



Universidade Federal do Amazonas  
Faculdade de Tecnologia  
Programa de Pós-Graduação em Engenharia Elétrica

# Compressão de Sinais de Eletrocardiograma Utilizando Técnicas de Codificação Distribuída

Patrese Emiron Barbosa de Souza

Manaus – Amazonas  
Setembro de 2019

Patrese Emiron Barbosa de Souza

# Compressão de Sinais de Eletrocardiograma Utilizando Técnicas de Codificação Distribuída

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica. Área de concentração: Processamento Digital de Sinais.

Orientador: Eddie Batista de Lima Filho

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S729c Souza, Patrese Emiron Barbosa de  
Compressão de Sinais de Eletrocardiograma Utilizando Técnicas de Codificação Distribuída / Patrese Emiron Barbosa de Souza.  
2019  
140 f.: il. color; 31 cm.

Orientador: Eddie Batista de Lima Filho  
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Compressão de ECG. 2. Codificação distribuída. 3. Códigos turbo. 4. Teoria da informação. I. Lima Filho, Eddie Batista de II. Universidade Federal do Amazonas III. Título

PATRESE EMIRON BARBOSA DE SOUZA

**COMPRESSÃO DE SINAIS DE ELETROCARDIOGRAMA UTILIZANDO  
TÉCNICAS DE CODIFICAÇÃO DISTRIBUIDA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 02 de setembro de 2019.

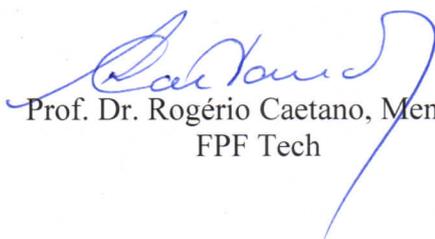
BANCA EXAMINADORA



Prof. Dr. Eddie Batista de Lima Filho, Presidente  
R&D/TPV do Brasil



Prof. Dr. Waldir Sabino da Silva Júnior, Membro  
Universidade Federal do Amazonas



Prof. Dr. Rogério Caetano, Membro  
FPF Tech

Patrese Emiron Barbosa de Souza

# Compressão de Sinais de Eletrocardiograma Utilizando Técnicas de Codificação Distribuída

Banca Examinadora

Prof. D.Sc. Eddie Batista de Lima Filho – Presidente e Orientador

TPV Technology

Prof. D.Sc. Waldir Sabino da Silva Júnior

Universidade Federal do Amazonas - UFAM

Prof. D.Sc. Rogério Caetano

FPF Tech

Manaus – Amazonas

Setembro de 2019

# Agradecimentos

À Deus pelo dom da vida.

À minha família, pelo amor, incentivo e apoio incondicional.

Ao meu orientador Eddie Batista de Lima Filho, pela amizade, comprometimento, paciência, empenho e sentido prático ao longo da jornada deste trabalho, assim como, durante a jornada do curso. Obrigado pelas correções e conselhos sempre práticos e objetivos.

Agradeço a todos que participaram dessa jornada de forma direta e indireta, em especial minha mãe Edina Maria e minha amada Larissa Melgueiro que sempre prestaram apoio incondicional.

*“Dedico este trabalho à minha mãe por todo apoio incondicional, pois sempre com muito carinho e com toda dificuldade não mediu esforços para que eu chegasse até aqui”*

# Resumo

Sinais de eletrocardiograma (ECG) podem ser utilizados tanto para acompanhamento quanto diagnóstico de doenças cardíacas, o que depende de infraestruturas de armazenamento e transmissão. Entretanto, como memória e banda de transmissão são recursos finitos, torna-se necessário comprimir dados de ECG. Além disso, com relação ao monitoramento à distância, espera-se que o hardware utilizado seja simples e barato. O presente trabalho ataca os problemas mencionados e propõe uma metodologia de compressão de sinais de ECG, onde experimentos com uma implementação desse paradigma mostraram que tal abordagem é possível e também eficaz, desde que alinhada às condições de cada caso específico. Este esquema é baseado em técnicas de codificação distribuída de sinais, cuja principal característica consiste em realizar a maior parte da exploração estatística desse tipo de processo em decodificadores e assim permitir um processo de codificação mais simples, o que tem o potencial de facilitar a sua implementação em dispositivos embarcados, como o *Holter* de ECG.

Palavras-chave: compressão de ECG, codificação distribuída, códigos turbo, teoria da informação.

