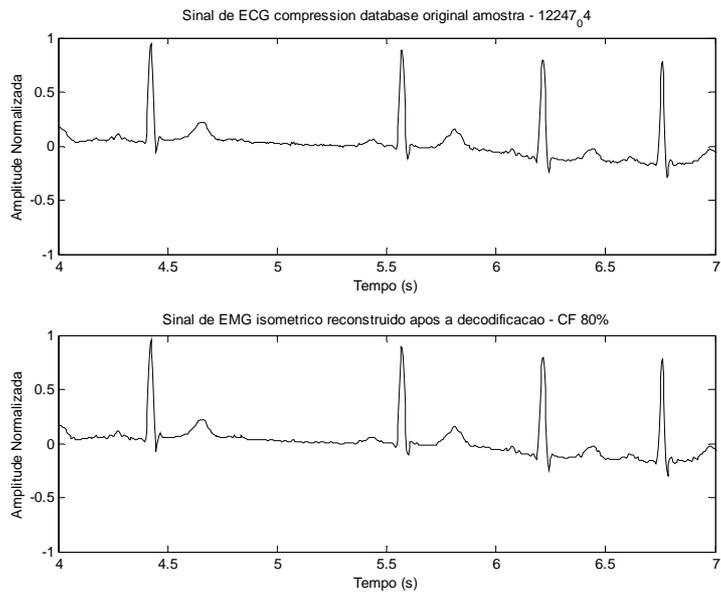


Figura B.67 – Resultado da compressão de sinais eletrofisiológicos sem *de-noising* n° 67.

## C - RESULTADOS DA COMPRESSÃO DE SINAIS ELETROFISIOLÓGICOS SUBMETIDOS AO PRÉ-PROCESSAMENTO COM *DE-NOISING*

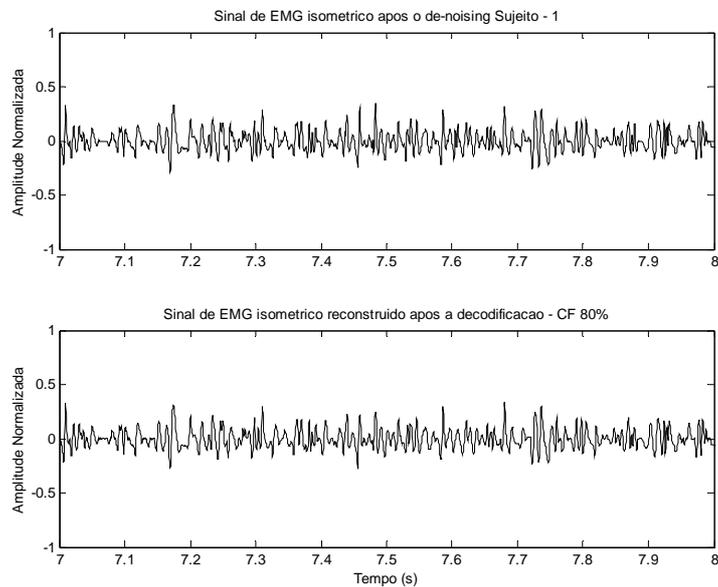


Figura C.1 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 1.

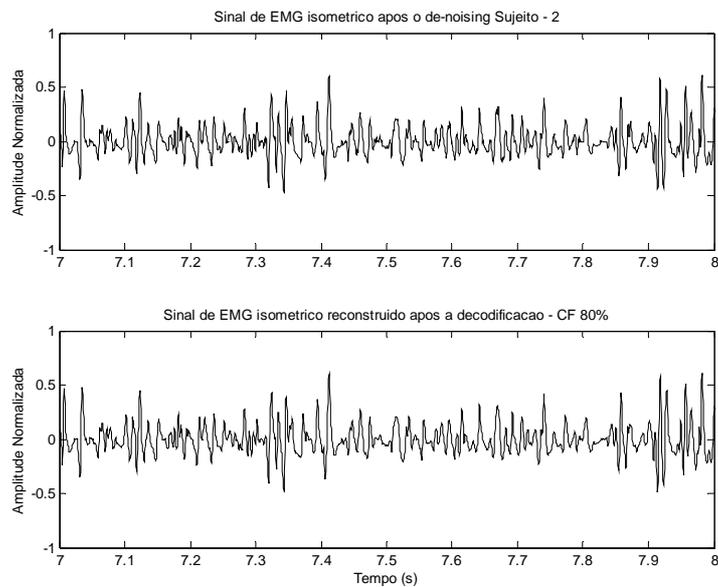


Figura C.2 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 2.

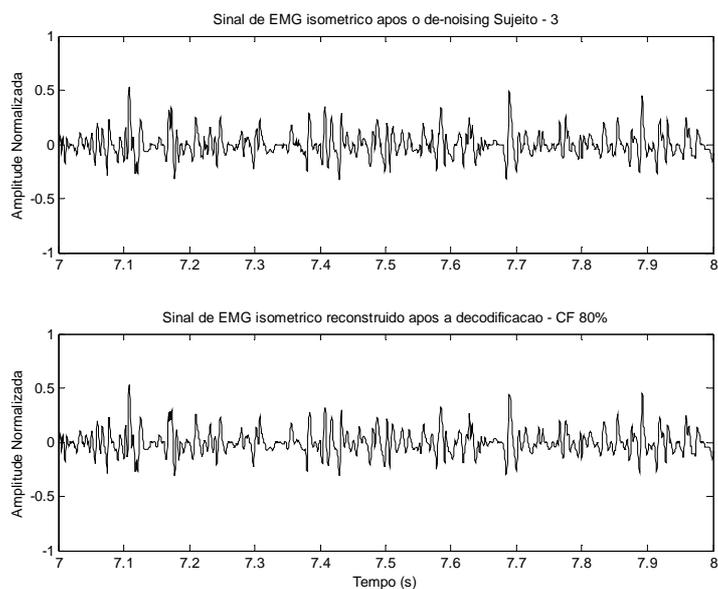


Figura C.3 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 3.

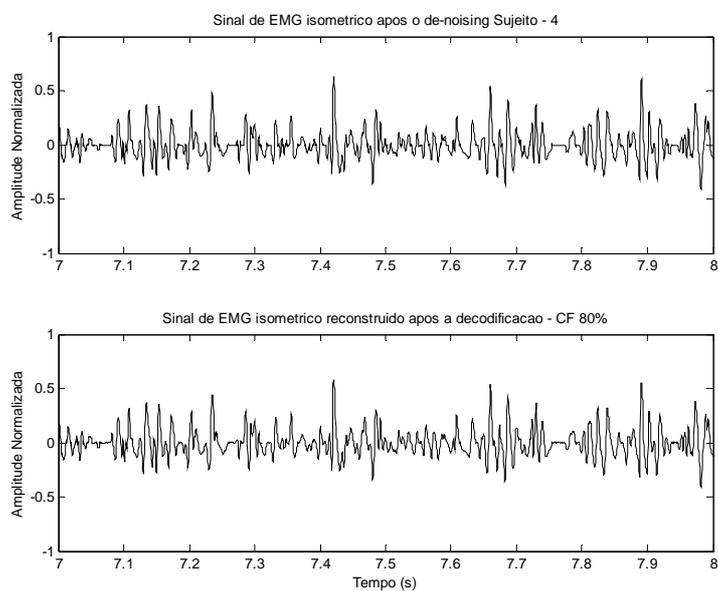


Figura C.4 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 4.

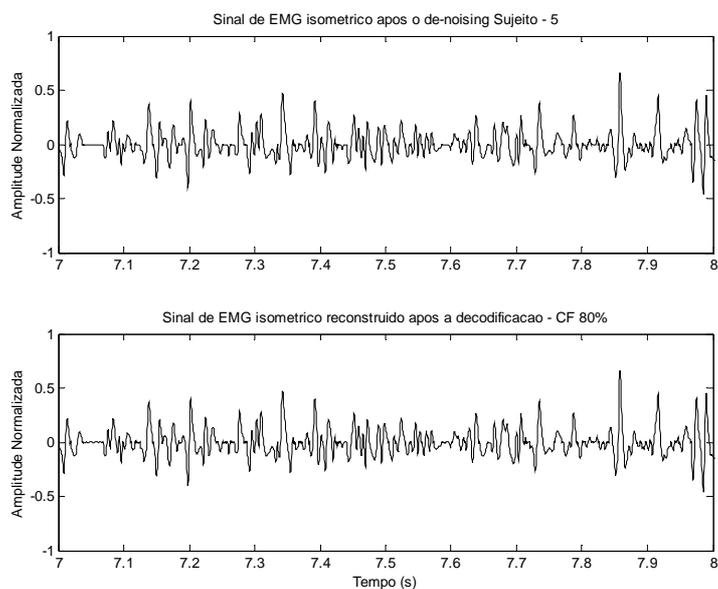


Figura C.5 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 5.

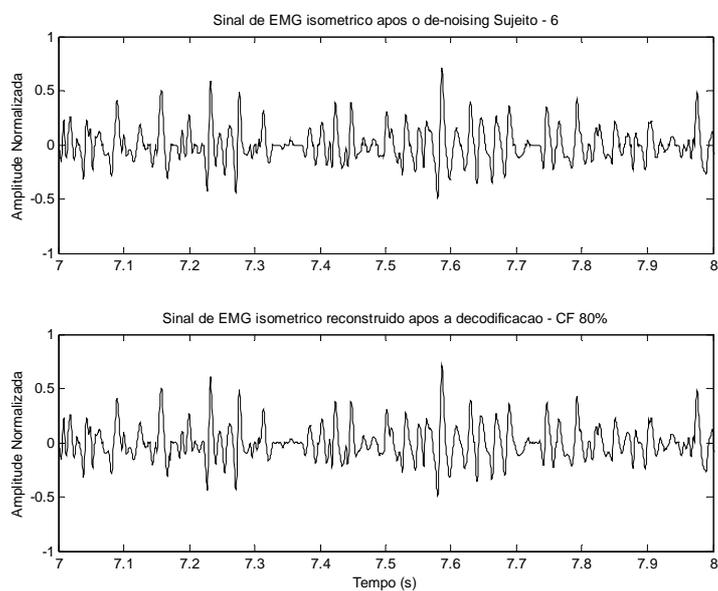


Figura C.6 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 6.

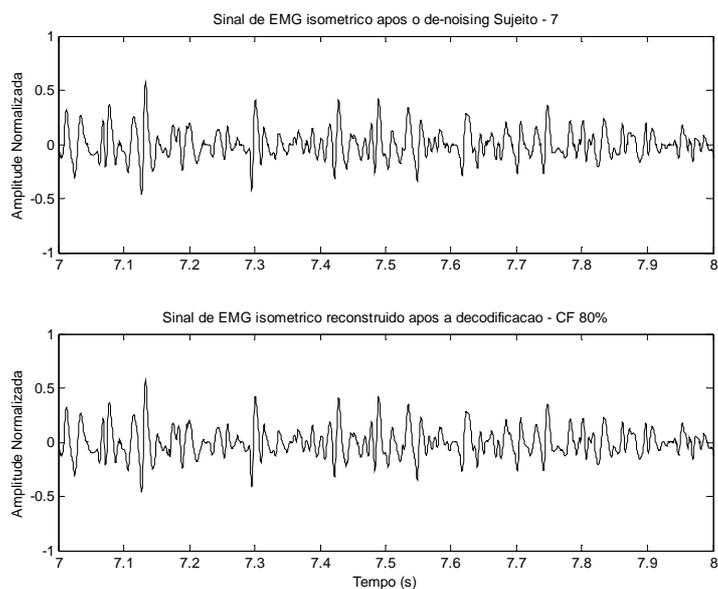


Figura C.7 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 7.

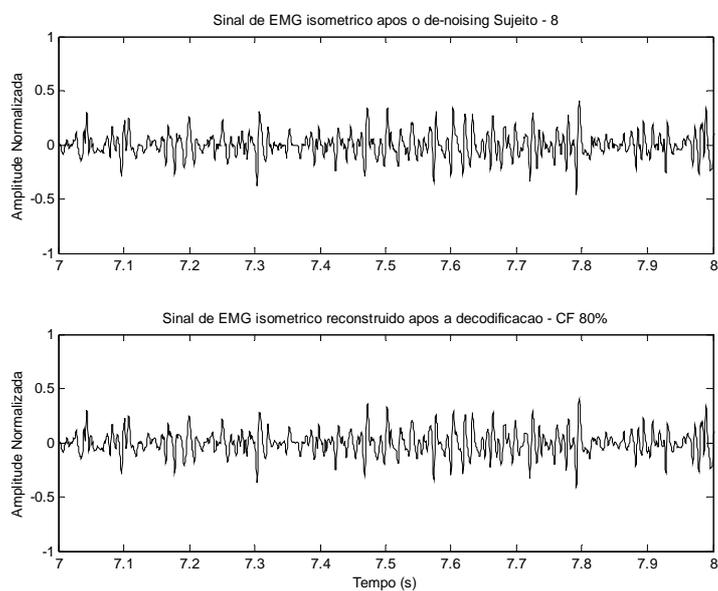


Figura C.8 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 8.

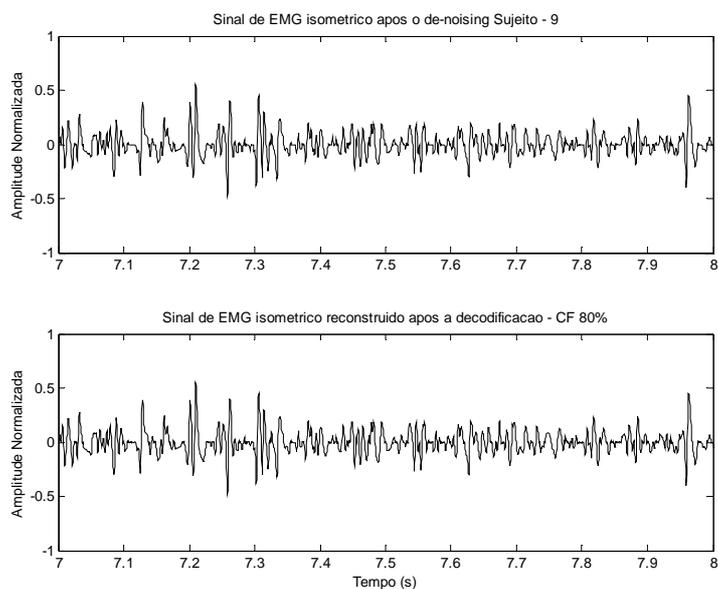


Figura C.9 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 9.

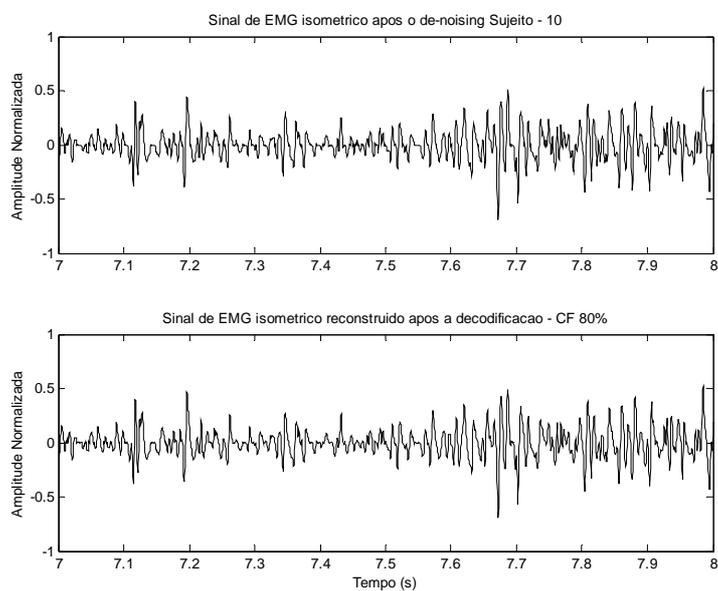


Figura C.10 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 10.

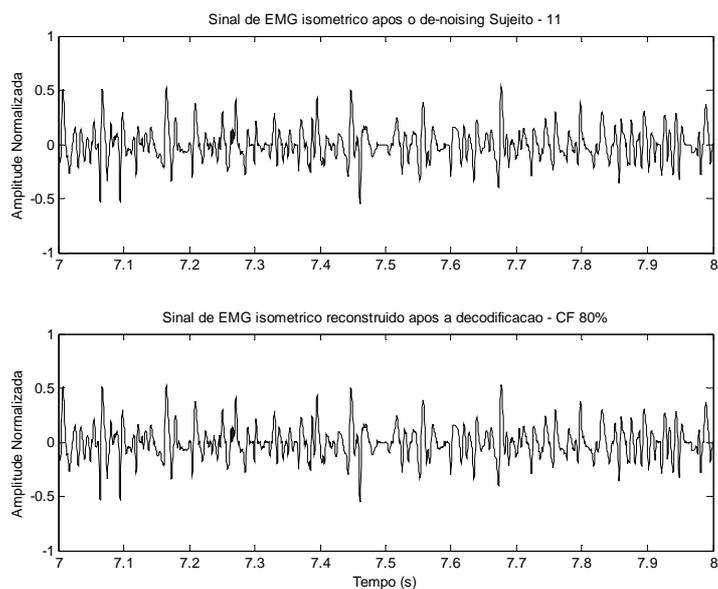


Figura C.11 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 11.

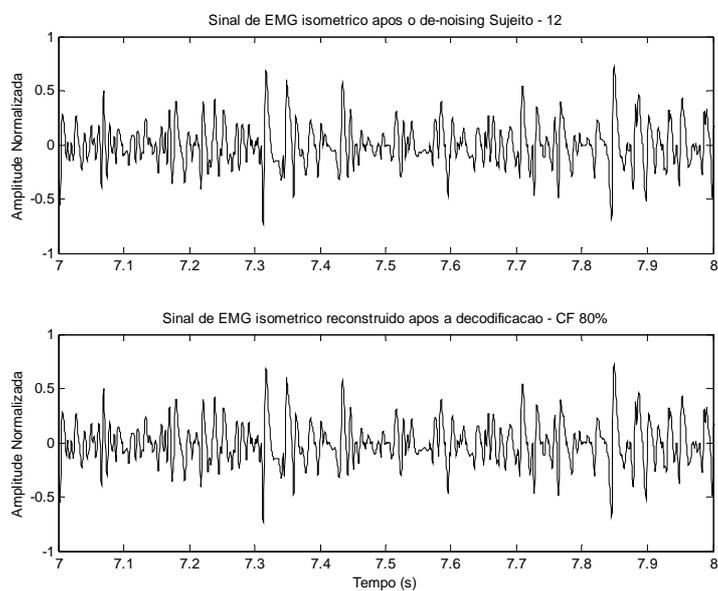


Figura C.12 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 12.

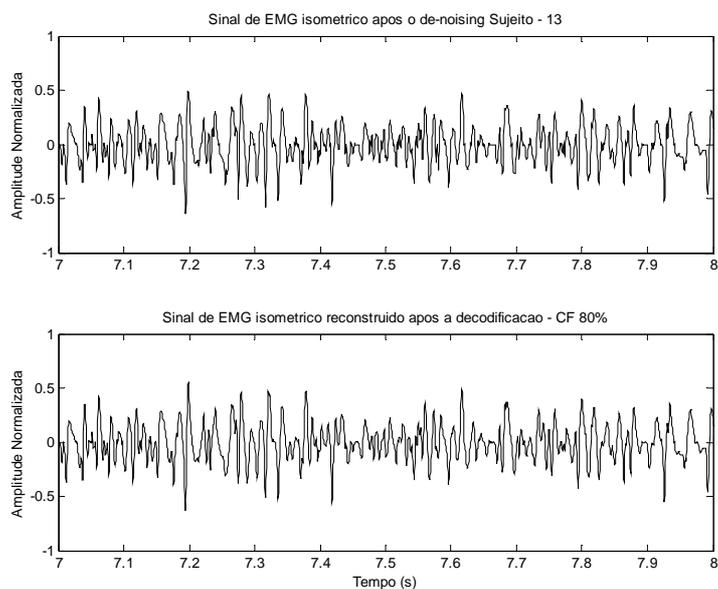


Figura C.13 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 13.