# Abstract

Electrocardiographic (ECG) signals can be used for both disease monitoring and diagnosis, which is also dependent on storage and transmission infrastructures. Nonetheless, as memory and transmission bandwidth are finite resources, it is necessary to compress ECG data. Additionally, with respect to remote monitoring, it is interesting that the target hardware be simple and affordable. The present work tackles the mentioned problems and proposes a ECG compression methodology, where experiments with a real implementation showed that such an approach is feasible and also effective, as long as aligned with an application's specific conditions. This scheme is based on distributed source coding techniques, whose main feature is to perform most of the associated statistical exploration in decoders, while allowing a simple encoding process, which has the potential to favor its implementation in embedded systems, such as ECG holter devices.

Keywords: ECG compression, distributed source coding, turbo codes, information theory.

# Índice

<b>Índice de Figuras</b>	<b>xi</b>
<b>Índice de Tabelas</b>	<b>xv</b>
<b>Abreviações</b>	<b>xvi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Caracterização do Problema . . . . .	2
1.2 Objetivo . . . . .	3
1.2.1 Objetivos Específicos . . . . .	4
1.3 Contribuições . . . . .	4
1.4 Organização da Dissertação . . . . .	5
<b>2 Trabalhos Relacionados</b>	<b>6</b>
<b>3 O sinal de ECG</b>	<b>11</b>
3.1 A origem do sinal de ECG . . . . .	11
3.2 Morfologia do sinal de ECG . . . . .	13
3.3 Derivações . . . . .	15
3.4 O dispositivo <i>Holter</i> . . . . .	18
<b>4 Compressão de sinais de ECG</b>	<b>19</b>
4.1 Compressão de Dados . . . . .	19
4.2 Compressão direta de ECG . . . . .	21
4.3 Compressão Paramétrica de ECG . . . . .	21
4.4 Compressão de ECG utilizando transformadas . . . . .	23
4.4.1 A transformada discreta dos cossenos na compressão de sinais de ECG .	23

---

4.5	Uma nova classe de compressão de ECG baseada na compressão distribuída . . .	24
<b>5</b>	<b>Codificação distribuída de sinais</b>	<b>27</b>
5.1	A codificação distribuída . . . . .	27
5.2	A codificação distribuída sem perdas . . . . .	27
5.3	A codificação distribuída com perdas . . . . .	32
5.4	Códigos turbo . . . . .	33
5.4.1	Codificação Turbo . . . . .	34
5.4.2	Decodificação Turbo . . . . .	35
5.4.3	O algoritmo MAP e a decodificação iterativa . . . . .	38
<b>6</b>	<b>A compressão distribuída de ECG</b>	<b>41</b>
6.1	Introdução . . . . .	41
6.2	Arquitetura de compressão distribuída de ECG . . . . .	42
6.3	Pré-processamento . . . . .	44
6.4	CODEC Wyner-Ziv no domínio da amostra . . . . .	50
6.4.1	O codificador Slepian-Wolf . . . . .	52
6.4.2	O entrelaçador . . . . .	53
6.4.3	A codificação . . . . .	54
6.4.4	Etapa de perfuração . . . . .	56
6.4.5	Interpolação de KSs . . . . .	57
6.4.6	A decodificação . . . . .	57
6.4.7	Reconstrução . . . . .	62
6.5	CODEC Wyner-Ziv no domínio da transformada . . . . .	63
6.6	Resultados . . . . .	68
6.6.1	Base de Dados . . . . .	69
6.6.2	Métricas de desempenho . . . . .	69
6.6.3	Resultados experimentais . . . . .	70
6.6.4	Discussão e análise de resultados . . . . .	82
<b>7</b>	<b>Utilizando resíduo de ECG</b>	<b>85</b>
7.1	Metodologia . . . . .	86
7.2	Resultados . . . . .	88

# Índice de Figuras

3.1	Ilustração da propagação elétrica em uma célula cardíaca . . . . .	12
3.2	Sistema de condução do coração e os potenciais de ação típicos de cada parte do coração e a sua correlação com a atividade elétrica adquirida na pele (ECG). . . . .	13
3.3	Período completo de um sinal de ECG. . . . .	14
3.4	Sistema de derivações bipolares. Representação visual das formas de onda. . . . .	15
3.5	Sistema de derivações unipolares. Exemplo da obtenção da derivação aVR. . . . .	16
3.6	Sistema de derivações pré-cordiais: (1) eletrodos posicionados no peito e terminal central conectando os membros. . . . .	16
3.7	Formas de onda de 12 derivações de ECG. . . . .	17
3.8	Ondas de ECG das derivações MLII e V1. . . . .	17
4.1	Esquema geral um sistema de compressão de dados. . . . .	20
4.2	Comparação entre os métodos: (a) TP, (b) AZTEC e (c) CORTES . . . . .	22
4.3	Arquitetura da metodologia de compressão JPEG, (a) codificador, (d) decodificador	25
5.1	Ilustração de esquema de codificação tradicional. . . . .	29
5.2	Codificação distribuída de duas fontes ( <i>i.i.d</i> ) $X$ e $Y$ . . . . .	30
5.3	Região de taxas atingíveis da codificação de Slepian Wolf. . . . .	31
5.4	Exemplo de codificação distribuída sem perdas. . . . .	31
5.5	Esquema de codificação Wyner-Ziv. . . . .	32
5.6	CODEC Wyner-Ziv. . . . .	33
5.7	Arquitetura básica de codificação Wyner-Ziv . . . . .	34
5.8	Codificador Turbo de taxa 1/3. . . . .	35
5.9	Decodificação turbo. . . . .	36
5.10	Exemplo de uma treliça de codificação turbo. . . . .	37

---

7.2.1	Discussão e análise de resultados . . . . .	97
<b>8</b>	<b>Retirando o canal de retorno</b>	<b>98</b>
8.1	Metodologia . . . . .	99
8.2	Resultados . . . . .	101
8.2.1	Discussão e análise de resultados . . . . .	107
<b>9</b>	<b>Conclusões</b>	<b>109</b>
	<b>Referências Bibliográficas</b>	<b>112</b>
<b>A</b>	<b>Artigo desenvolvido nesta dissertação</b>	<b>121</b>

6.1	Arquitetura da compressão distribuída de ECG. . . . .	43
6.2	Etapa de pré-processamento executada em codificadores. . . . .	45
6.3	Detecção de períodos de ECG. . . . .	45
6.4	Períodos de ECG normalizados. . . . .	46
6.5	Etapa de pré-processamento. (a) Períodos originais, (b) Períodos normalizados, (c) Sinal com equalização DC, (d) Sinal com ordenação por similaridade. . . . .	47
6.6	Registro 117 do MIT-BIH [23] após etapa de pré-processamento. (a) Períodos originais, (b) Períodos normalizados, (c) Sinal com equalização DC, (d) Sinal com ordenação por similaridade. . . . .	48
6.7	Registro 119 do MIT-BIH [23] após etapa de pré-processamento. (a) Períodos originais, (b) Períodos normalizados, (c) Sinal com equalização DC, (d) Sinal com ordenação por similaridade. . . . .	49
6.8	Esquema da predição de WZSs utilizando método de interpolação de KSs. . . . .	50
6.9	Esquema geral do CODEC Wyner-Ziv no domínio da amostra. . . . .	51
6.10	Visão geral do codificador Wyner-Ziv. . . . .	53
6.11	Codificador recursivo sistemático convolucional com 4 memórias, 16 estados e a sua matriz geradora representada pela equação (6.5). . . . .	54
6.12	(a) Representação da transição de estados da função geradora determinada pela equação (6.5). A Figura (b) mostra a saída do codificador, estado atual e transição para o estado futuro em resposta ao <i>bit</i> 0. A Figura (b) mostra a saída do codificador, estado atual e transição para o estado futuro em resposta ao <i>bit</i> 1. . . . .	55
6.13	Esquema de perfuração utilizando blocos com largura de 4 <i>bits</i> . . . . .	57
6.14	Esquema de geral de interligação entre codificador e decodificador. . . . .	58
6.15	Distribuição residual do registro 100 da base de dados do MIT-BIH [23]. . . . .	59
6.16	Reversão da etapa de pré-processamento. . . . .	63
6.17	Sinal de ECG reconstruído com CDA: (a) sinal original; (b) sinal reconstruído; (c) erro de reconstrução. . . . .	64
6.18	Esquema geral do CODEC Wyner-Ziv no domínio da transformada. . . . .	65
6.19	Matrizes de quantização. . . . .	67
6.20	Bandas de coeficientes geradas após etapa da transformada. . . . .	67
6.21	Resultados da transformada DCT em 4 sub-blocos de uma matriz $X_n$ de um registro de ECG utilizado nesta dissertação. . . . .	68