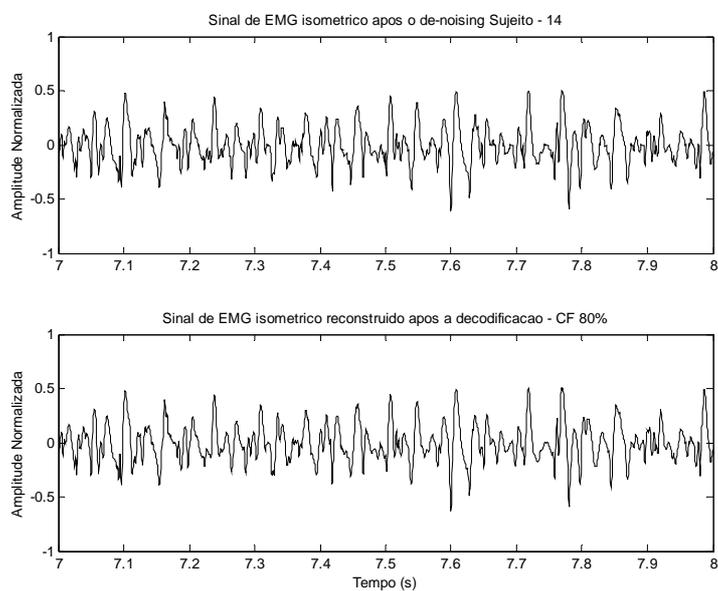


Figura C.14 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 14.

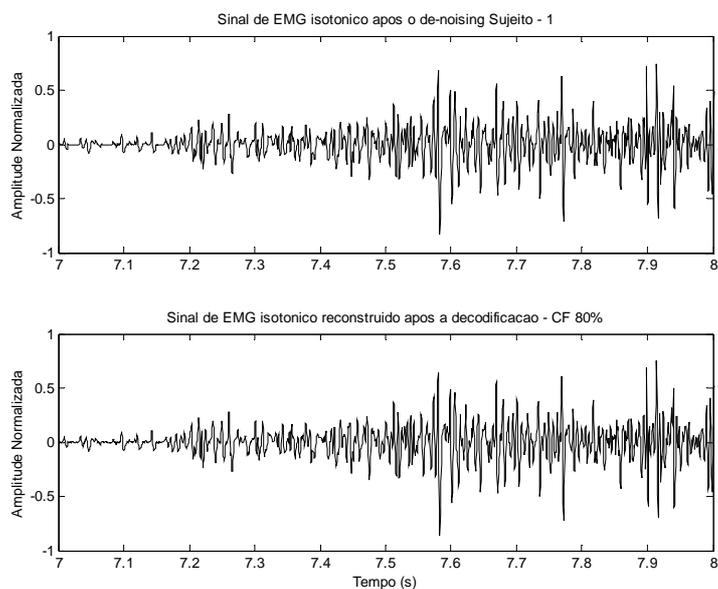


Figura C.15 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 15.

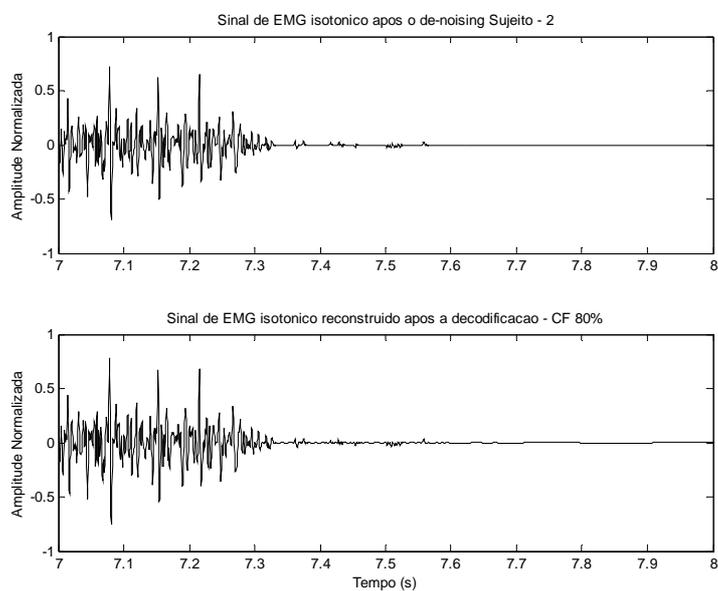


Figura C.16 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 16.

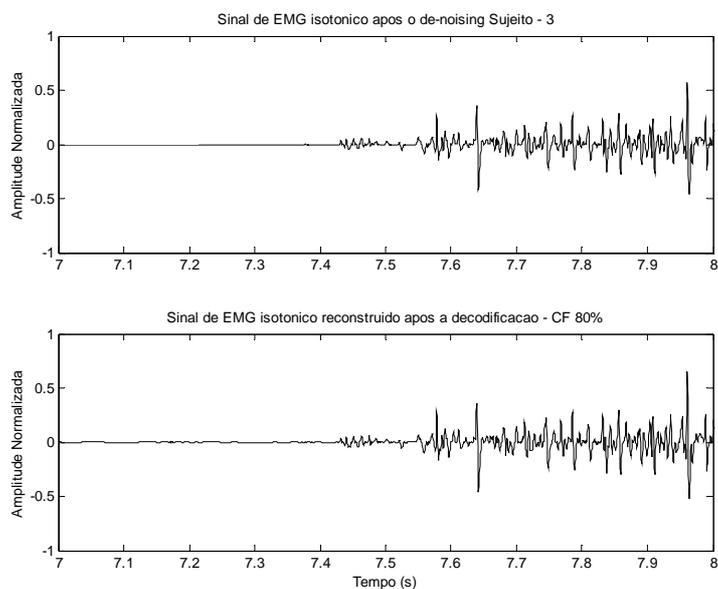


Figura C.17 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 17.

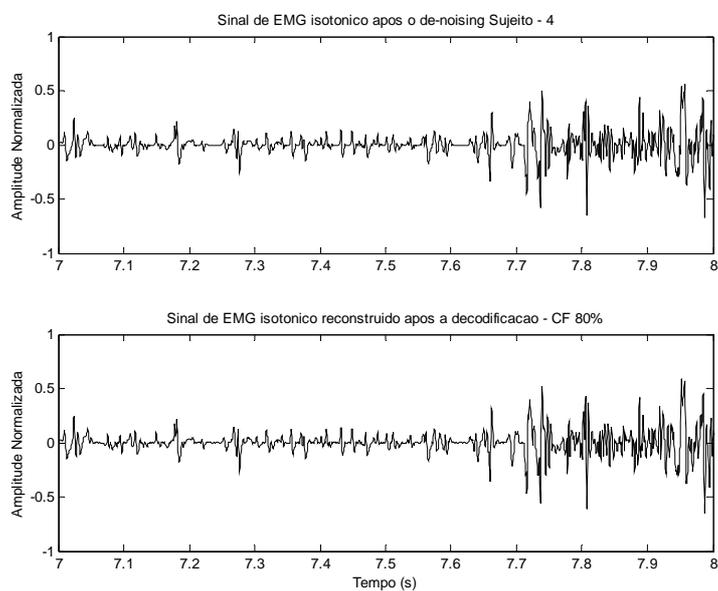


Figura C.18 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 18.

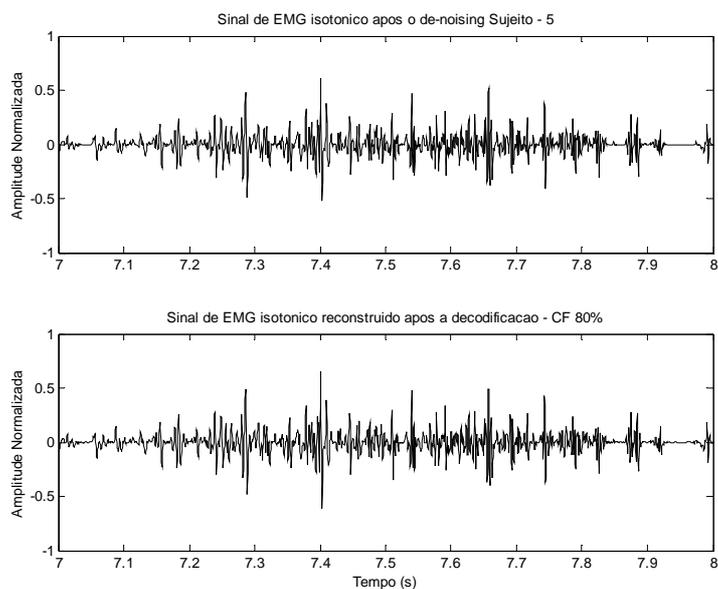


Figura C.19 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 19.

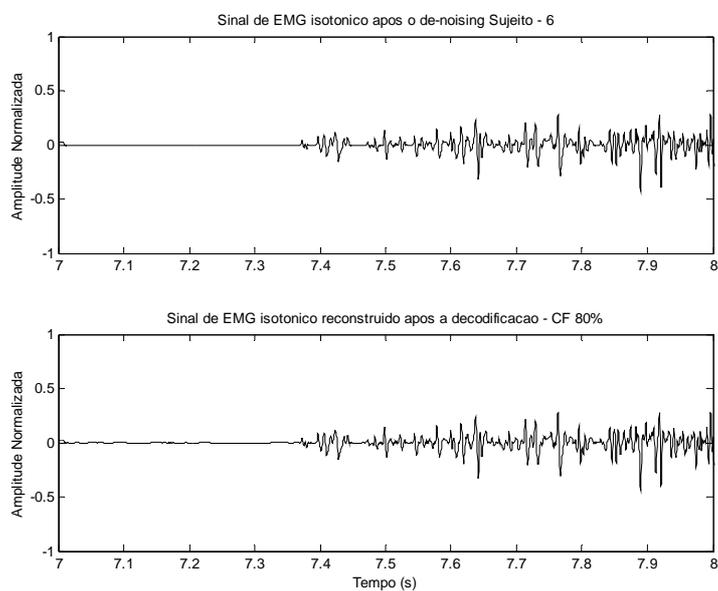


Figura C.20 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 20.

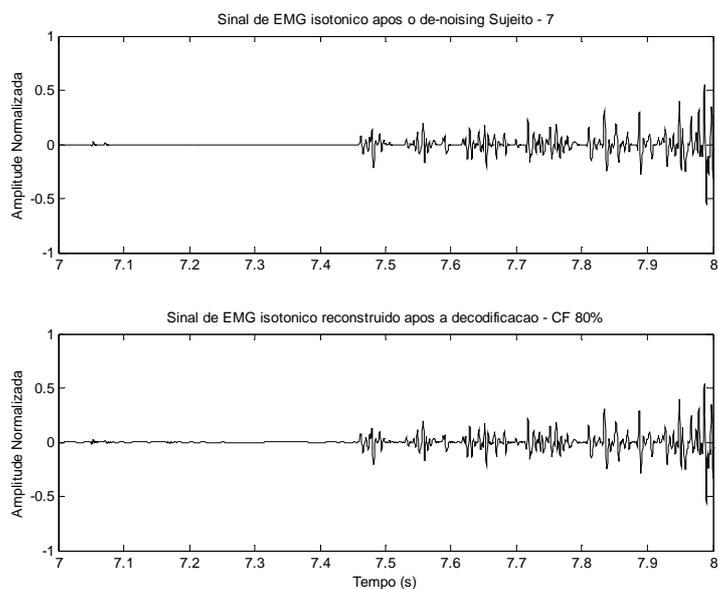


Figura C.21 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 21.

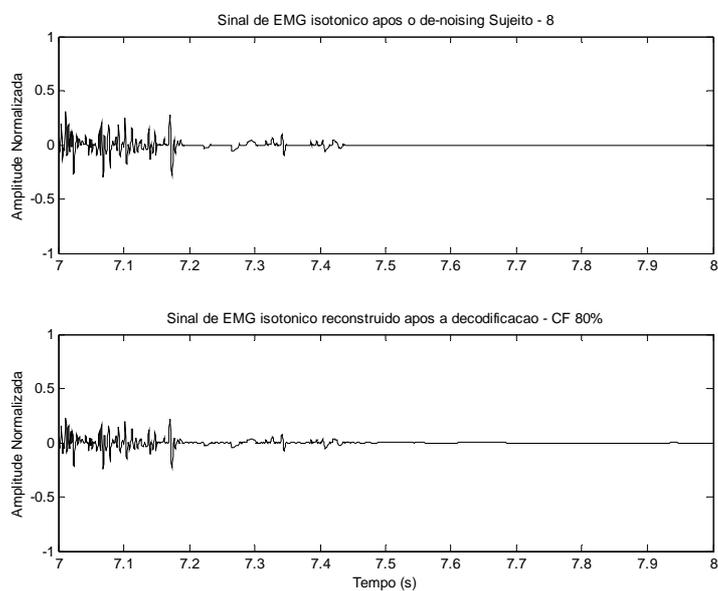


Figura C.22 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 22.

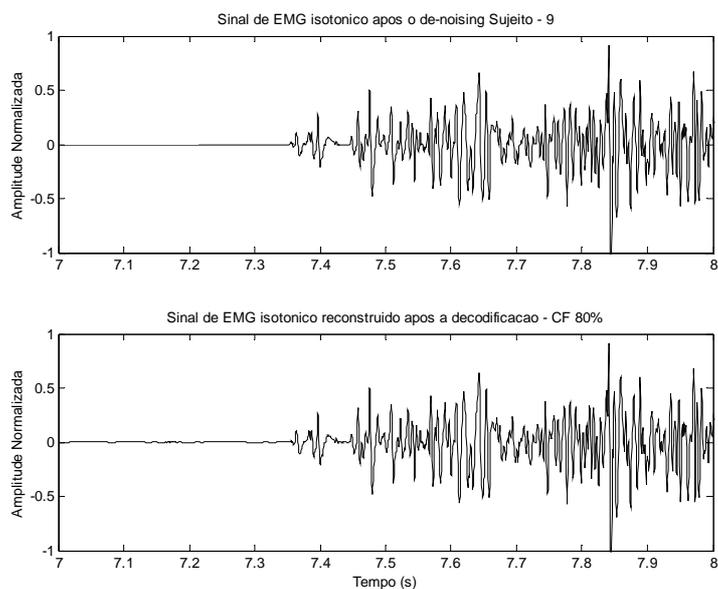


Figura C.23 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 23.

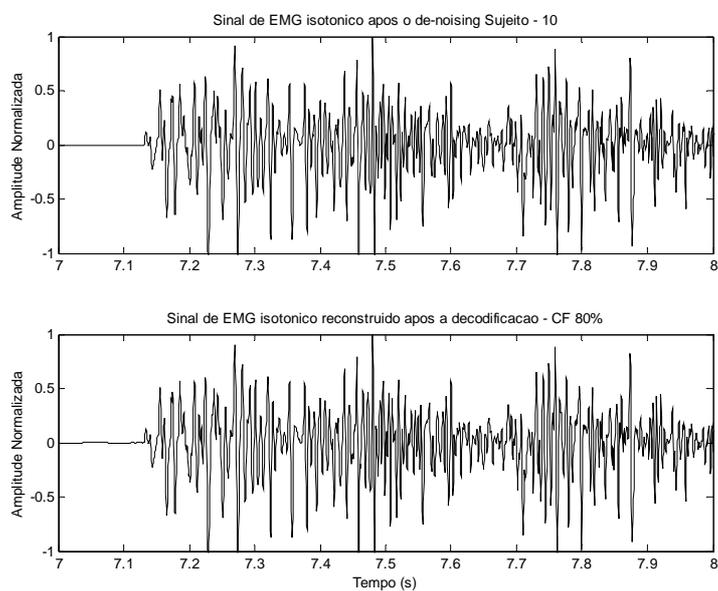


Figura C.24 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 24.

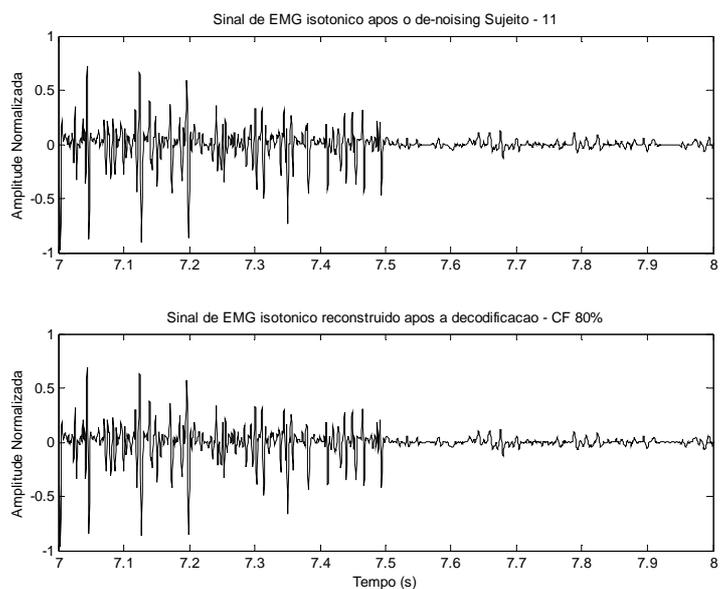


Figura C.25 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 25.

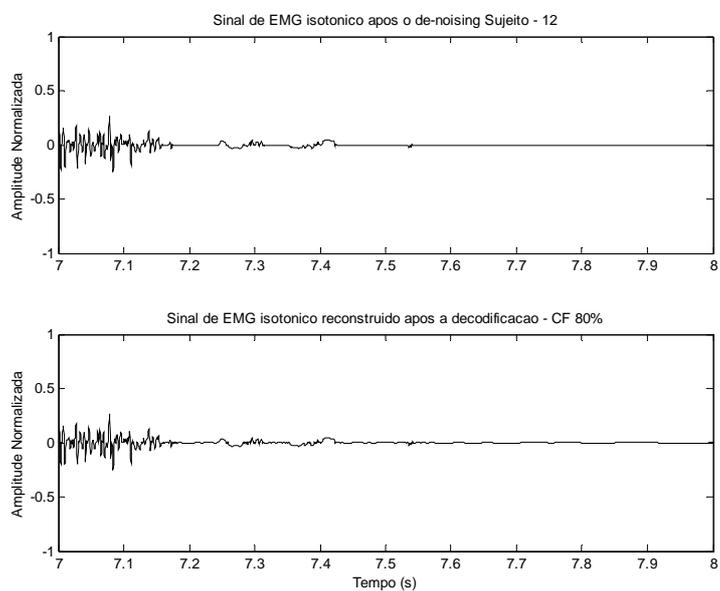


Figura C.26 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 26.

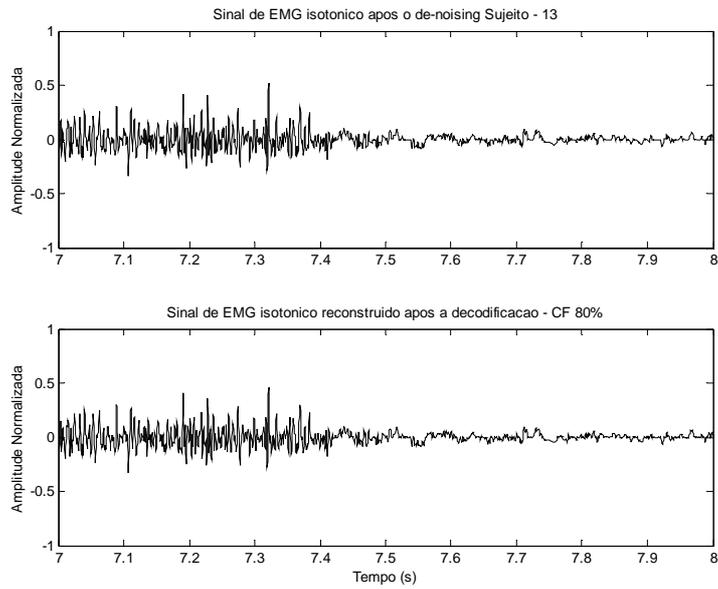


Figura C.27 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 27.