6.22	Registros após etapa de pré-processamento. (a) registro 107; (b) registro 117. . . . .	72
6.23	Espectro de frequência do registro 100 da base de dados do MIT-BIH. . . . .	74
6.24	Curvas de desempenho; (a) CDA GoS-3; (b) CDA GoS-5; (c) Média das curvas de CDA GoS-3 e GoS-5. . . . .	75
6.25	Curvas de desempenho; (a) CDT -GoS-3; (b) CDT GoS-5; (c) Média das curvas de CDT GoS-3 e GoS-5. . . . .	76
6.26	Desempenho de compressão de CDA e CDT. . . . .	77
6.27	Registro 100 reconstruído com CDA GoS-3 com $CR = 11,81$ e $PRD = 3,81$ . . . . .	78
6.28	Registro 107 reconstruído com CDA GoS-3 com $CR = 6,37$ e $PRD = 4,74$ . . . . .	78
6.29	Registro 100 reconstruído com CDA GoS-5 com $CR = 8,66$ e $PRD = 3,59$ . . . . .	79
6.30	Registro 107 reconstruído com CDA GoS-5 com $CR = 10,19$ e $PRD = 6,62$ . . . . .	79
6.31	Registro 100 reconstruído com CDT GoS-3 com $CR = 9,95$ e $PRD = 3,35$ . . . . .	80
6.32	Registro 107 reconstruído com CDT GoS-3 com $CR = 15,00$ e $PRD = 5,20$ . . . . .	80
6.33	Registro 100 reconstruído com CDT GoS-5 com $CR = 10,96$ e $PRD = 3,75$ . . . . .	81
6.34	Registro 107 reconstruído com CDT GoS-5 com $CR = 7,35$ e $PRD = 5,20$ . . . . .	81
7.1	Codificação de resíduo de ECG. . . . .	86
7.2	Esquema do CODEC DCT com utilização do resíduo de ECG. . . . .	87
7.3	Sinal residual do registro 117 gerado a partir da equação (7.4). . . . .	88
7.4	Blocos de bandas de coeficientes geradas pela transformada DCT. . . . .	89
7.5	Resíduo - Histograma de bandas de coeficientes BC: (a) $BC_0$ , (b) $BC_1$ , (c) $BC_2$ , (d) $BC_3$ , ..., (p) $BC_{15}$ . . . . .	89
7.6	Sinal ECG completo - Histograma de bandas de coeficientes BC: (a) $BC_0$ , (b) $BC_1$ , (c) $BC_2$ , (d) $BC_3$ , ..., (p) $BC_{15}$ . . . . .	90
7.7	WZSs do registro 107. (a) Sinal completo; (b) Resíduo . . . . .	92
7.8	WZSs do registro 117. (a) Sinal completo; (b) Resíduo . . . . .	92
7.9	Desempenho de compressão do CDTR GoS-3 e GoS-5 dos registros 100, 102, 107, 115, 117 e 119. (a) CDTR GoS-3; (b) CDTR GoS-5; (c) Média de CDTR GoS-3 e GoS-5 . . . . .	94
7.10	Média do desempenho de compressão do CDA, CDT e CDTR GoS-3 e GoS-5 . . . . .	95
7.11	Registro 100 reconstruído com CDTR GoS-3 com $CR = 35,69$ e $PRD = 2,75$ . . . . .	95
7.12	Registro 100 reconstruído com CDTR GoS-5 com $CR = 11,51$ e $PRD = 3,52$ . . . . .	96

---

7.13	Registro 107 reconstruído com CDTR GoS-3 com $CR = 19,12$ e $PRD = 4,08$ . . . .	96
7.14	Registro 107 reconstruído com CDTR GoS-5 com $CR = 9,18$ e $PRD = 5,64$ . . . .	97
8.1	Estrutura do CDT com estimador da taxa de perfuração. . . . .	98
8.2	Estrutura de controle da taxa de perfuração. . . . .	101
8.3	Porcentagem de <i>bits</i> excedidos dos registros 100, 102, 107, 115, 117 e 119 do CDT sem canal de retorno, utilizando GoS-3 e GoS-5 . . . . .	102
8.4	Desempenho de taxa de perfuração do registro 107 do CDT GoS-3 com e sem canal de retorno. . . . .	102
8.5	Desempenho de taxa de perfuração do registro 117 do CDT GoS-3 com e sem canal de retorno. . . . .	103
8.6	Desempenho de taxa de perfuração do registro 107 do CDT GoS-5 com e sem canal de retorno. . . . .	103
8.7	Desempenho de taxa de perfuração do registro 117 do CDT GoS-5 sem canal de retorno. . . . .	103
8.8	Desempenho de compressão do CDT sem canal de retorno, (a) CDT sem retorno GoS-3, (b) CDT sem retorno GoS-5 e (c) a média de resultados de CDT sem retorno, utilizando GoS-3 e GoS-5. . . . .	106
8.9	Média do desempenho de compressão dos CODEC's CDT com e sem canal de retorno, utilizando GoS-3 e GoS-5. . . . .	107

# Índice de Tabelas

2.1	Sumário dos trabalhos relacionados de codificação distribuída. . . . .	9
6.1	Comparação de desempenho ( <i>PRD</i> x <i>CR</i> ) entre os algoritmos CDA, CDT, e algoritmos da literatura [17, 38, 69–71]. . . . .	74
6.2	Etapas relevantes de algoritmos da literatura [38, 69–71] e seus valores de complexidade em notação $O(f)$ . . . . .	83
6.3	Etapas do codificador do CDA e CDT e seus valores de complexidades em notação $O(f)$ . . . . .	83
7.1	Comparação de desempenho ( <i>PRD</i> x <i>CR</i> ) entre os algoritmos CDA, CDT e CDTR e algoritmos da literatura [17, 38, 69–71, 71] . . . . .	91
8.1	Comparação de desempenho ( <i>PRD</i> x <i>CR</i> ) entre os algoritmos CDA, CDT, CDTR, CDT sem retorno e algoritmos da literatura [14, 38, 69–71, 84, 90], . . . . .	105
8.2	Desempenho de compressão das metodologias apresentadas nesta dissertação utilizando diferentes valores de taxas de compressão de KSs . . . . .	108

# Abreviações

**AZTEC** - *Amplitude Zone Time Epoch Coding*

**BCH** - *Bose Chaudhury Hocquenghem*

**CDA** - *Codificador no Domínio da Amostra*

**CDT** - *Codificador no Domínio da Transformada*

**CDTR** - *Codificador no Domínio da Transformada Residual*

**CODEC** - *enCOder/DECoder*

**CR** - *Compression Ratio*

**CPU** - *Computer Processing Unit*

**DSC** - *Distributed Source Coding*

**DCT** - *Discrete Cosine Transform*

**DC** - *Direct Current*

**DWT** - *Discret Wavellet Transform*

**GoS** - *Group of Segments*

**HPF** - *High Pass Filter*

**HPF** - *High Pass Filter*

**IDCT** - *Inverse Discrete Cosine Transform*

**JPEG** - *Joint Photographic Experts Group*

**KS** - *Key Segments*

**LDPCA** - *Low Density Parity Check Accumulate*

**LLR** - *Log Likelihood Ratio*

**LPF** - *Low Pass Filter*

**MAP** - *Maximum A posteriori Probability*

**MIT-BIH** - *Massachussetts Istitute Technology - Beth Israel Hospital*

**MMP** - *Multidimensional Multscale posteriori Parser*

**NSA** - *Nódulo Sino Atrial*

**PRD** - *Percent Root mean square Difference*

**RSC** - *Recursivo Sstemático Convolucional*

**SI** - *Side Information*

# Capítulo 1

## Introdução

Nas últimas décadas, a engenharia biomédica evoluiu tanto em técnicas quanto em equipamentos, sendo que estes últimos permitiram novos tratamentos, avaliações mais precisas e melhor acompanhamento de pacientes. Nesse contexto, os bio-sinais, ou seja, sinais adquiridos de seres vivos, desempenham papel de extrema importância, pois estes podem ser digitalizados e tanto analisados por médicos quanto processados por sistemas computacionais, seja para avaliação de enfermidades, montagem de histórico clínico ou mesmo determinação de diagnóstico. Dentre os bio-sinais mais conhecidos, é possível destacar o eletromiograma (EMG), que representa impulsos elétricos enviados pelo sistema nervoso humano a fibras musculares [1], o eletroencefalograma (EEG) [2], que permite a visualização da atividade elétrica de cérebros humanos, e o eletrocardiograma (ECG), que representa a atividade elétrica do coração, durante batimentos cardíacos [3]. Com relação a este último, que é obtido através de eletrodos em contato com a pele humana, durante um período, a informação obtida está relacionada ao processo de despolarização e repolarização de músculos cardíacos. Além disso, o sinal de ECG é um dos mais importantes sinais da medicina, devido a sua capacidade de exibir a atividade elétrica do coração através de gráficos e, conseqüentemente, por meio de análise, ajudar no diagnóstico de diversas enfermidades, como arritmias, sobrecargas, áreas eletricamente inativas, alterações de repolarização, bloqueio de ramo, angina e infarto do miocárdio [4].