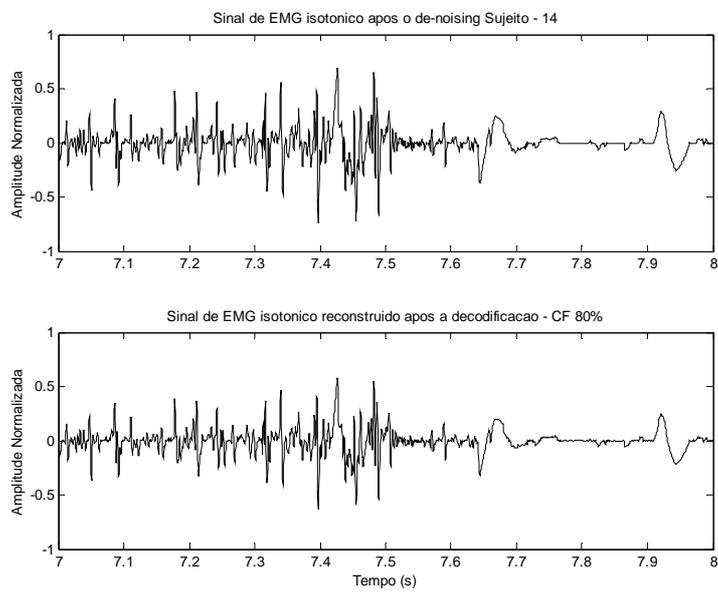


Figura C.28 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 28.

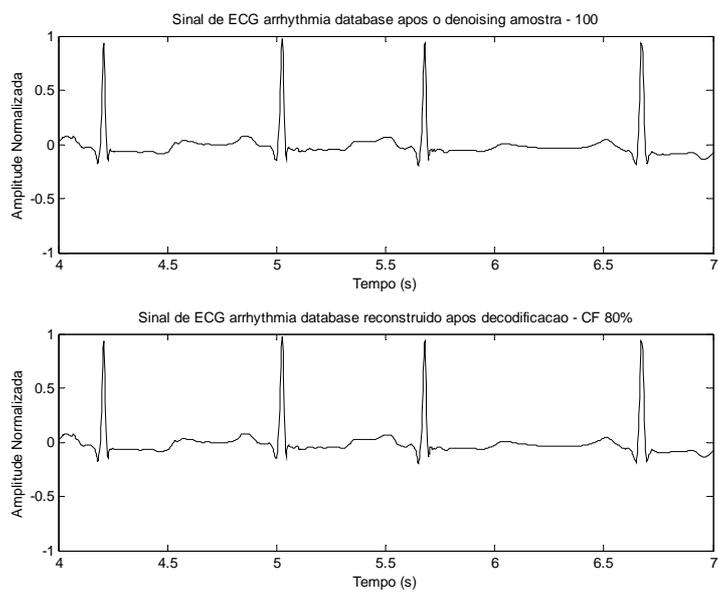


Figura C.29 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 29.

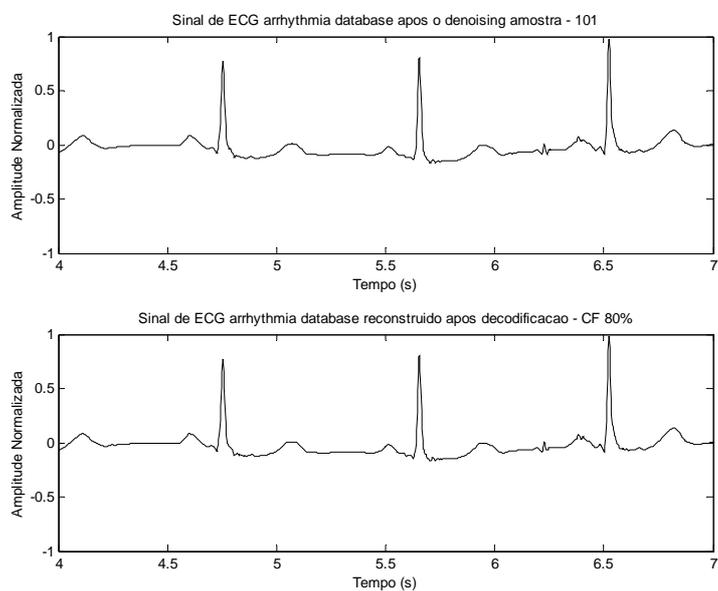


Figura C.30 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 30.

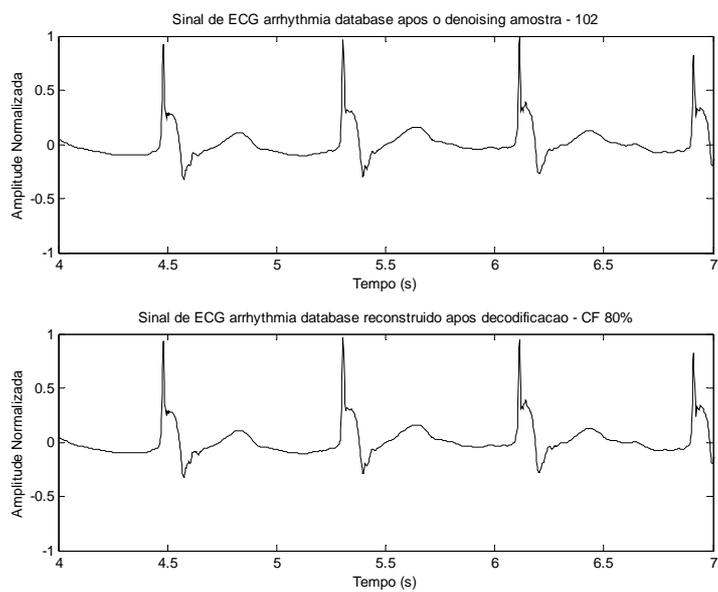


Figura C.31 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 31.

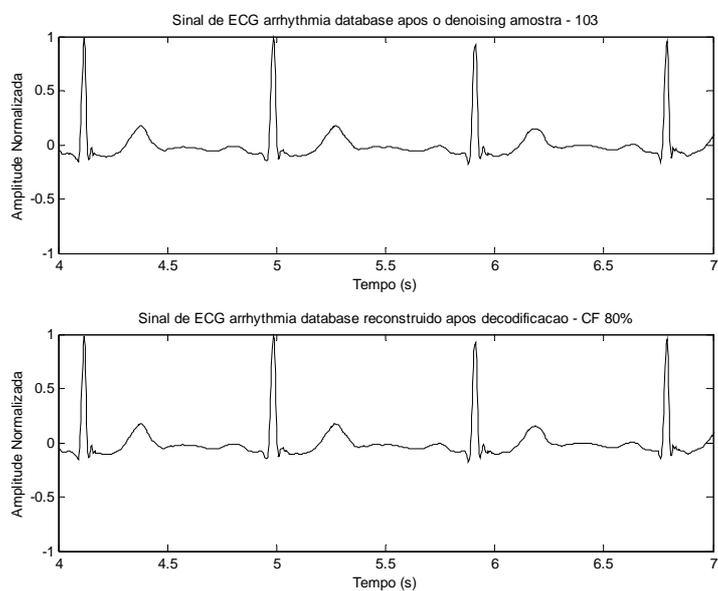


Figura C.32 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 32.

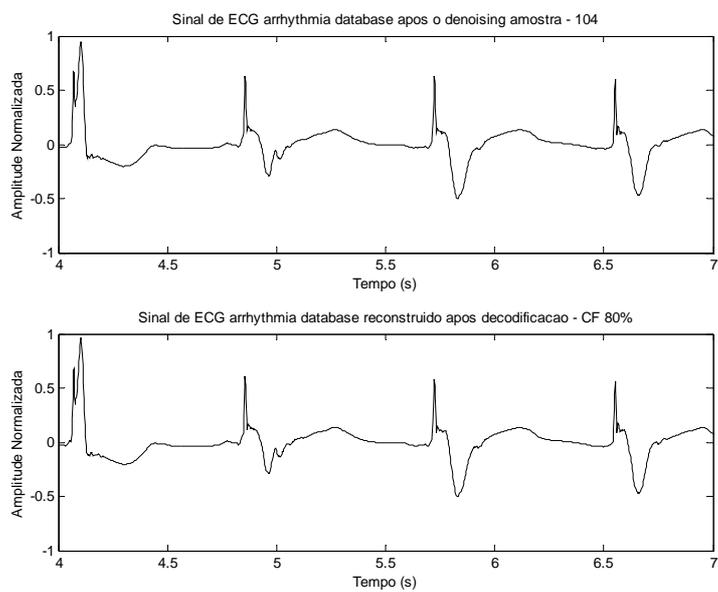


Figura C.33 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 33.

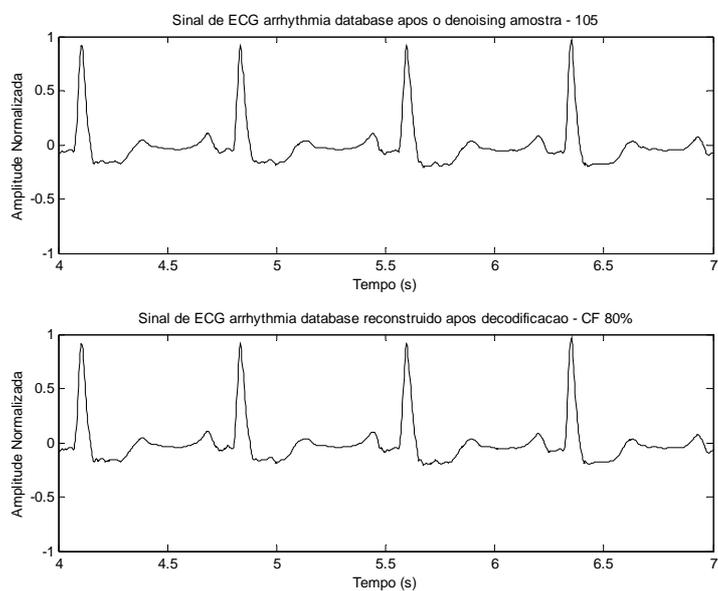


Figura C.34 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 34.

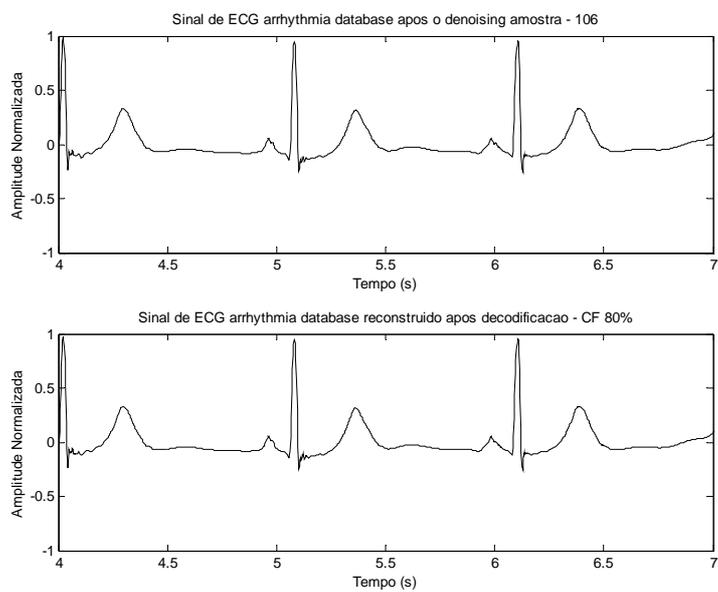


Figura C.35 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 35.

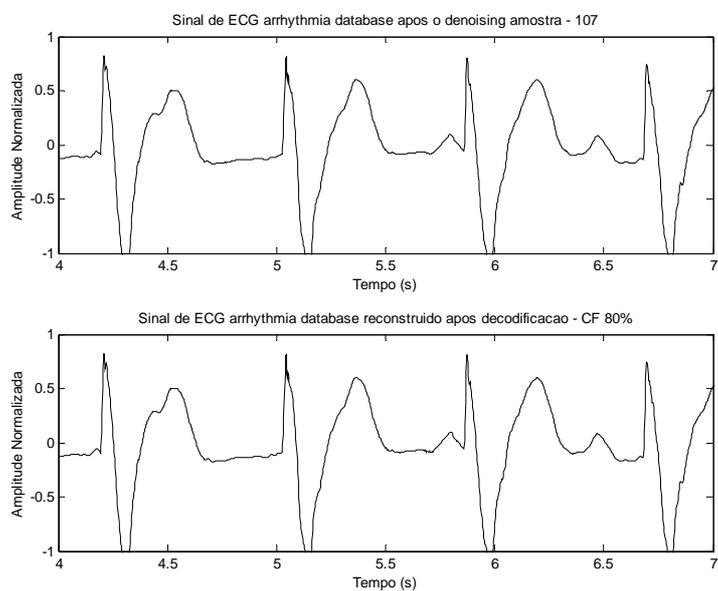


Figura C.36 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 36.

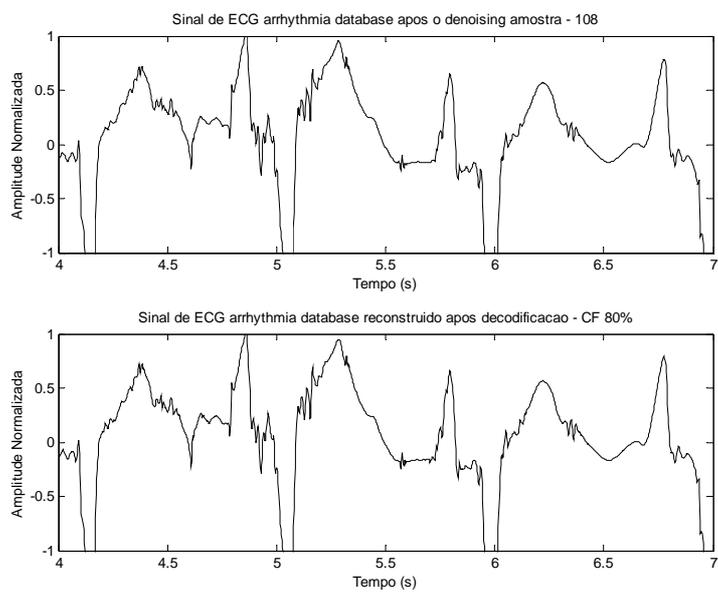


Figura C.37 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 37.

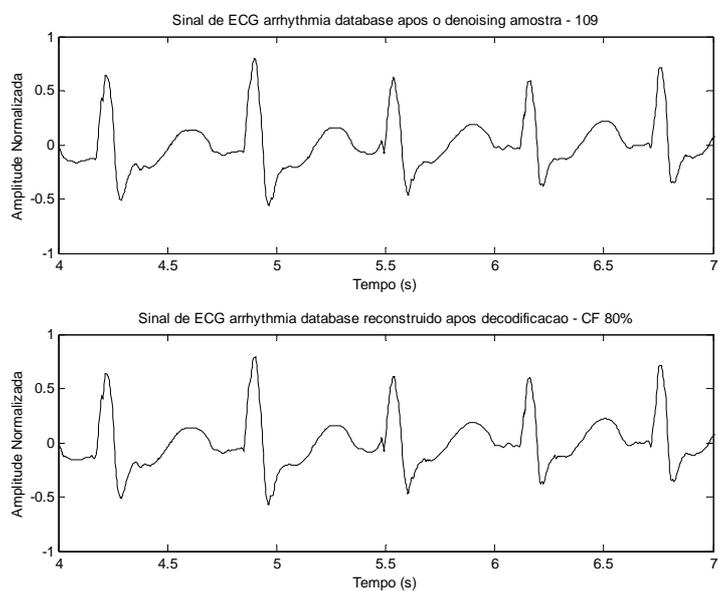


Figura C.38 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 38.

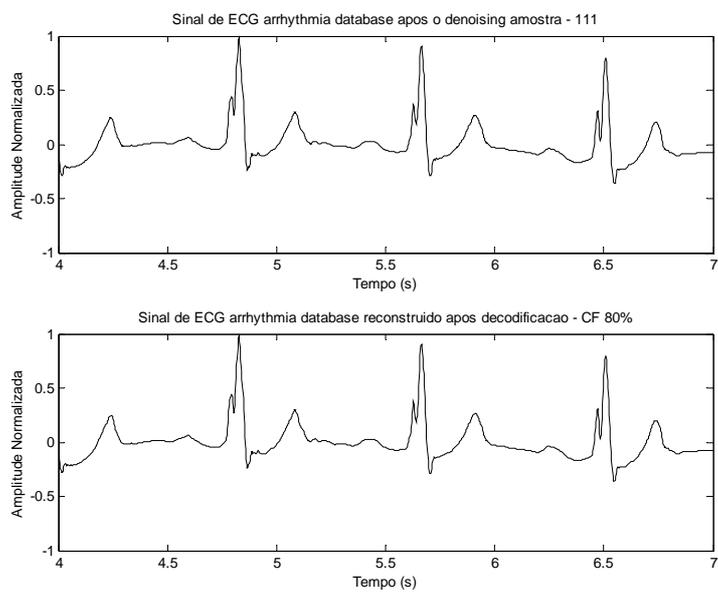


Figura C.39 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 39.

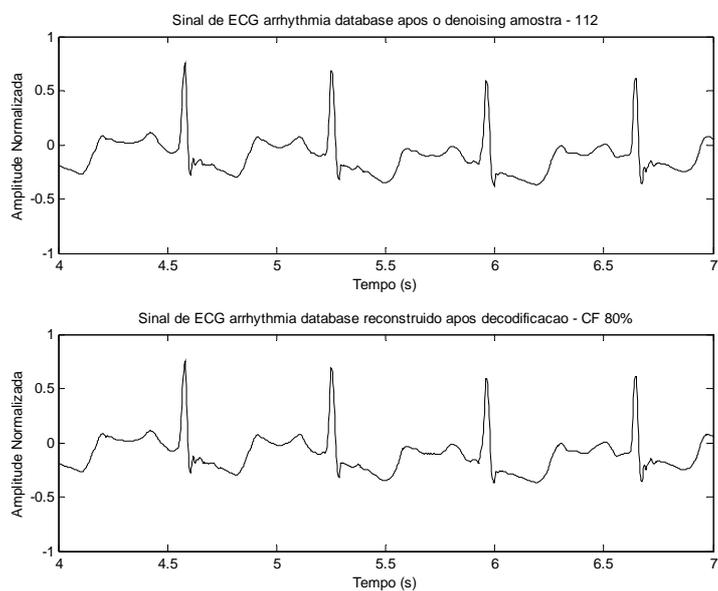


Figura C.40 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 40.

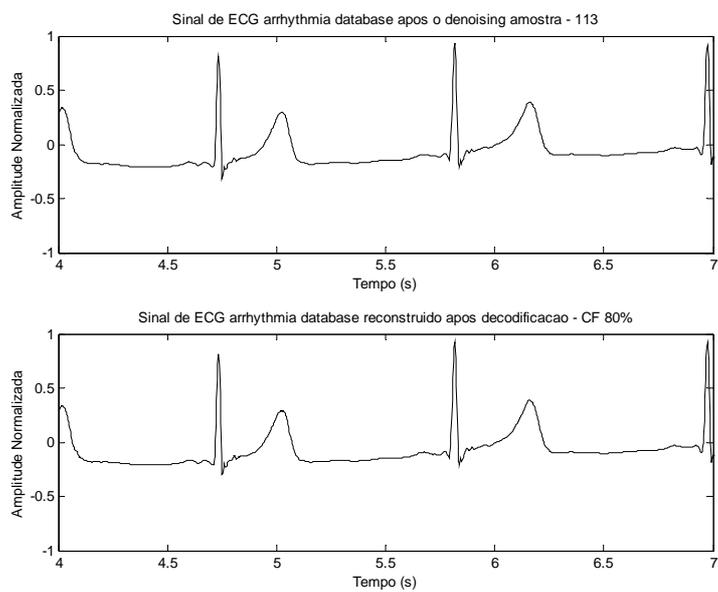


Figura C.41 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 41.

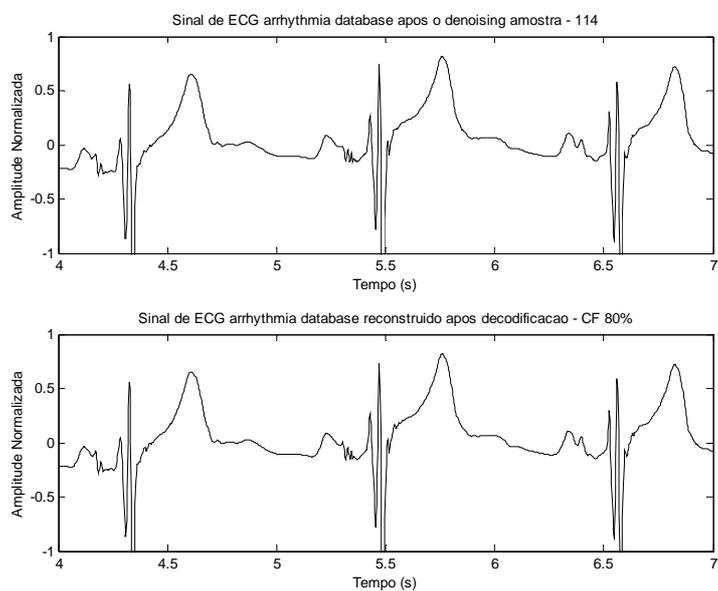


Figura C.42 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 42.

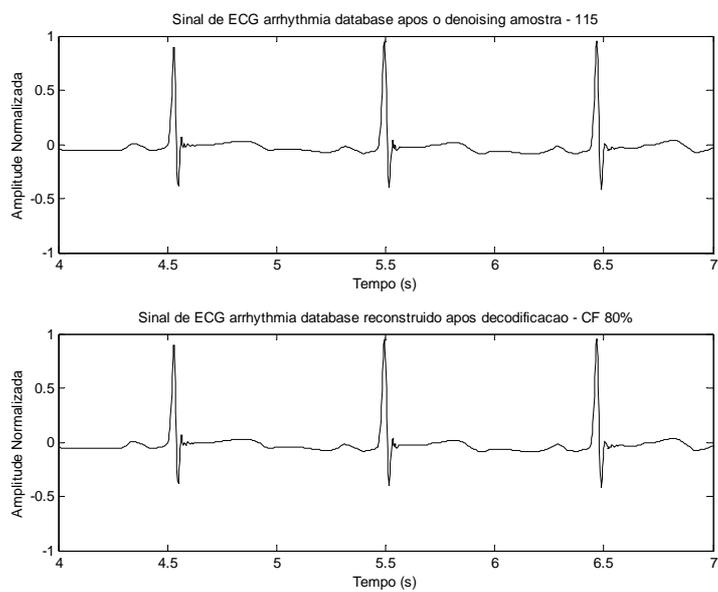


Figura C.43 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 43.

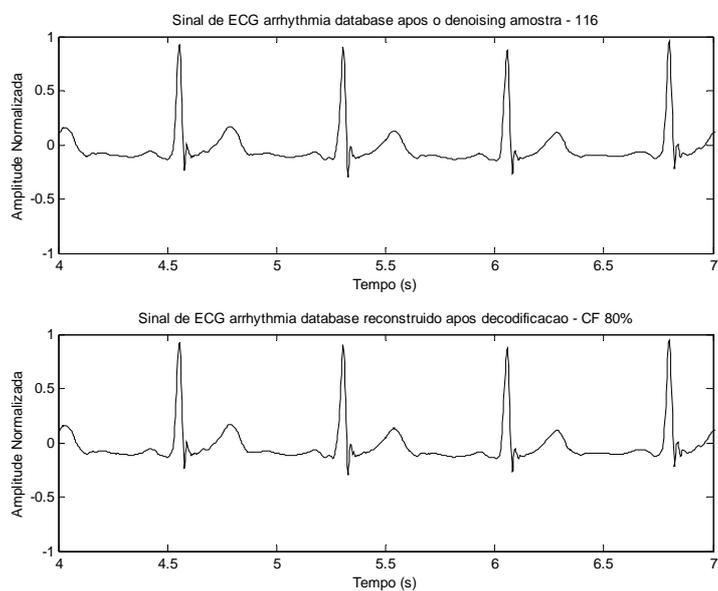


Figura C.44 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 44.

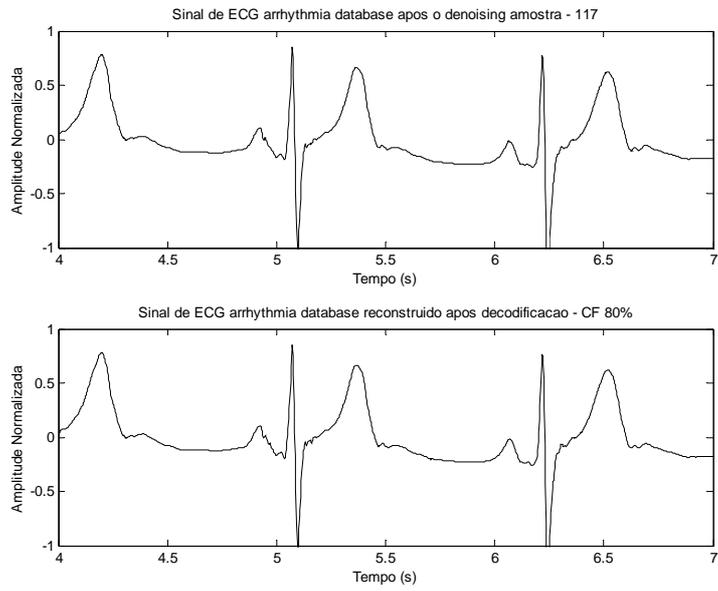


Figura C.45 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 45.

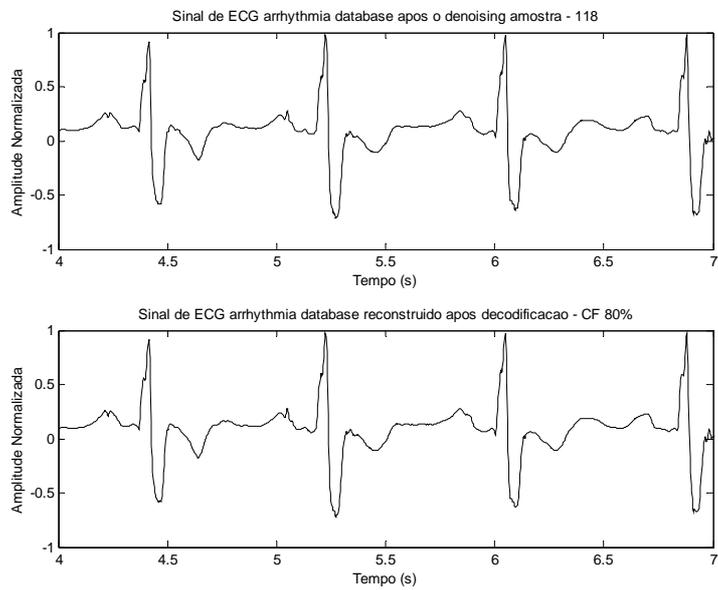


Figura C.46 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 46.

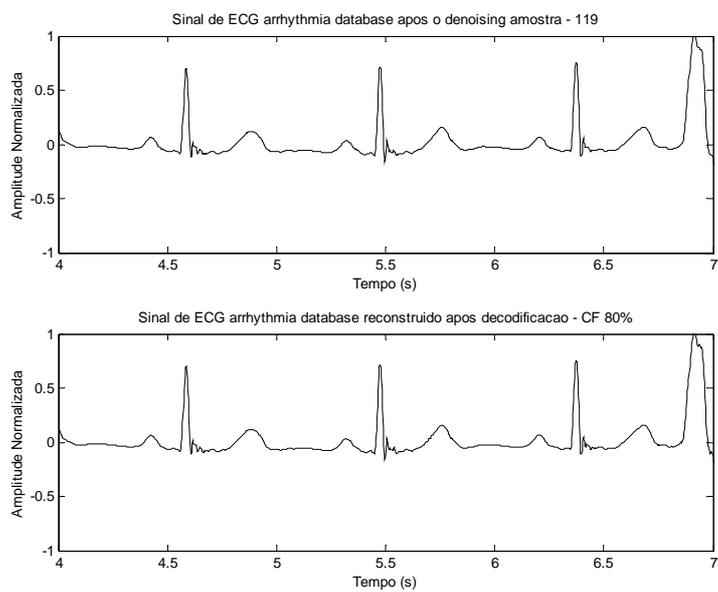


Figura C.47 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 47.

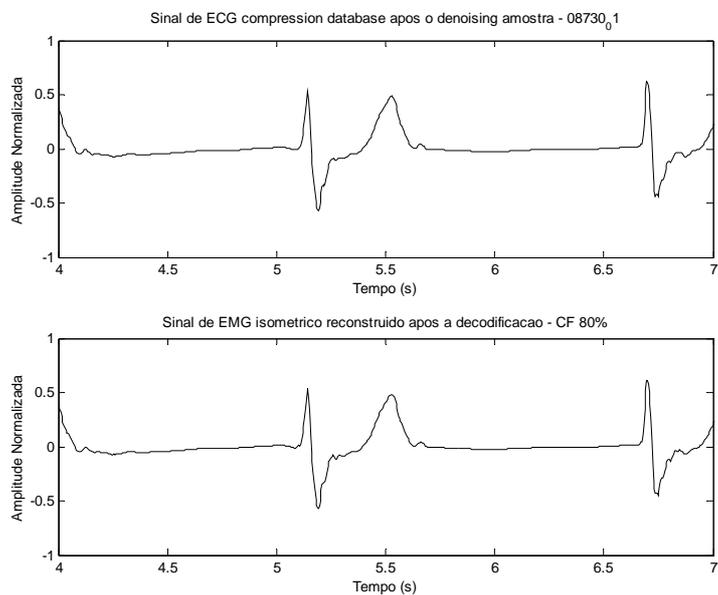


Figura C.48 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 48.

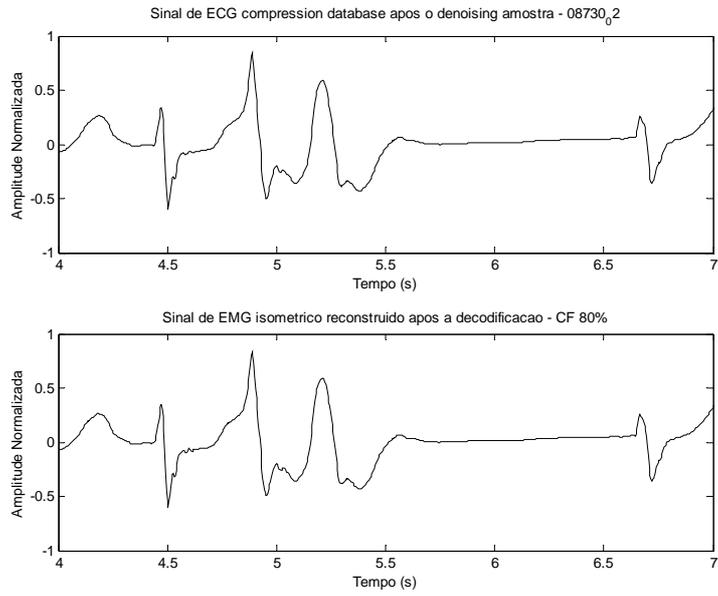


Figura C.49 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 49.

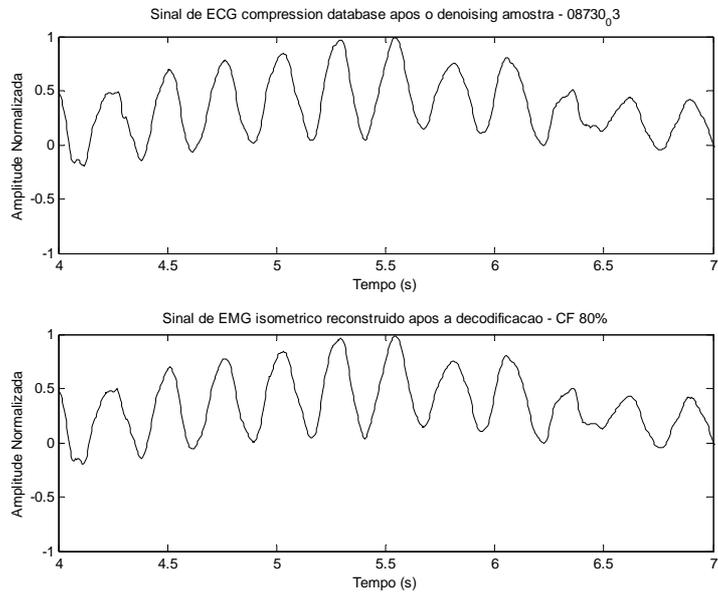


Figura C.50 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 50.

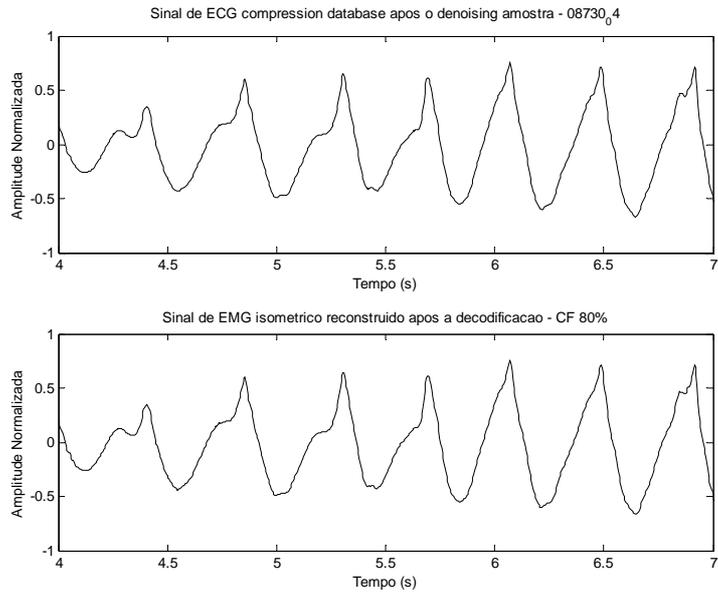


Figura C.51 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 51.

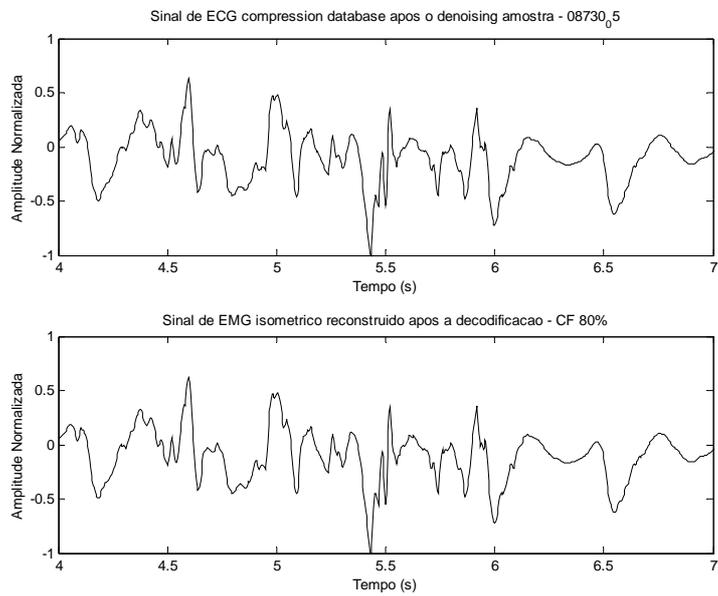


Figura C.52 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 52.

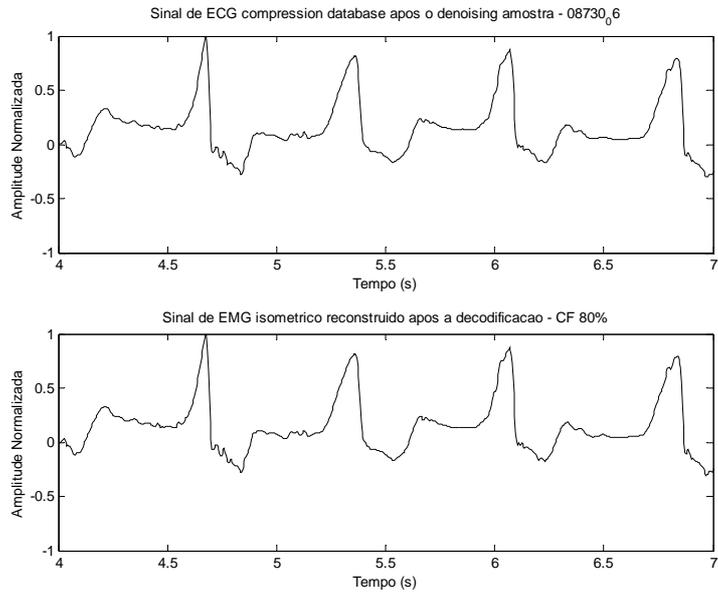


Figura C.53 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 53.

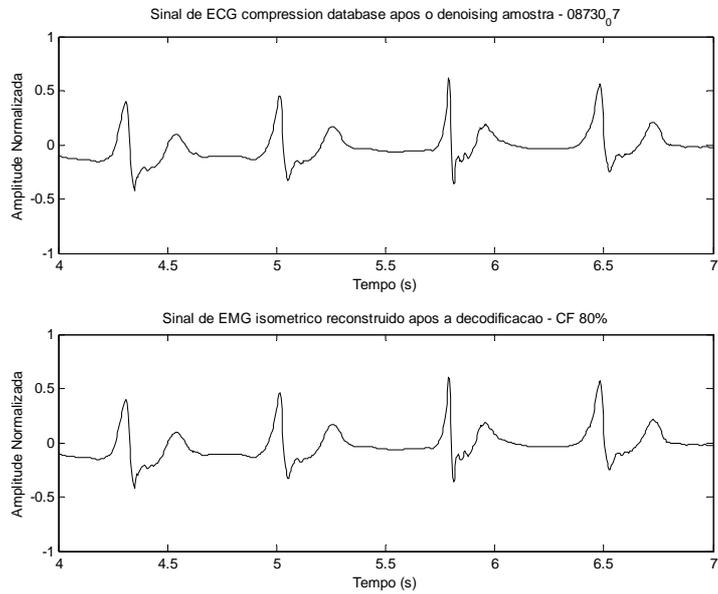


Figura C.54 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 54.