## 1.1 Caracterização do Problema

Registros de ECG podem tanto ser capturados em consultórios, com equipamentos modernos, quanto por equipamentos portáteis, como o *Holter* de ECG [3], quando existe a necessidade de registro por um longo período de tempo. Entretanto, tais sinais podem ser capturados com muitas derivações [3], o que leva a altas exigências de circuitaria eletrônica e memória, sendo esta última relacionada ao armazenamento dos resultados das medições. Além disso, como pode ser necessário transmitir esses exames, sistemas de comunicação e suas respectivas bandas são também utilizados. Devido a isso, vale ressaltar que tanto memória quanto bandas de comunicação são recursos escassos e que precisam ser poupados, o que leva ao uso de técnicas de compressão.

A compressão de sinais, em geral, pode ser realizada de duas maneiras: com e sem perda de dados. Na forma sem perdas, os fatores de compressão alcançados são geralmente menores, pois apenas explora-se a organização dos respectivos dados e nenhuma informação é descartada [5], ou seja, o sinal reconstruído é exatamente igual ao original. Entretanto, quando a compressão é executada com perdas, há descarte de dados não necessários, de acordo com a aplicação específica, o que significa que a informação reconstruída é diferente da original [5] e maiores taxas de compressão são obtidas. Por exemplo, em compressão de imagens, informação de alta frequência, que não é facilmente percebida pelo olho humano, é normalmente retirada, ao passo que, em compressão de áudio, remove-se aquilo que não é ouvido, principalmente devido a processos de mascaramento [5]. Segundo Sayood [5], a compressão ocorre através da exploração da redundância dos dados, gerando então uma versão compactada desse conjunto, através de técnicas de codificação. O processo de decodificação, ou seja, a operação inversa, pode reconstruir a forma exata ou uma aproximada do sinal original, dependendo da compressão utilizada, como visto anteriormente.

Sistemas de codificação modernos exploram conjuntos de dados de forma estatística, gerando então uma nova representação destes [5], o que geralmente exige muito processamento do codificador, de modo que a representação compacta seja criada. Tal situação é ideal para sistemas ponto-multiponto, como, por exemplo, transmissão de sinais de radiodifusão, onde um transmissor de uma emissora possui um codificador que realiza inúmeros processos e decodificadores (receptores), que fazem o processo inverso com mais simplicidade. Contudo, tal padrão não é o ideal para sistemas multiponto-ponto, como redes de sensores sem fio [6], pois neles

é necessário utilizar metodologias de codificação que processe sinais com mais simplicidade. Essa percepção também é verdade para sistemas embarcados com baixo poder de processamento, como dispositivos *Holter* de ECG, que são muito utilizados para monitoramento de pacientes cardíacos, como já mencionado, e permitem avaliação fora de consultórios e sem o suporte de sistemas médicos de alto valor.

Nesse sentido, a comunidade científica tem produzido inúmeros trabalhos [7–14], na maior parte relacionados à codificação de vídeo. Além disso, com a aplicação de conceitos de redes distribuídas [15], codificadores tornam-se simples e transferem a maior parte da exploração estatística aos respectivos decodificadores. De fato, a compressão distribuída de sinais (*Distributed Source Coding* - DSC) [6] é a área de pesquisa que fornece ferramentas para a exploração de dependências estatísticas de conjuntos de dados em decodificadores e a sua base está na aplicação dos conceitos de *cosets*, conforme exposto por Forney [16].