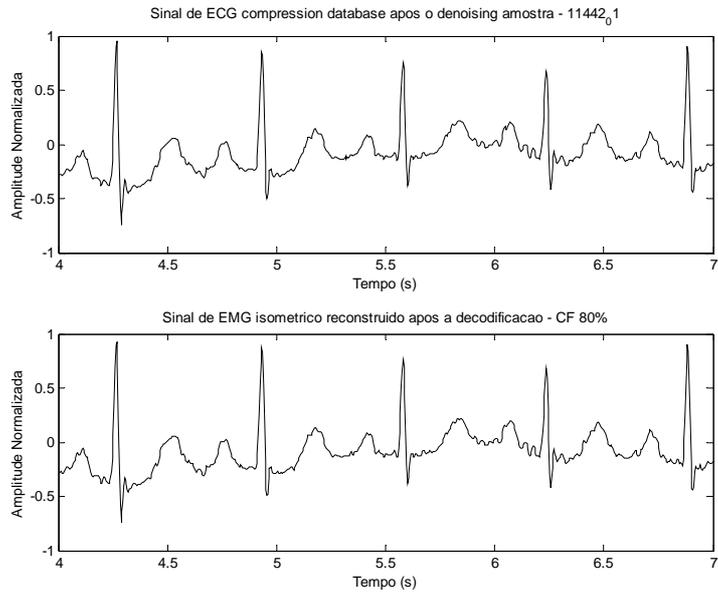


Figura C.55 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 55.

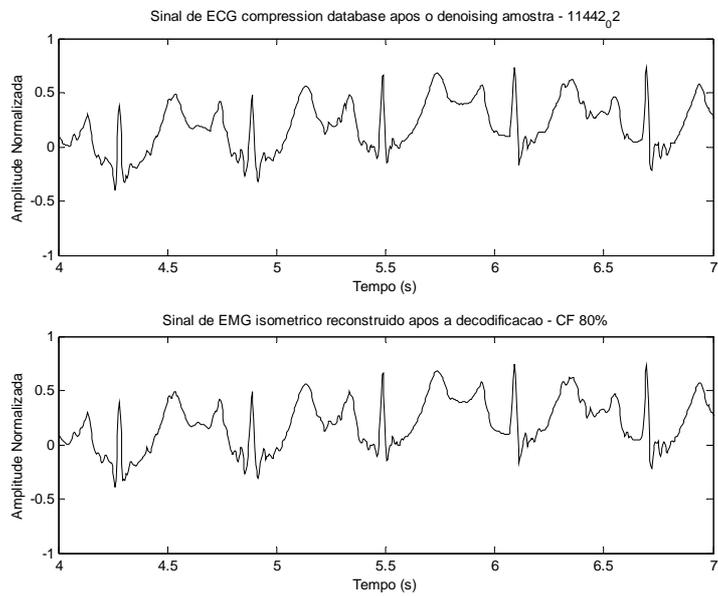


Figura C.56 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n<sup>o</sup> 56.

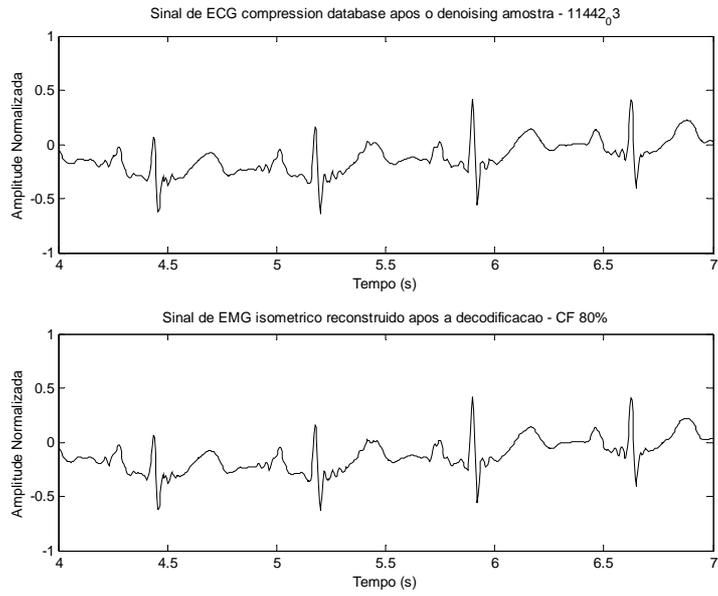


Figura C.57 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 57.

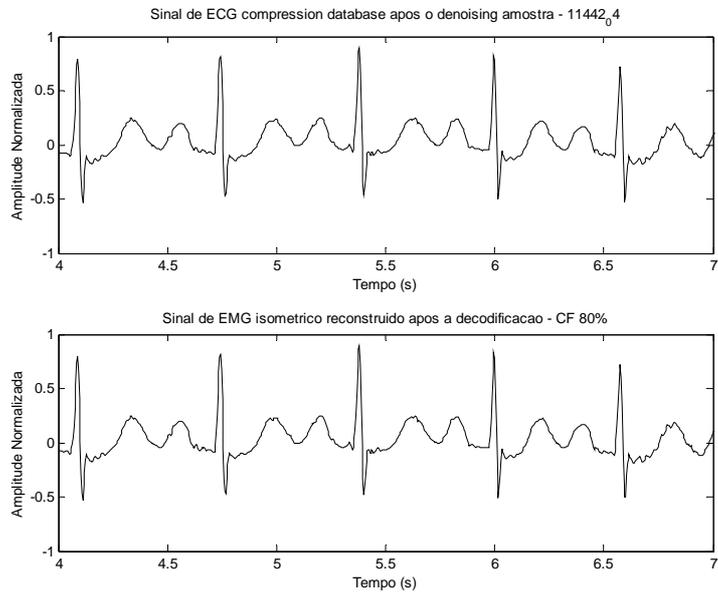


Figura C.58 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 58.

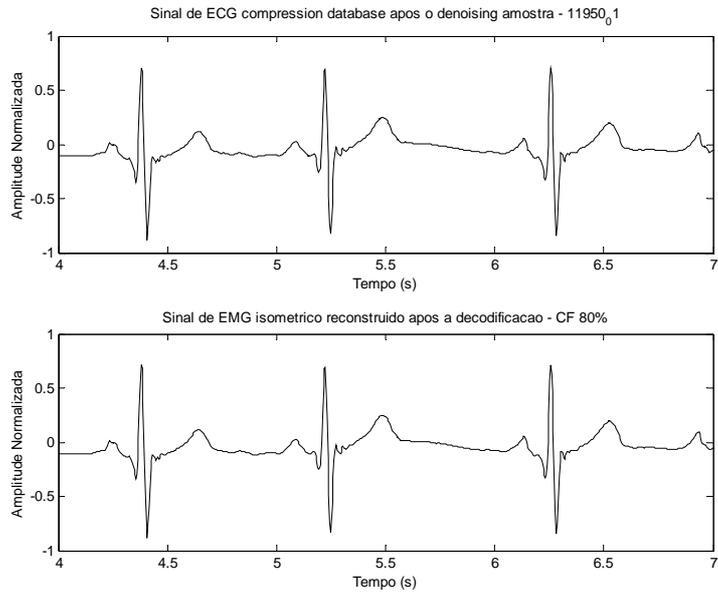


Figura C.59 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 59.

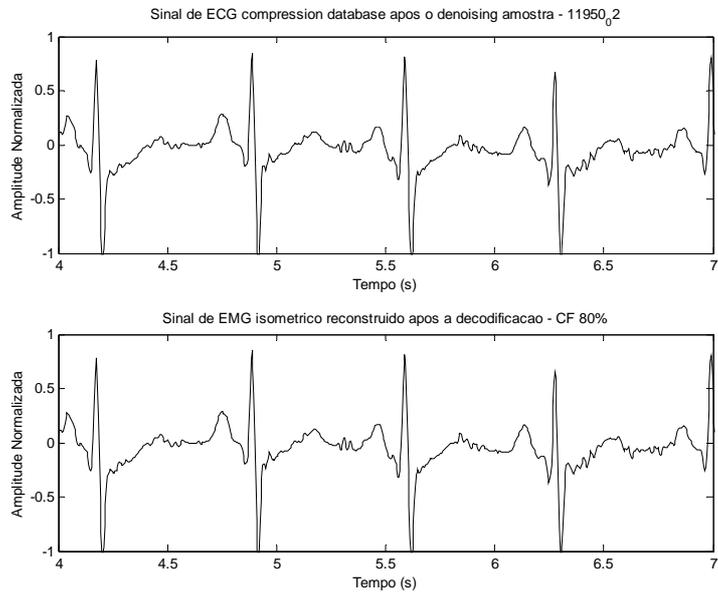


Figura C.60 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 60.

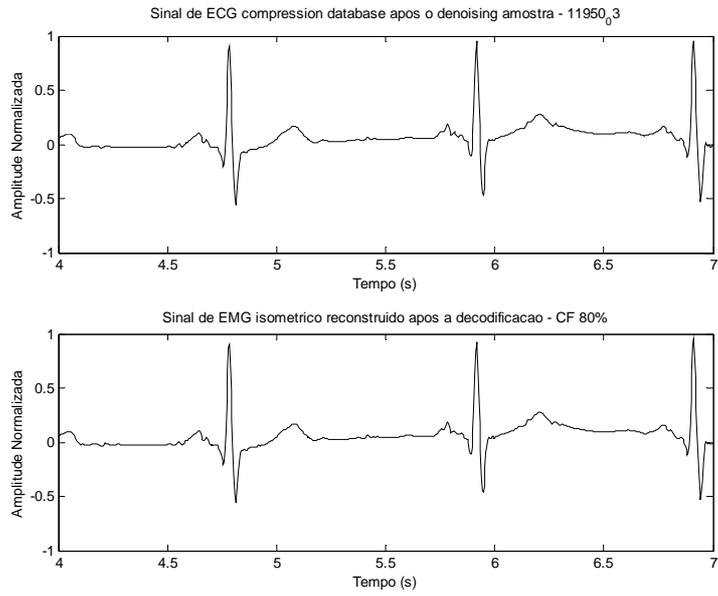


Figura C.61 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 61.

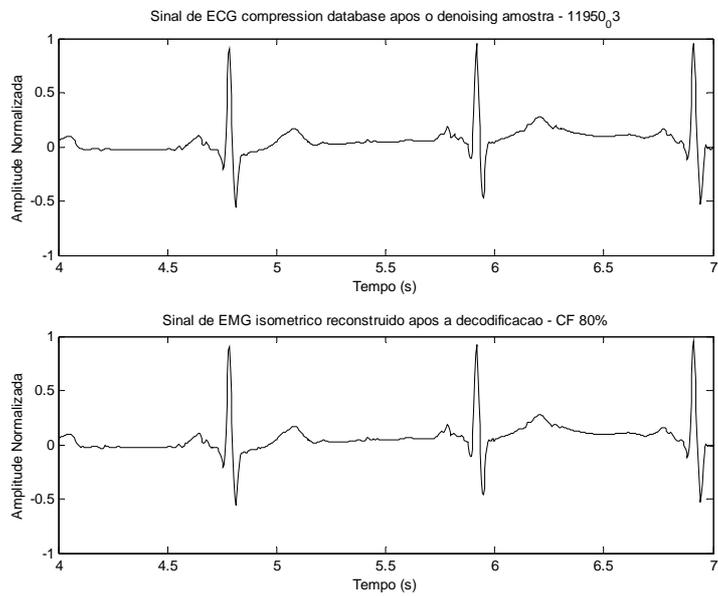


Figura C.62 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 62.

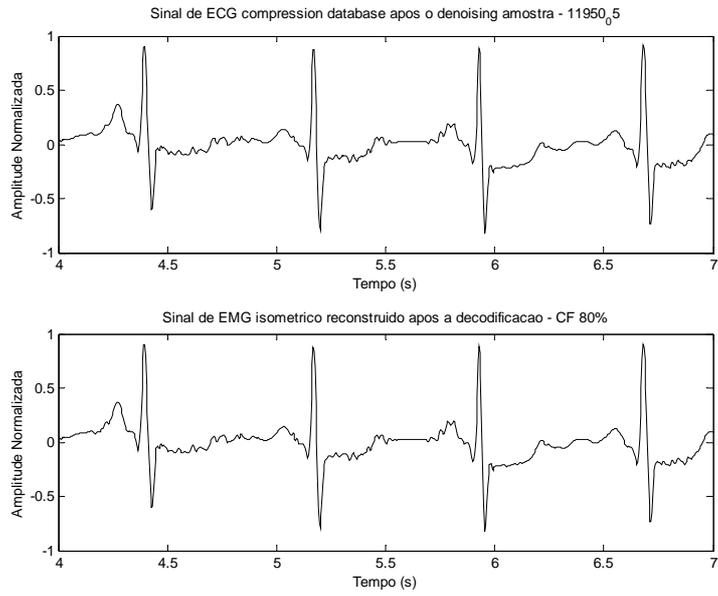


Figura C.63 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 63.

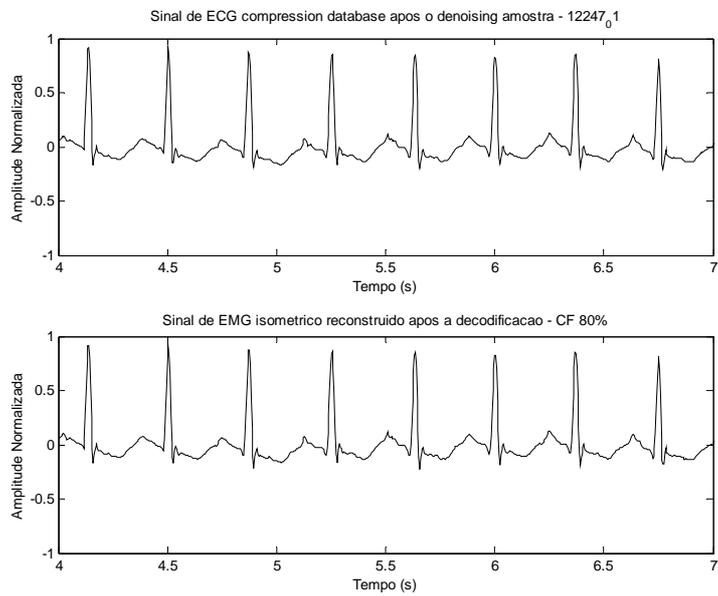


Figura C.64 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 64.

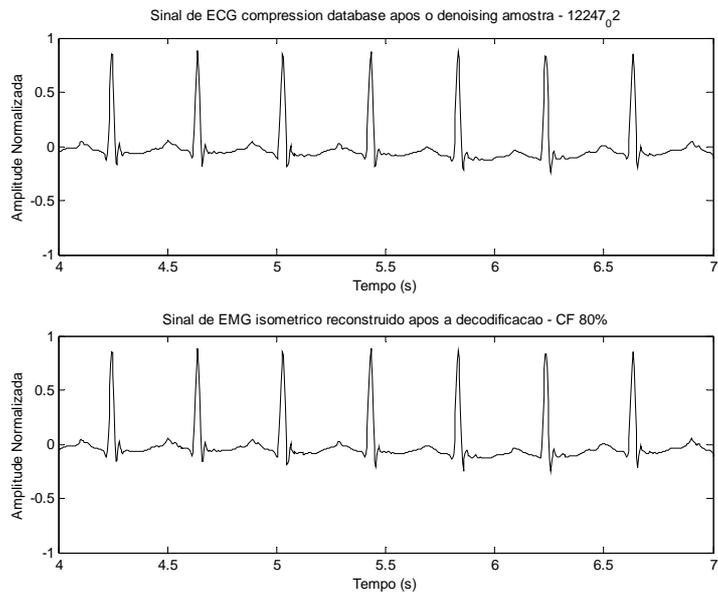


Figura C.65 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 65.

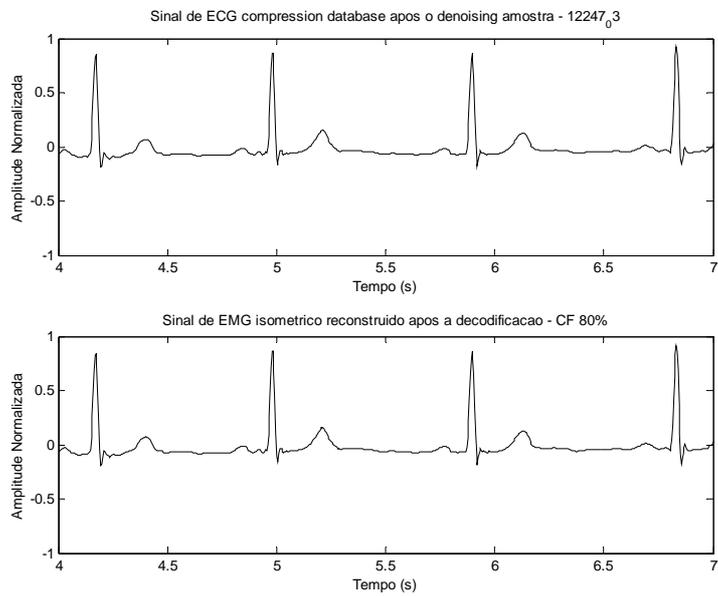


Figura C.66 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n ° 66.

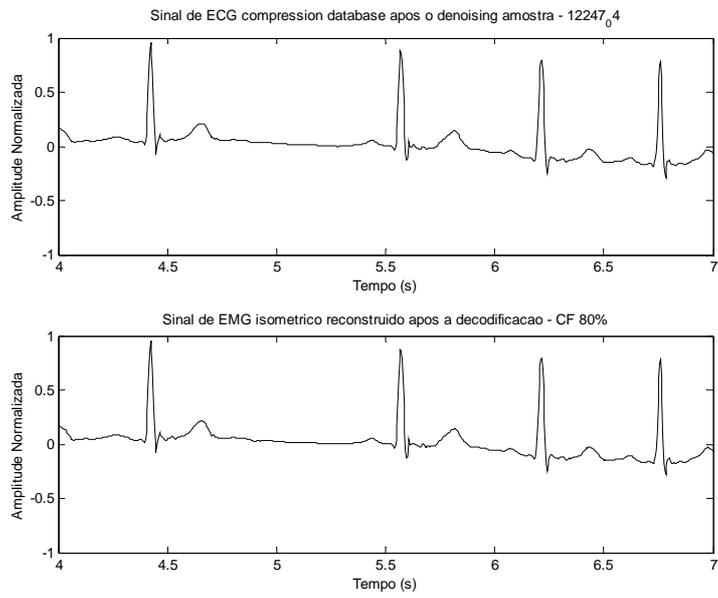


Figura C.67 – Resultado da compressão de sinais eletrofisiológicos com *de-noising* n° 67.

## D - RESULTADOS DA COMPRESSÃO DE VOZ DE BANDA LARGA

Tabela D.1 – Resultados PESQ para 32 kbps.

Indivíduo	PESQ
1	4.132
2	4.294
3	3.63
4	4.302
5	4.08
6	4.144
7	3.848
8	4.325
9	4.302
10	4.26
Média	4.1317

Tabela D.2 – Resultados PESQ para 24 kbps.

Indivíduo	PESQ
1	4.093
2	4.322
3	3.626
4	4.298
5	4.048
6	4.168
7	3.823
8	4.317
9	4.285
10	4.26
Média	4.124

Tabela D.3 – Resultados PESQ para 16 kbps

Indivíduo	PESQ
1	3.443
2	3.505
3	3.311
4	3.574
5	3.334
6	3.603
7	3.265
8	3.596
9	3.559
10	3.597
Média	3.4787

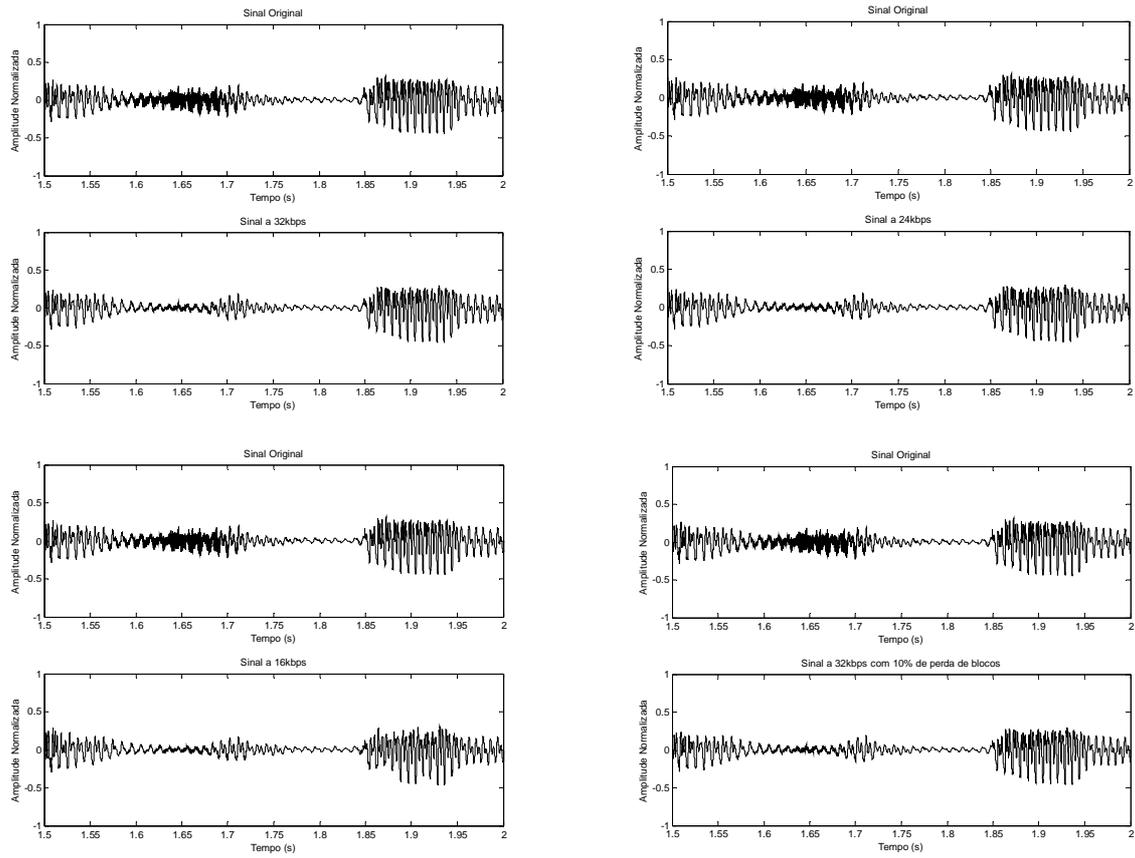


Figura D.1 – Exemplos de formas de onda para os sinais de voz codificados a 32, 24 e 16 kbps.

## **E - DETALHES SOBRE O CODIFICADOR AC-3**

### **E.1 - INTRODUÇÃO**

O AC-3 é um padrão de codificação de áudio de alta fidelidade largamente utilizado em aplicações comerciais como a HDTV e DVD. Nesse apêndice, estudamos e descrevemos em detalhes o sistema AC-3 na codificação de sinais de áudio.

### **E.2 - O SISTEMA AC-3**

O codificador AC-3 é atualmente o padrão de codificação de áudio adotado pelo sistema de HDTV da *United States Grand Alliance* e é largamente utilizado em filmes de DVD. A transmissão de áudio nos sistemas HDTV e DVD é feita através de 6 canais (considerando o formato de 5.1 canais de áudio). O formato 5.1 proporciona o bom desempenho dos sons fortes como explosões, motores e efeitos especiais. No sistema HDTV, a frequência de amostragem é de 48 kHz por canal, o que demandaria (sem a codificação) uma taxa de aproximadamente 5 Mbits/s para a transmissão dos 5.1 canais. Para diminuir essa taxa esse sistema oferece um potente mecanismo de compressão baseado na sensibilidade do ouvido humano, que é diferente para as diversas componentes de frequência que formam o espectro de audição. Isso permite designar os níveis de quantização adequados, de acordo com o nível de percepção de uma determinada componente em frequência. Assim, as degradações que porventura o sinal vier a sofrer não seriam percebidas pelo sistema de audição. Com a aplicação desses critérios o codificador AC-3 é capaz de transmitir os 5.1 canais de áudio a uma taxa de 384 kbit/s, conservando os níveis de qualidade de áudio exigidos para os sistemas HDTV e DVD.

O processo de codificação do AC-3 pode ser ilustrado pela Figura E.1. A seqüência de áudio é transformada para o domínio da frequência usando a MDCT (do inglês, *Modified Discrete Cosine Transform*). A MDCT funciona como um banco de filtros que separa o sinal de áudio em bandas iguais de frequência. Cada linha espectral é representada como expoentes e mantissas de um formato em ponto flutuante. Os expoentes são codificados diferencialmente e alimentam o modelo psico-acústico. O modelo psico-acústico calcula a resolução perceptiva usando os expoentes codificados (envelope espectral) e alguns parâmetros perceptivos. Os parâmetros perceptivos de entrada do codificador podem ser otimizados através de um sistema de análise por síntese ou serem predeterminados em uma implementação mais simples do

codificador. Finalmente, a resolução perceptiva e a quantidade de bits disponíveis são usadas para calcular a alocação de bits e decidir a estratégia de quantização das mantissas.

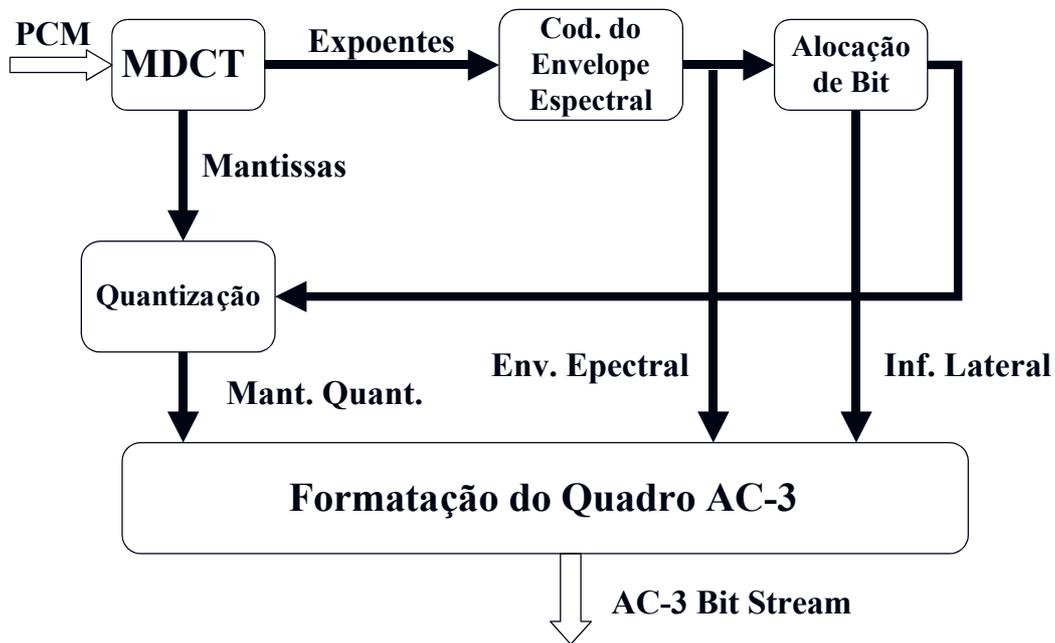


Figura E.1 – O codificador AC-3.

O decodificador AC-3 tem o mesmo modelo psico-acústico do codificador e usa o envelope espectral (expoentes codificados) e os parâmetros psico-acústicos, que lhe são transmitidos, para calcular a alocação de bits das mantissas e realizar a decodificação das mesmas. Uma vez que as mantissas e os expoentes são decodificados, é realizada a IMDCT (do inglês *Inverse Modified Discrete Cosine Transform*) resultando no sinal de áudio decodificado. Esse processo pode ser ilustrado pelo diagrama de blocos do decodificador na Figura E.2.

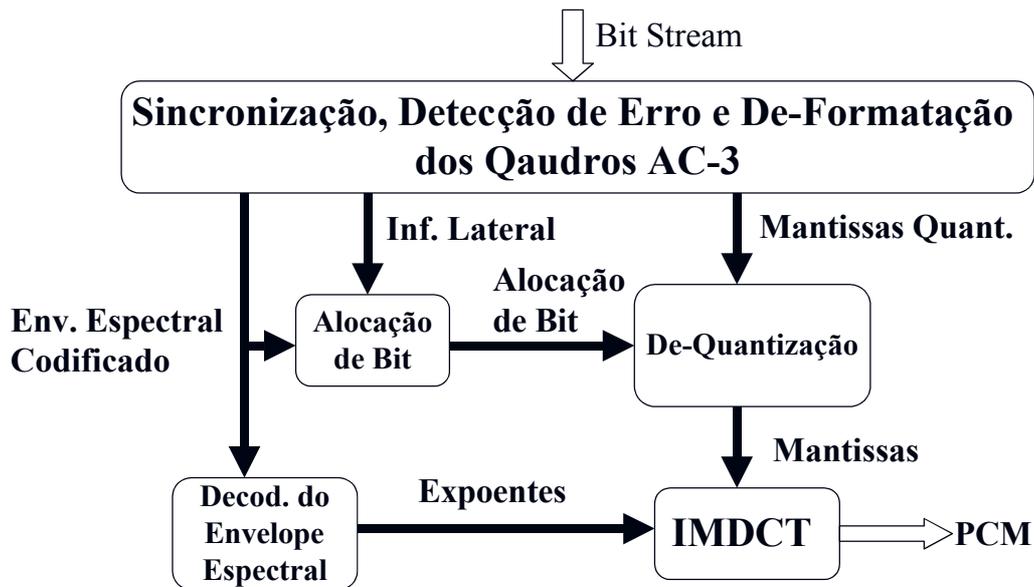


Figura E.2 – O decodificador AC-3.

É importante ressaltar que o decodificador AC-3 possui um modelo psico-acústico idêntico ao do codificador, e por isso o decodificador realiza os cálculos da alocação de bits com os mesmos resultados do codificador, dessa forma não existe a necessidade de se transmitir a alocação de bits do sinal. A transmissão dos parâmetros perceptivos é uma estratégia híbrida e permite uma grande flexibilidade ao sistema, além de diminuir o processamento necessário no decodificador.

A seguir descrevemos em detalhes o codec AC-3. Os detalhes do decodificador podem ser deduzidos de uma análise direta do codificador, por isso discutimos os dois indistintamente, a menos de algumas raras diferenças que serão consideradas oportunamente.

### E.3 - DETALHES DO CODIFICADOR AC-3

O codificador AC-3 transmite ao decodificador uma série de parâmetros do modelo psico-acústico que são calculados e adaptados durante o processo de codificação. Isso significa que a especificação do decodificador viabiliza diferentes implementações do codificador e, que seu comportamento pode variar durante o tempo sem abalar a compatibilidade entre sistemas diferentes. A única restrição para a implementação de um codificador AC-3 é que os bits na saída do codificador devem ser armazenados e/ou transmitidos seguindo a sintaxe especificada no decodificador. Por isso, o documento A/52 só especifica uma implementação simples do codificador (chamado de *core encoder*) que pode ser aprimorada para atender as

necessidades de várias aplicações. Os detalhes do codec que descrevemos a seguir são os especificados para o *core encoder*.

#### E.4 - A MDCT

No codificador AC-3, a transformada do sinal de áudio para o domínio da frequência é realizada através da MDCT que toma blocos no domínio do tempo de  $L=512$  amostras com superposição de  $M=256$  amostras. Cada bloco de áudio  $x(n)$  é multiplicado por uma janela  $w(n)$  do tipo Kaiser-Bessel derivada, mostrada na Figura E.3.

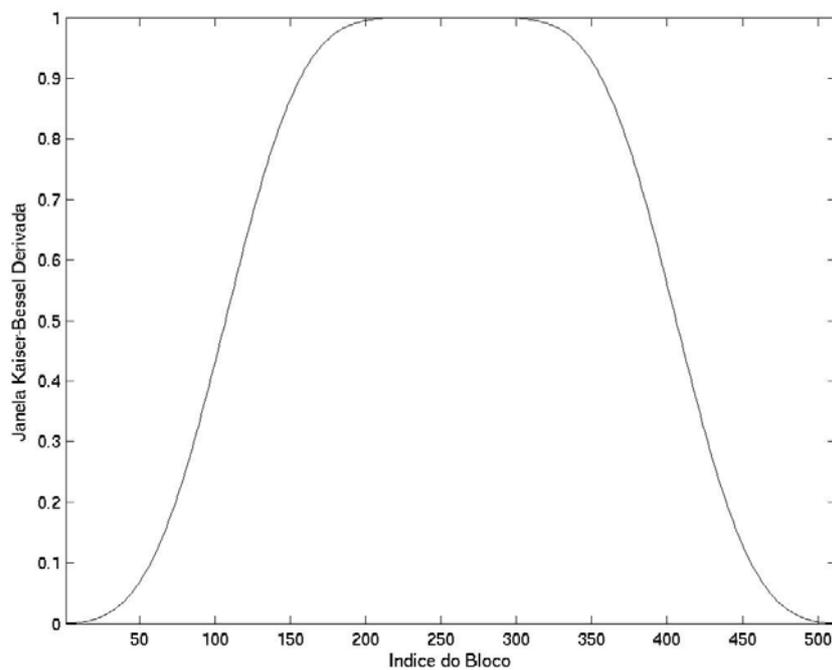


Figura E.3 – Janela aplicada ao sinal de áudio pelo sistema AC-3.

As funções de base da MDCT se estendem por dois blocos no tempo, minimizando o efeito de bloco que afeta consideravelmente os codificadores com transformadas sem superposição. Apesar da superposição de 50% a MDCT é criticamente amostrada (somente  $M$  coeficientes transformados são gerados para cada  $2M$  amostras de entrada). Dado um bloco de entrada  $x(n)$ , os coeficientes transformados  $X(j)$  são definidos por:

$$X(j) = \sum_{n=0}^{2M-1} x(n)h_j(n), \text{ onde:} \tag{E.1}$$

$$h_j(n) = w(n) \sqrt{\frac{2}{M}} \cos \left[ \frac{(2n+M+1)(2j+1)\pi}{4M} \right], \quad 0 \leq j \leq M-1$$

A MDCT realiza uma série de produtos internos entre a resposta ao impulso de um banco de filtros de análise,  $h_j(n)$ , e a entrada  $x(n)$ . A IMDCT resulta em uma reconstrução perfeita do bloco de entrada, calculando uma soma dos vetores de base (ou respostas ao impulso  $h_j(n)$  do banco de filtros de análise) ponderados pelos coeficientes transformados de dois blocos. As primeiras  $M$  amostras do  $j$ -ésimo vetor de base, para  $0 \leq n \leq M-1$ , são multiplicadas pelo  $j$ -ésimo coeficiente do bloco transformado corrente,  $X(j)$ . Simultaneamente, as  $M$  amostras seguintes, são multiplicadas pelo  $j$ -ésimo coeficiente do bloco transformado anterior,  $X^P(j)$ . Em seguida, os vetores de base são somados com superposição de 50% em cada instante de tempo  $n$ . Então as amostras reconstruídas  $x(n)$ , para  $0 \leq n \leq M-1$ , são obtidas pela IMDCT como:

$$x(n) = \sum_{j=0}^{M-1} [X(j)h_j(n) + X^P(j)h_j(n+M)] \tag{E.2}$$

A Figura E.4 ilustra este procedimento.

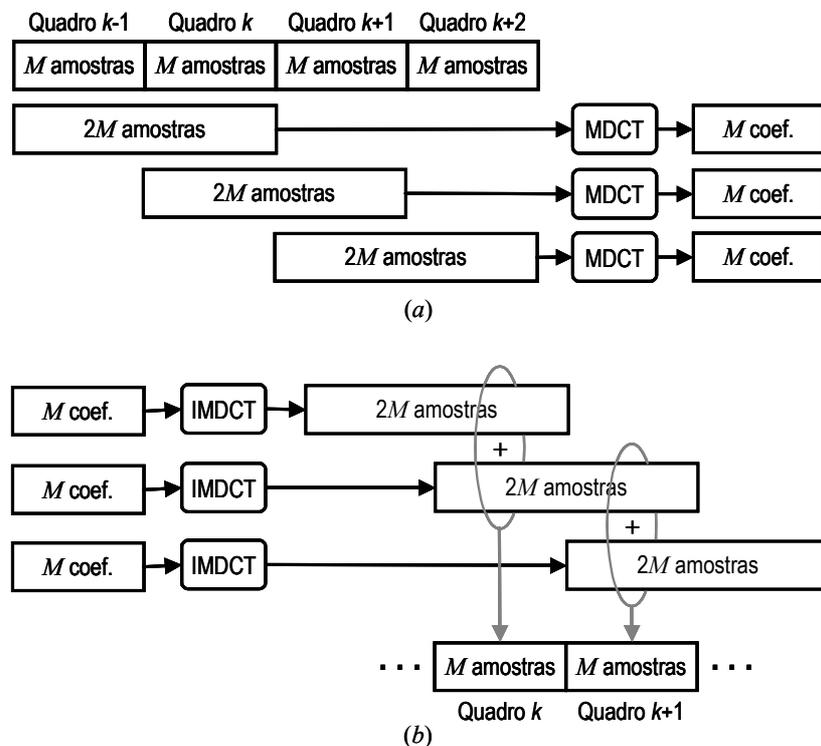


Figura E.4 – MDCT e IMDCT

A Figura E.5 mostra o resultado da MDCT aplicada a um bloco de 512 amostras de sinal de áudio amostrado a 44.1 kHz.

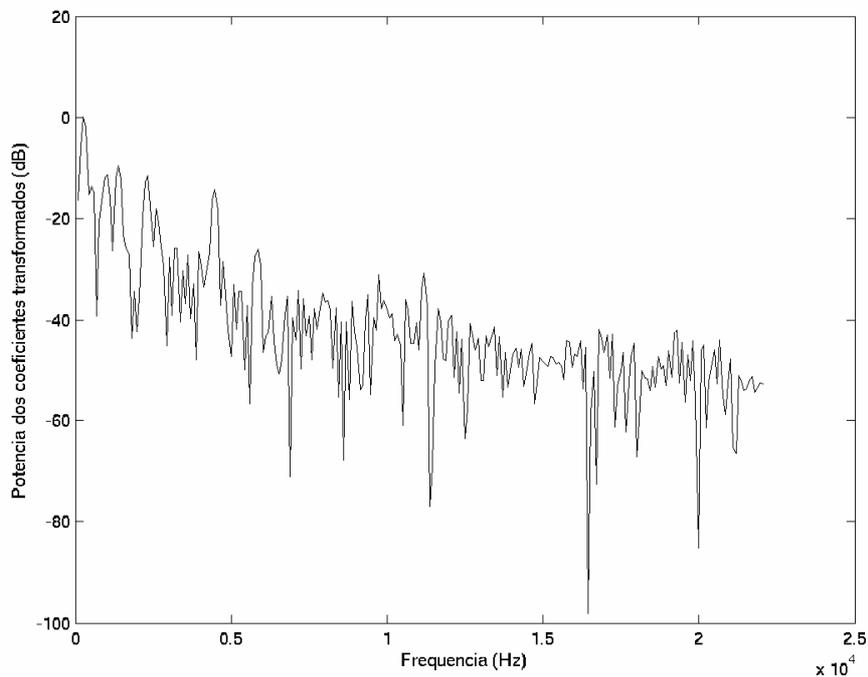


Figura E.5 – Exemplo da MDCT

Após a transformação do sinal de áudio para o domínio da frequência, cada coeficiente MDCT é representado por números reais cujos valores são escalonados na faixa de  $-1$  a  $1$ . O formato empregado pelo AC-3 para a representação desse tipo de números é a notação em ponto flutuante. Assim, um coeficiente com valor  $0.000101011000100_2$  expresso em sistema binário e com precisão de 16 bits pode ser representado em formato ponto flutuante como sendo  $2^{-3} \times 0.1101011000100_2$ ; onde o número de zeros depois do ponto decimal representa o valor do expoente que neste caso é  $-3$ . Os 13 bits consecutivos depois do ponto decimal representam o valor da mantissa normalizada que será posteriormente quantizada e codificada junto com o expoente. Além disso, acrescenta-se um bit de sinal para identificar valores positivos e negativos.