Diferentemente de outros estudos concentrados na codificação de vídeo, este trabalho propõem uma metodologia de compressão de sinais de ECG fundamentada na teoria de codificação distribuída. Ainda nessa metodologia, uma etapa de transformada discreta do cosseno (*discrete cosine transform* - DCT) foi acrescentada, gerando assim um segundo esquema. Técnicas de pré-processamento foram utilizadas, como separação de períodos de ECG em grupos de segmentos. Além disso, um esquema baseado em codificação de resíduos de ECG também foi desenvolvido, o qual permite melhores desempenhos, quando comparado à codificação do sinal ECG completo. Por fim, uma metodologia para a definição da quantidade de informação a ser enviada ao decodificador foi criada, a qual elimina o canal de retorno (*feedback*) entre decodificador e codificador.

## 1.2 Objetivo

O objetivo geral desta dissertação é a implementação de uma metodologia de compressão de sinais de ECG baseada em conceitos de codificação distribuída, para utilização em aplicações com baixo poder computacional na ponta de aquisição, o que permite assim menor utilização de memória e largura de banda reduzida.

### 1.2.1 Objetivos Específicos

- Compreender os conceitos teóricos de DSC;
- Explorar trabalhos diretamente relacionados ao tema proposto;
- Implementar uma metodologia baseada em DSC para comprimir sinais de ECG;
- Utilizar técnicas de pré-processamento que aumentem a correlação do sinal gerado e assim melhorem os resultados de compressão;
- Realizar experimentos utilizando a metodologia de DSC proposta;
- Investigar e descrever os resultados obtidos das metodologias implementadas;

## 1.3 Contribuições

A principal contribuição deste trabalho reside na metodologia de compressão distribuída proposta. Entretanto, contribuições específicas também podem ser citadas, como:

- Organização dos dados através da utilização de grupos de segmentos (*Group of Segments* - GoS). Adequando assim a utilização do sinal de ECG à estrutura de codificação distribuída;
- Codificação de resíduo de ECG em vez da codificação do sinal completo, no domínio da transformada, o que também pode ser empregado em outros esquemas de compressão. Permitindo assim desempenho de compressão superior em relação a codificação do sinal completo;
- Eliminação do canal de retorno entre decodificador e codificador, geralmente utilizado para informar a redundância necessária ao código corretor de erros base (no presente caso, códigos turbo), de forma a proporcionar compressão satisfatória com menor necessidade de *hardware* que a solução com canal de retorno;

Sendo assim, delinea-se o caminho para compressão de sinais de ECG utilizando técnicas de DSC, com aplicação em sistemas embarcados e portáteis e potencial para promover maior utilização por parte de pacientes [17]. Para isso, geraram-se implementações de referência nos

domínios da amostra e da transformada, com utilização de técnicas de pré-processamento de sinais de ECG. Por fim, a metodologia básica é composta por um codificador turbo operando em conjunto com um codificador convencional, conforme a metodologia de fontes distribuídas, obtendo-se assim resultados que demonstraram a sua viabilidade.

## 1.4 Organização da Dissertação

Neste capítulo, o contexto e a motivação ao tema deste estudo foram descritos. O Capítulo 2 mostra os trabalhos relacionados à esta dissertação. Em seguida, o Capítulo 3, por sua vez, mostra uma breve introdução sobre o sinal ECG, incluindo a captura com várias derivações e informações sobre o dispositivo de medição *Holter*, e o Capítulo 4 descreve as metodologias atuais de compressão de ECG. O Capítulo 5 aborda a metodologia de codificação distribuída, incluindo uma breve descrição dos trabalhos de Slepian-Wolf [18] e Wyner-Ziv [15], além de tópicos relacionados à esse assunto, como a utilização de códigos corretores de erros, como os códigos turbo. No Capítulo 6, a metodologia baseada em compressão distribuída é descrita, com apresentação de abordagens no domínio da amostra e da transformada para a compressão de sinais de ECG. O Capítulo 7 apresenta uma metodologia que utiliza o resíduo de sinais de ECG, em vez do sinal ECG completo. O Capítulo 8 mostra uma abordagem do codificador que elimina o canal de retorno (*feedback*). Por fim, o capítulo 9 mostra as conclusões acerca das metodologias apresentadas.

## Capítulo 2

### Trabalhos Relacionados

A codificação distribuída teve seu início em 1973, com o trabalho de David Slepian e Jack Wolf [18], o qual consistiu em teoria relativa à codificação, sem perdas, de duas fontes correlacionadas, o que seria feito com codificação separada e decodificação conjunta de dados. Posteriormente, em 1976, Aaron D. Wyner e Jacob Ziv [15] estenderam a pesquisa de Slepian e Wolf e apresentaram um trabalho complementar relativo à DSC, com desenvolvimento de teoria taxa-distorção. Apesar desses trabalhos datarem da década de 70, tentativas práticas baseadas na ideia de DSC foram iniciadas somente 30 anos depois, desta vez por Pradhan e Ramchandran [19].