#### E.4.1 - Codificação dos expoentes

Cada coeficiente transformado é codificado em uma representação expoente e mantissa. O conjunto de expoentes codificado forma uma representação global do espectro do sinal, e é denominado envelope espectral. Apenas o primeiro expoente do bloco transformado é enviado

integralmente, codificado com uma palavra de 5 bits. Os demais expoentes são codificados diferencialmente, com deltas de 0,  $\pm 1$  ou  $\pm 2$ . Cada 3 deltas são codificados em uma única palavra de 7 bits. Desta forma, cada expoente requer aproximadamente 2.33 bits de código. Este método de codificação dos expoentes é chamado de D15. Quando o espectro do sinal não é muito estável, é de grande benefício enviar uma estimativa espectral com mais frequência do que para cada bloco de mantissas. Nestes casos opta-se por reduzir a resolução em frequência, resultando em mais dois métodos de codificação, chamados de D25 e D45. No método D25 é codificado diferencialmente cada par de expoentes e no método D45 é codificado diferencialmente cada grupo de 4. Por outro lado, quando o espectro do sinal é bem estável pode-se alcançar ganhos de compressão maior diminuindo a resolução no tempo, ou seja, enviando somente um envelope espectral para mais de um bloco de mantissas. Dessa forma, o mesmo conjunto de expoentes codificados é reutilizado para mais de um conjunto de mantissas. A reutilização dos expoentes pode chegar a 6 blocos (1 quadro) de coeficientes transformados. Na diminuição da resolução em frequência e na reutilização dos expoentes em blocos consecutivos, durante a codificação diferencial, as mantissas que têm seus coeficientes alterados devem ser escalonadas para compensar as mudanças feitas nos expoentes e preservar o valor do coeficiente transformado. Assim, o erro na codificação dos coeficientes transformados é controlado exclusivamente pela quantização das mantissas.

#### **E.4.2 - Alocação de Bits**

O cálculo da alocação de bits consiste de um algoritmo (modelo psico-acústico) cuja finalidade é determinar o número de níveis de quantização necessário para a codificação de cada mantissa. Nesse algoritmo, busca-se calcular um limiar de mascaramento que serve como limite superior para o ruído de quantização, ou seja, todas as componentes de ruído que forem inferior a esse limiar são encobertas pelo próprio sinal de áudio e não são percebidas pelo ouvido humano. A seguir mostramos em detalhes o modelo psico-acústico do AC-3.

#### **E.4.3 - Cálculo da Densidade Espectral de Potência**

O codificador calcula a DEP (Densidade Espectral de Potência) com base nos valores dos expoentes de cada coeficiente transformado. A função DEP é expressa através de valores de 13 bits com sinal e é calculada por:

$$DEP(j)=3072-\exp(j) \times 128 \quad (E.3)$$

onde  $j$  é o índice da componente em frequência, enquanto  $\exp(j)$  é o expoente do coeficiente  $j$ .

Os expoentes assumem valores de 0 a 24, por isso os valores da DEP calculados pela equação (E.3) resultam em valores entre 0 e 3072, conforme mostrado na Figura E.6.

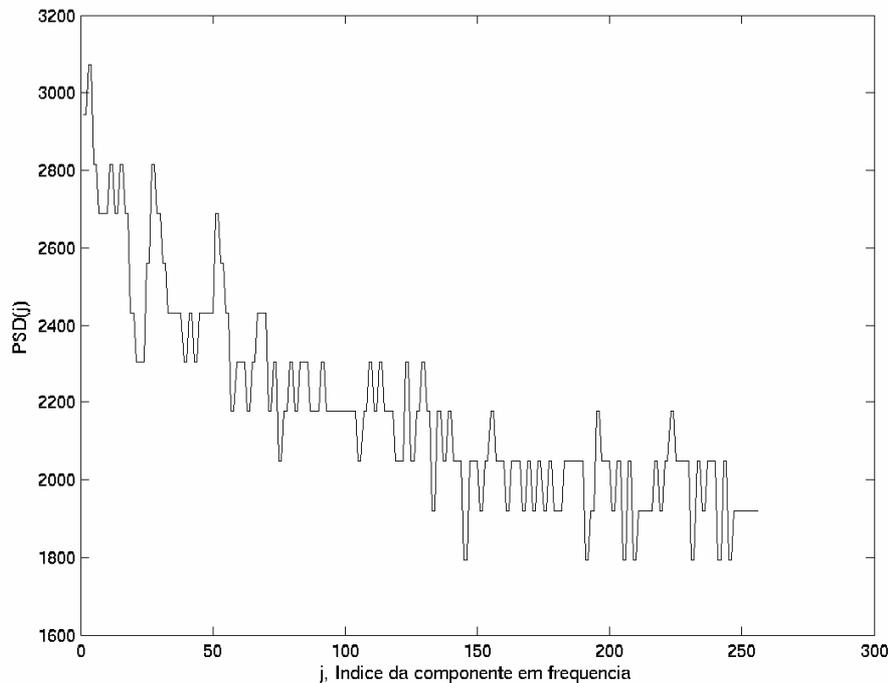


Figura E.6 – Exemplo do cálculo da DEP

Nessa escala logarítmica cada 128 unidades correspondem a 6 dB. A DEP é uma referência do nível de intensidade que as componentes do espectro de um determinado bloco de áudio apresentam.

#### E.4.4 - Integração em Bandas Críticas

O codificador toma o espectro transformado e o divide em 50 bandas de largura variável, denominadas bandas críticas, sendo que as bandas correspondentes às baixas frequências são mais estreitas do que aquelas que correspondem às altas frequências. Esse formato de divisão do espectro aproxima a forma como o ouvido humano percebe o sinal de áudio. Isto é, as componentes em frequência agrupadas em uma mesma banda críticas, são percebidas aproximadamente com a mesma intensidade e, portanto, podemos assumir que representam a mesma quantidade de informação e devem ser representadas com a mesma quantidade de bits. A tabela 3.9 no final do capítulo mostra a divisão do espectro nas bandas críticas do sistema AC-3.

Essa forma de divisão do espectro auditivo implica que a DEP deve ser integrada ou mapeada logaritmicamente nas bandas críticas. Assim, a integração em bandas críticas é realizada usando uma operação chamada de log-add, dada pela seguinte equação:

$$\log(a + b) = \max[\log(a), \log(b)] + \log(1 + e^d) \quad \text{onde} \quad (E.4)$$

$$d = |\log(a) - \log(b)|$$

A Figura E.7 mostra o resultado da integração em bandas críticas da DEP do exemplo da Figura E.6.

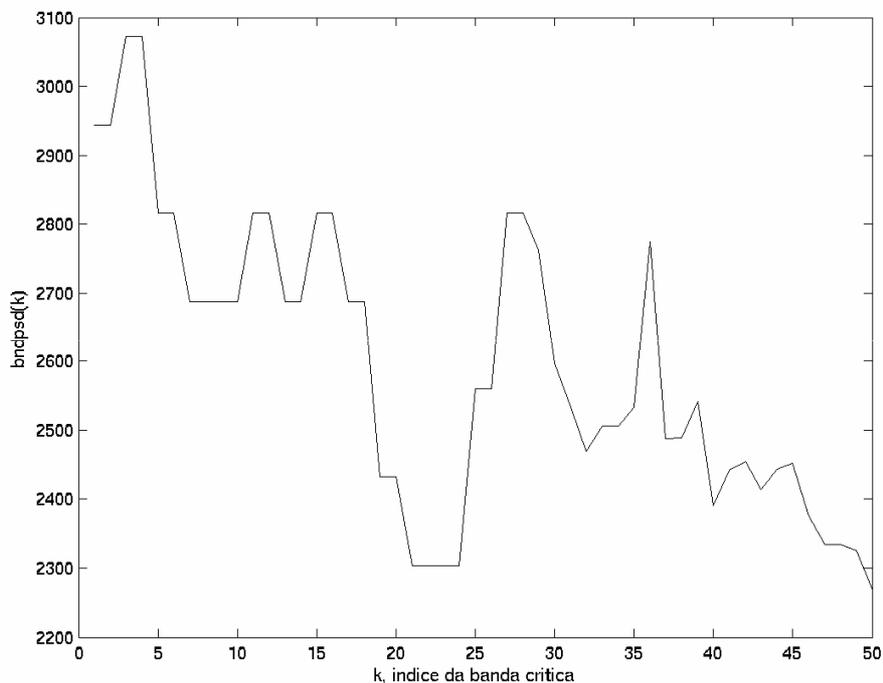


Figura E.7 – Resultado da integração em bandas críticas.

#### E.4.5 - Cálculo da função de espalhamento

A função de espalhamento especifica o limiar de ruído permitido para cada banda crítica k usando um modelo paramétrico que é ajustado pela DEP integrada em cada bandas crítica, bndpsd(k). A função de espalhamento é gerada a partir da exploração do fenômeno de mascaramento auditivo de componentes de baixa potência, próximas em frequência, por componentes de alta potência. O AC-3 faz a aproximação dessa função por dois segmentos lineares especificados por suas inclinações e ganhos. A Figura E.8 ilustra o modelo da função de espalhamento usado no AC-3.

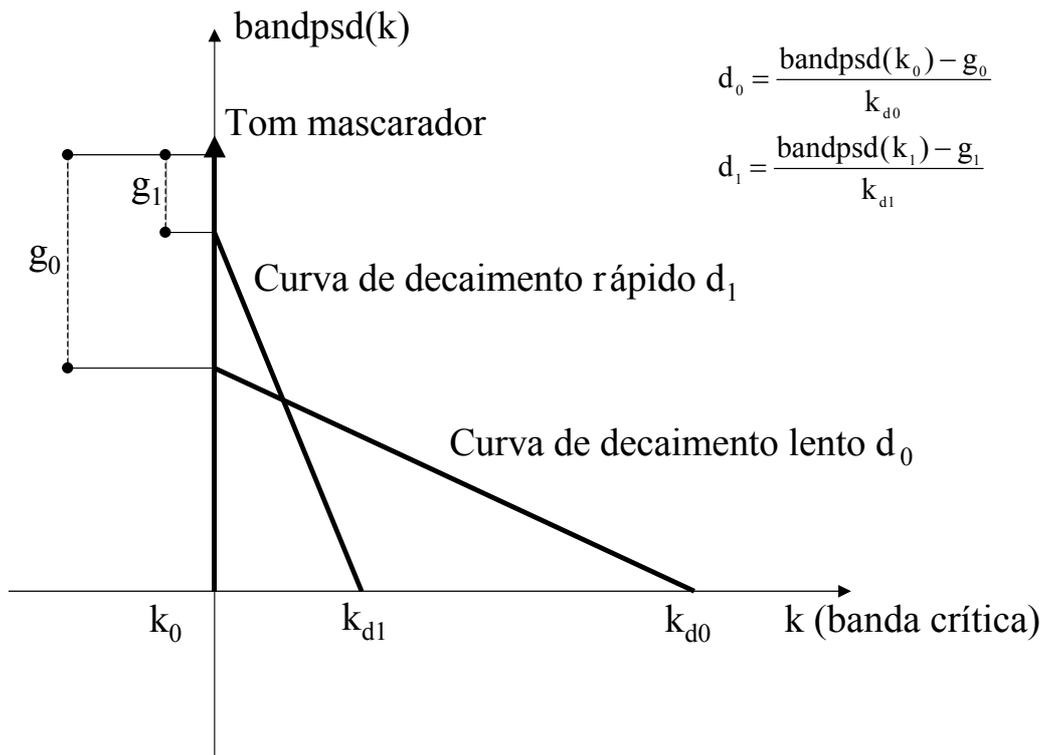


Figura E.8 – Modelo da função de espalhamento.

Usando as especificações de inclinações e os ganhos da função de mascaramento, o modelo psico-acústico realiza uma convolução entre a PSD integrada em bandas críticas,  $\text{bandpsd}(k)$ , e a função de espalhamento. O resultado desta convolução é chamado de função de excitação e determina um limiar superior para o ruído de quantização das componentes em frequência. A equação (E.5) descreve esse processo:

$$\begin{aligned}
 x_0 &= (x_0 - d_0) \oplus (\text{bandpsd}(k) - g_0) \\
 x_1 &= (x_1 - d_1) \oplus (\text{bandpsd}(k) - g_1) \\
 E(k) &= \max(x_0, x_1)
 \end{aligned}
 \tag{E.5}$$

onde  $d_0$  e  $d_1$  são as inclinações da função de espalhamento,  $g_0$  e  $g_1$  são os ganhos,  $\text{bandpsd}(k)$  é a DEP integrada em bandas críticas,  $E(k)$  a função de espalhamento e  $\oplus$  é a operação de log-add. Os possíveis valores de  $d_0, d_1, g_0$  e  $g_1$  especificados pelo AC-3 são dados na tabela a seguir.

Tabela E.4 – Parâmetros da função de espalhamento.

$d_0$ (decaimento lento)	$d_1$ (decaimento rápido)	$g_0$ (ganho do segmento de decaimento lento)	$g_1$ (ganho do segmento de decaimento rápido)

15	63	1344	128
17	83	1240	256
19	103	1144	384
21	123	1040	512
-	-	-	640
-	-	-	768
-	-	-	896
-	-	-	1024

Embora os valores da tabela acima estejam representados lado a lado, é importante ressaltar que qualquer combinação desses valores pode ser usada pelo sistema. A figura a seguir mostra o resultado do cálculo da função de excitação a partir da  $\text{bandpsd}(k)$  da Figura E.9.

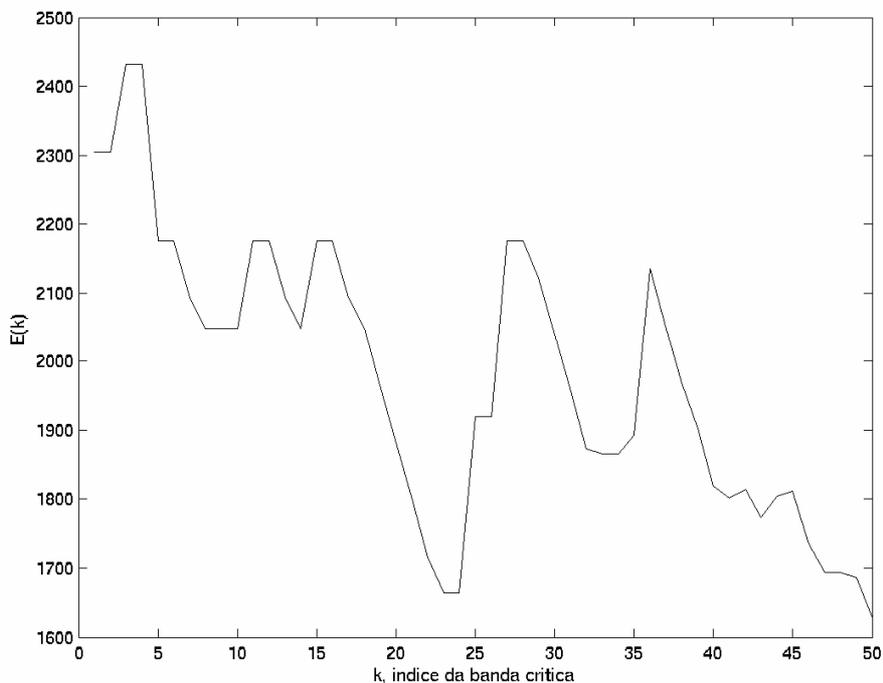


Figura E.9 – Função de excitação.

#### E.4.6 - Cálculo da Função de Mascaramento Global

A função de excitação determina os níveis de ruído de quantização permitidos para o segmento de áudio que está sendo processado, mas o codificador e o decodificador também devem levar em consideração o limiar absoluto de audição do ouvido humano para calcular um limiar global de ruído. O limiar absoluto de audição especifica as componentes de baixa potência que não serão percebidas pelo ouvido humano em uma determinada faixa de frequência, mesmo sem a ocorrência de um tom mascador em frequências próximas.

A função de excitação é comparada com o limiar absoluto de audição humana (Tabela E.13 no final deste apêndice) e o máximo entre os dois é tomado para formar a função de mascaramento global. A Figura E.10 apresenta os limiares de audição absoluta definidos no sistema AC-3.

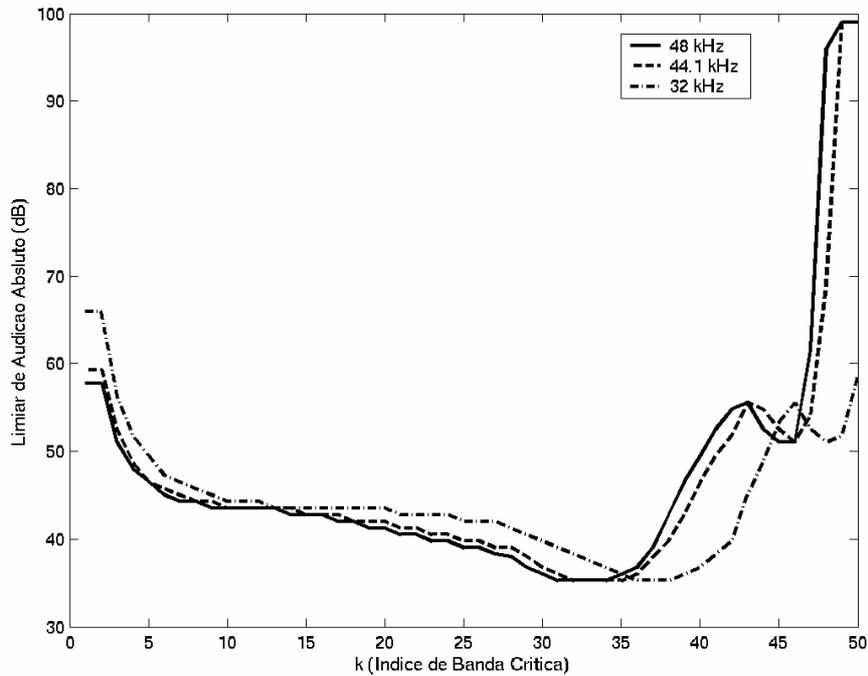


Figura E.10 – Limiar de audição absoluto.

O limiar absoluto de audição é dependente da taxa de amostragem do sinal de áudio e por isso temos uma curva para cada frequência de amostragem disponível. A figura a seguir mostra o resultado do cálculo da função de mascaramento global a partir da  $E(k)$  da Figura E.9.

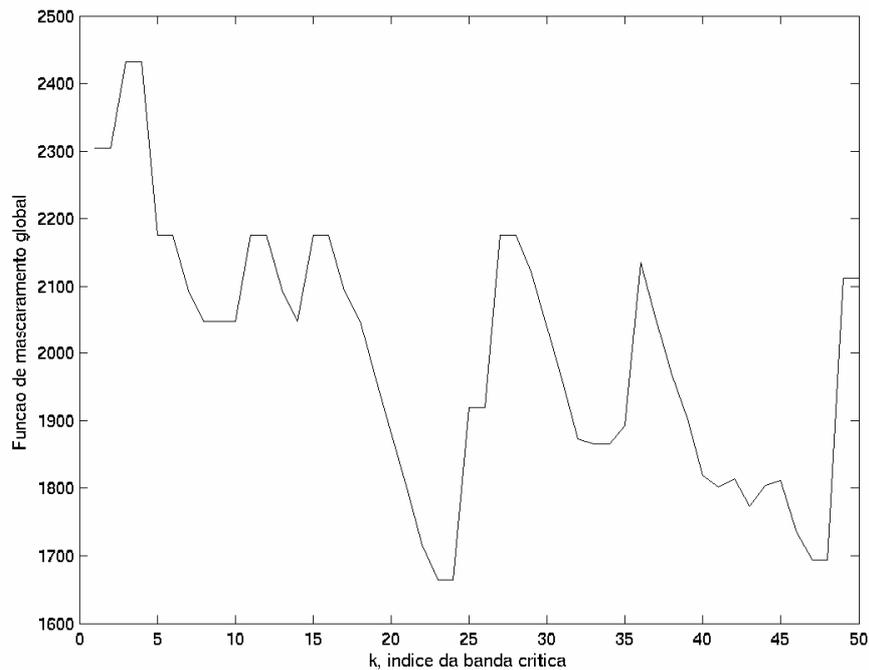


Figura E.11 – Função de mascaramento global.

#### E.4.7 - Cálculo da Alocação de Bits

Nessa etapa final o codificador calcula o tipo de quantizador que será usado para as mantissas de cada componente em frequência. Primeiro são comparados os valores da DEP com a curva de mascaramento global, subtraindo a curva de mascaramento da DEP e deslocando o resultado à direita em 5 bits. Esse procedimento equivale ao cálculo de uma razão sinal ruído, SNR (do inglês, *Signal to Noise Ratio*). A SNR é usada para endereçar uma tabela que especifica a quantidade de bits ou número de níveis de quantização para a mantissa de cada componente em frequência. Quanto maior a SNR de um determinado coeficiente, maior também será o número de níveis de quantização designado para a mantissa correspondente. Se considerarmos um quantizador uniforme, cada bit adicional alocado para um componente espectral corresponderia a um ganho de 6 dB na SNR. Esse critério é impreciso para um número pequeno de bits. Por isso, o número de níveis de quantização é encontrado pelo sistema AC-3 através de um ponteiro de alocação de bits,  $pab(k)$ , que considera um ganho menor do que 6 dB por bit e varia de acordo com a SNR segundo a Tabela E.14 no final do capítulo.

A soma do número de bits alocados para cada mantissa é comparada com a quantidade de bits disponíveis para cada quadro. Deste ponto em diante o sistema começa uma rotina de ajuste da alocação de bits.

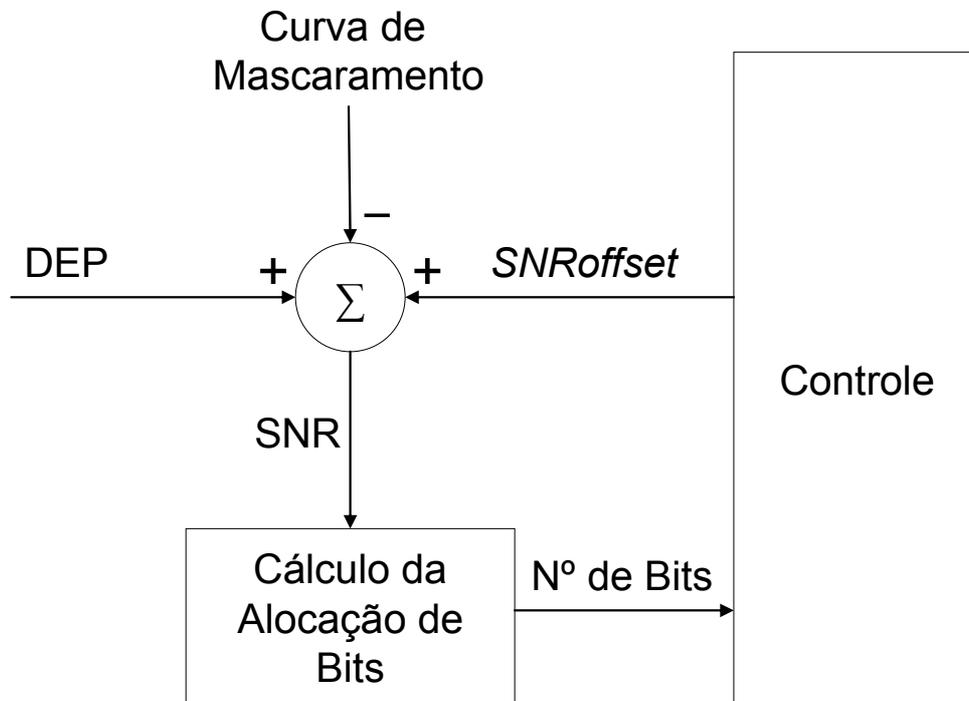


Figura E.12 – Ajuste da alocação de bits.

No ajuste da alocação de bits a rotina de controle identifica se a quantidade de bits calculada foi menor do que a quantidade de bits disponíveis, caso isso aconteça, um parâmetro chamado *SNRoffset* é subtraído da SNR e recomeça uma nova alocação de bits. A forma como este parâmetro é adaptado não é completamente especificada e uma busca exaustiva demanda uma grande quantidade de processamento. A sugestão do documento A/52 é que esse parâmetro seja ajustado em incrementos de 3 dB, começando de um valor inicial padrão. Quando a quantidade de bits calculada excede a quantidade de bits disponíveis, o valor de *SNRoffset* imediatamente anterior é tomado e inicia-se um ajuste fino. No ajuste fino, o valor de *SNRoffset* é incrementado de 3/8 dB e o mesmo processo anterior se repete até que a quantidade bits disponíveis seja excedida novamente.

Se a quantidade de bits calculada originalmente for superior à quantidade de bits disponíveis, uma espécie de exceção é sinalizada e a rotina de controle pode tentar ajustar os parâmetros

da função de espalhamento (inclinações e ganhos dos segmentos lineares) para recomençar o algoritmo de alocação de bits do início. A figura a seguir apresenta um exemplo do resultado da alocação de bits do codec AC-3.

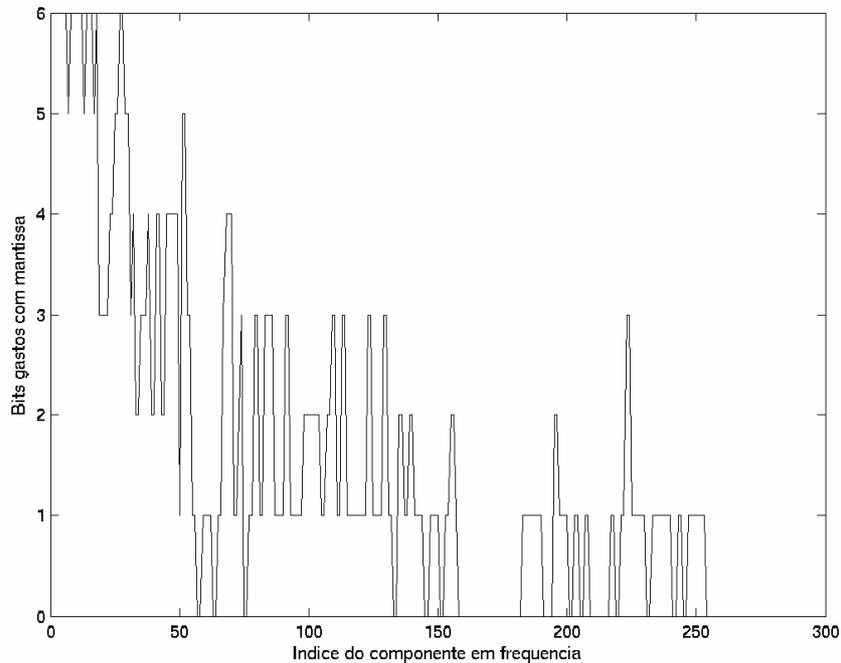


Figura E.13 – Resultado da alocação de bits.

## E.5 - QUANTIZAÇÃO DAS MANTISSAS

Depois de calcular a alocação de bits, o codificador realiza a quantização das mantissas para cada componente em frequência. O número de níveis de quantização pode variar de 0 até 65535, o que corresponde a uma precisão de 16 bits na representação do valor de cada mantissa. A quantização pode ser feita de duas formas distintas, quantização simétrica ou assimétrica.

A quantização assimétrica é feita quando os valores do ponteiro de alocação de bits estão no intervalo de 6 a 15, e denomina-se assimétrica porque o número de níveis de quantização para valores positivos e negativos de cada mantissa não é o mesmo. Assim, os valores de cada mantissa variam de  $(1.0 - 2^{\text{qntztzb}[\text{pab}(k)]})$  a  $-1.0$ , sendo os valores negativos quantizados com um nível a mais do que os valores positivos. Onde, qntztzb é uma tabela do número de níveis de quantização indexada pelo ponteiro de alocação de bits, pab(k), dada a seguir.

Tabela E.5 – Quantizador assimétrico.

Ponteiro de alocação de bits pab(k)	Nº de níveis de quantização (qntztzb)	Nº de bits gasto por mantissa
0	0	0
1*	3	1.67 (5 bits / 3 mantissas)
2*	5	2.33 (7 bits / 3 mantissas)
3	7	3
4*	11	3.5 (7 bits / 2 mantissas)
5	15	4
6	32	5
7	64	6
8	128	7
9	256	8
10	512	9
11	1024	10
12	2048	11
13	4096	12
14	16384	14
15	65536	16

\* Para estes valores de pab(k), mais de uma mantissa é codificada em uma mesma palavra.

A quantização simétrica é considerada quando os valores do ponteiro pab(k) estão no intervalo de 1 a 5 e oferece o mesmo número de níveis de quantização para os valores positivos e negativos das mantissas. As tabelas a seguir especificam os quantizadores simétricos usados no sistema AC-3.

Tabela E.6 – Quantizador de 3 níveis (pab(k)=1).

Código para a mantissa	Valor da mantissa
0	-2/3
1	0
2	2/3

Tabela E.7 – Quantizador de 5 níveis (pab(k)=2).

Código para a mantissa	Valor da mantissa
0	-4/5
1	-2/5
2	0
3	2/5
4	4/5

Tabela E.8 – Quantizador de 7 níveis (pab(k)=3).

Código para a mantissa	Valor da mantissa
------------------------	-------------------

0	-6/7
1	-4/7
2	-2/7
3	0
4	2/7
5	4/7
6	6/7

Tabela E.9 – Quantizador de 11 níveis ( $pab(k)=4$ ).

Código para a mantissa	Valor da mantissa
0	-10/11
1	-8/11
2	-6/11
3	-4/11
4	-2/11
5	0
6	2/11
7	4/11
8	6/11
9	8/11
10	10/11

Tabela E.10 – Quantizador de 15 níveis ( $pab(k)=5$ )

Código para a mantissa	Valor da mantissa
0	-14/15
1	-12/15
2	-10/15
3	-8/15
4	-6/15
5	-4/15
6	-2/15
7	0
8	2/15
9	4/15
10	6/15
11	8/15
12	10/15
13	12/15
14	14/15

Quando a alocação de bits especifica que uma determinada mantissa deve ser codificada com 0 bit, o codificador não transmite o seu valor e o decodificador a substitui por uma variável aleatória uniforme com valores entre 0.707 e -0.707. Esse procedimento preserva a informação contida no expoente, o que não aconteceria se substituíssemos a mantissa por 0.

## **E.6 - FUNCIONALIDADES EXTRAS DO SISTEMA AC-3**

Até agora discutimos a implementação básica do sistema AC-3, mas a sua estrutura híbrida permite a utilização de uma série de outras funcionalidades que não foram implementadas. Na implementação básica do AC-3, só é necessário transmitir informação lateral para os parâmetros do modelo psico-acústico ( $SNR_{offset}$ ,  $d_0, d_1$ ,  $g_0$  e  $g_1$ ). Entretanto algumas aplicações exigem qualidades de reconstrução do sinal de áudio e taxas de transmissão de bits que não podem ser atendidas pelo codificador básico. Então, algumas vezes ao custo da transmissão de um overhead de informação, o sistema pode habilitar as funcionalidades extras discutidas resumidamente a seguir.

### **E.6.1 - Alocação delta de bits**

A alocação delta de bits é uma maneira de aumentar a qualidade de blocos de áudio, atendendo a requisitos de qualidade de algumas aplicações, sem que o sistema tenha que modificar os parâmetros padrões do modelo psico-acústico. Assim, uma informação lateral adicional de alocação de bits é transmitida pelo codificador ao decodificador, informando um aumento ou diminuição na alocação de bits calculada pelo modelo psico-acústico padrão. A informação lateral transmitida é formada por um vetor de incrementos múltiplos de  $\pm 6$  dB na curva de mascaramento global para cada banda crítica. Um incremento de  $-6$  dB corresponde a um aumento de precisão de 1 bit nas mantissas que estão dentro da mesma banda crítica. Embora esta estratégia sacrifique o ganho de compressão obtido pelo codificador, uma vez que há um aumento na informação lateral a ser transmitida, a qualidade do sinal de áudio pode ser consideravelmente melhorada.

### **E.6.2 - Acoplamento de Canais de Áudio**

No acoplamento de canais de áudio o sistema tenta diminuir a redundância entre os diversos canais de áudio que são transmitidos. O ouvido humano não distingue, com precisão, componentes de alta frequência de vários canais de áudio reproduzidos ao mesmo tempo. Por isso, quando o acoplamento de canais é habilitado, o sistema transmite os canais individualmente somente até um determinado coeficiente de baixa frequência. Todos os outros coeficientes são transmitidos uma única vez, junto com um fator de escala para reconstrução de cada canal. Para cada um dos 5 canais de áudio, o sistema agrupa os coeficientes de número 37 a 252 em 18 sub-bandas de 12 coeficientes. A média desses 12 coeficientes em cada sub-banda é tomada e um fator de escala é calculado para cada canal.

Portanto, no lugar de transmitir os coeficientes de número 37 a 252, para cada canal, o codificador transmite somente 18 valores médios para todos os canais e um fator de escala para cada canal. A Tabela E.11 ilustra como a divisão das 18 sub-bandas é feita para cada taxa de amostragem disponível.

Tabela E.11 – Divisão das sub-bandas no acoplamento de canais

Nº da sub-banda	Primeiro coeficiente	Último coeficiente	Freq. do primeiro coeficiente (kHz) fs=48 kHz	Freq. Do último coeficiente (kHz) fs=48 kHz	Freq. do primeiro coeficiente (kHz) fs=44.1 kHz	Freq. Do último coeficiente (kHz) fs=44.1 kHz
0	37	48	3.42	4.55	3.14	4.18
1	49	60	4.55	5.67	4.18	5.21
2	61	72	5.67	6.8	5.21	6.24
3	73	84	6.8	7.92	6.24	7.28
4	85	96	7.92	9.05	7.28	8.31
5	97	108	9.05	10.17	8.31	9.35
6	109	120	10.17	11.3	9.35	10.38
7	121	132	11.3	12.42	10.38	11.41
8	133	144	12.42	13.55	11.41	12.45
9	145	156	13.55	14.67	12.45	13.48
10	157	168	14.67	15.8	13.48	14.51
11	169	180	15.8	16.92	14.51	15.55
12	181	192	16.92	18.05	15.55	16.58
13	193	204	18.05	19.17	16.58	17.61
14	205	216	19.17	20.3	21.42	18.65
15	217	228	20.3	21.42	18.65	19.68
16	229	240	21.42	22.55	19.68	20.71
17	241	252	22.55	23.67	20.71	21.75

\* Obs: A 32 kHz as frequências das sub-bandas são iguais a 2/3 do valor para 48 kHz.

### E.6.3 - Rematriciamento de canais de áudio

O rematriciamiento é uma técnica na qual a soma de diferentes canais de áudio, altamente correlacionados, é codificada no lugar dos canais individualmente. No lugar de transmitir os canais, esquerdo e direito de um áudio estéreo, o sistema codifica:

$$\text{esquerdo}^* = 0.5(\text{esquerdo} + \text{direito}); \quad (\text{E.6})$$

$$\text{direito}^* = 0.5(\text{esquerdo} - \text{direito});$$

A operação de quantização e empacotamento de dados é feita sobre os canais: esquerdo\* e direito\*. Se o sinal estéreo original é composto por dois canais mono iguais, esta técnica resultará em um canal esquerdo\* igual aos dois canais mono e um canal direito\* igual a zero. Como resultado disso, o canal direito\* pode ser codificado com poucos bits e a precisão do canal esquerdo\* pode ser aumentada. O rematriciamiento é feito para os coeficientes de 0 a 36 que não foram incluídos no acoplamento de canais. Maiores detalhes sobre a alocação delta de bits, o acoplamento de canais e o rematriciamiento de canais podem ser encontrados no documento (A/52,1995).

Tabela E.12 – Divisão do espectro em bandas críticas.

Banda crítica (k)	Índice da componente de frequência	Quantidade de coeficientes de frequência por banda crítica
0	0	1
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1
11	11	1
12	12	1
13	13	1
14	14	1
15	15	1
16	16	1
17	17	1
18	18	1
19	19	1
20	20	1
21	21	1
22	22	1
23	23	1
24	24	1
25	25	1
26	26	1
27	27	1
28	28	3
29	31	3
30	34	3
31	37	3
32	40	3
33	43	3
34	46	3
35	49	6
36	55	6
37	61	6
38	67	6
39	73	6
40	79	6
41	85	12
42	97	12
43	109	12

44	121	12
45	133	24
46	181	24
47	181	24
48	205	24
49	229	24

Tabela E.13 – Limiar absoluto de audição

Nº da banda crítica	Valores da função de espalhamento (dB)		
	48 kHz	44.1 kHz	32kHz
0	57.7500	59.2500	66.0000
1	57.7500	59.2500	66.0000
2	51.0000	52.5000	56.2500
3	48.0000	48.7500	51.7500
4	46.5000	46.5000	49.5000
5	45.0000	45.7500	47.2500
6	44.2500	45.0000	46.5000
7	44.2500	44.2500	45.7500
8	43.5000	44.2500	45.0000
9	43.5000	43.5000	44.2500
10	43.5000	43.5000	44.2500
11	43.5000	43.5000	44.2500
12	43.5000	43.5000	43.5000
13	42.7500	43.5000	43.5000
14	42.7500	42.7500	43.5000
15	42.7500	42.7500	43.5000
16	42.0000	42.7500	43.5000
17	42.0000	42.0000	43.5000
18	41.2500	42.0000	43.5000
19	41.2500	42.0000	43.5000
20	40.5000	41.2500	42.7500
21	40.5000	41.2500	42.7500
22	39.7500	40.5000	42.7500
23	39.7500	40.5000	42.7500
24	39.0000	39.7500	42.0000
25	39.0000	39.7500	42.0000
26	38.2500	39.0000	42.0000
27	37.9688	39.0000	41.2500
28	36.7500	37.9688	40.5000
29	36.0000	36.7500	39.7500
30	35.2500	36.0000	39.0000
31	35.2500	35.2500	38.2500
32	35.2500	35.2500	37.5000
33	35.2500	35.2500	36.7500
34	36.0000	35.2500	36.0000
35	36.7500	36.0000	35.2500
36	39.0000	37.9688	35.2500

37	42.7500	39.7500	35.2500
38	46.5000	42.7500	36.0000
39	49.5000	46.5000	36.7500
40	52.5000	49.5000	38.2500
41	54.7500	51.7500	39.7500
42	55.5000	55.5000	45.0000
43	52.5000	54.7500	48.7500
44	51.0000	52.5000	53.2500
45	51.0000	51.0000	55.5000
46	61.5000	54.0000	52.5000
47	96.0000	69.0000	51.0000
48	99.0000	99.0000	51.7500
49	99.0000	99.0000	58.5000

Tabela E.14 – Ponteiro de Alocação de bits versus SNR

SNR	pab(k)
0	0
1	1
2	1
3	1
4	1
5	2
6	2
7	3
8	3
9	3
10	4
11	4
12	5
13	5
14	6
15	6
16	6
17	6
18	7
19	7
20	7
21	7
22	8
23	8
24	8
25	8
26	9
27	9
28	9
29	9
30	10

31	10
32	10
33	10
34	11
35	11
36	11
37	11
38	12
39	12
40	12
41	12
42	13
43	13
44	13
45	13
46	14
47	14
48	14
49	14
50	14
51	14
52	14
53	14
54	15
55	15
56	15
57	15
58	15
59	15
60	15
61	15
62	15
63	15