Desde então, a pesquisa em compressão distribuída tem sido bastante expressiva. Existem inúmeras publicações com aplicações em compressão de vídeo, que é uma das áreas mais exploradas pela metodologia DSC [7, 10–14, 17]. Um exemplo é o trabalho de Aaron , Zhang e Girod [7], os quais propõem uma metodologia de compressão distribuída de vídeo composta por um codificador de baixa complexidade, com bom desempenho. Para isso, os autores propuseram uma metodologia de compressão assimétrica, onde quadros são codificados de forma direta e independentemente (codificação *intraframe*) e decodificados condicionalmente (decodificação *interframe*). Isso está diretamente relacionado às fontes  $X$  e  $Y$  da metodologia DSC de Slepian e Wolf [18], sendo a primeira referente aos quadros pares e a segunda aos ímpares. Com isso,  $X$  é comprimido por um codificador *intraframe*, sem conhecimento de  $Y$ , e a sua decodificação é realizada com a ajuda de informação lateral, formada por quadros interpolados a partir de quadros  $Y$  adjacentes. O processo de codificação é realizado através de códigos turbo e, durante a reconstrução, uma função que limita o valor da distorção para um valor máximo,

com auxílio de informação lateral, é utilizada.

Aaron, Rane e Girod [13] incorporaram a DCT à metodologia anterior, utilizando uma segmentação de quadros diferente. Assim como na metodologia base, quadros são separados em dois grupos: quadros Wyner-Ziv e quadros-chaves, ambos resultantes da segmentação de vídeo. Dessa forma, eles implementaram a segmentação utilizando diferentes tamanhos de quadros, proporcionando diferentes relações taxas-distorção, conforme a segmentação. Além disso, apenas o quadro reconstruído anterior foi utilizado como informação lateral, com o auxílio do envio de *hash bits*, que permitem uma compensação de movimentos mais precisa e não utilizam a interpolação dos quadros-chaves adjacentes.

Já no trabalho de Brites, Ascenso e Pereira [9], que foi realizado no domínio da amostra e com aplicação em codificação de vídeo, mostra-se que a eficiência das soluções de codificação Wyner-Ziv depende da capacidade de modelar a dependência estatística entre a informação original, no codificador, e a informação lateral computada no decodificador. Tal constatação implica em complexidade, pois a informação original não está disponível no decodificador e a qualidade da informação lateral varia ao longo da sequência, ou seja, a distribuição de erro não é constante. Por exemplo, quando há muito movimento em uma sequência, é mais difícil prever o quadro Wyner-Ziv e, dessa forma, erros na informação lateral aumentam significativamente. Sendo assim, os autores mencionados propuseram um novo método para a estimativa do modelo de ruído de correlação, o que funciona de forma adaptativa.

Trabalhos mais recentes e também aplicados à codificação de vídeo, como o desenvolvido por Vijayanagar *et al.* [10], os quais utilizam uma segmentação de quadros diferente da empregada por Aaron, Zhang e Girod [7], introduziram grupos de segmentos com tamanhos adaptáveis à estatística de vídeo, sem aumentar latência e sem a necessidade de armazenar quadros adicionais no codificador, sendo a codificação baseada em códigos *Bose-Chaudhuri-Hocquenghem* (BCH).

Recentemente, sinais de ECG começaram a ser explorados com metodologia DSC [14, 17]. Por exemplo, o trabalho de Lima Filho [17], que utiliza compressão no domínio da amostra e também da transformada, o qual alcançou resultados satisfatórios, se comparados a esquemas de compressão tradicional apresentados. Além disso, a utilização de técnicas de pré-processamento, nesse mesmo trabalho, apresentaram melhoria de desempenho, devido à reorganização proporcionada aos dados disponíveis, de modo a facilitar a exploração de redundância. De fato, registros de ECG são considerados sinais quase periódicos [20], o que

foi explorado com a normalização de todos os períodos existentes [21], além da eliminação de patamares e níveis DC. Após esse processamento, uma técnica que aumenta a correlação entre partes desses sinais foi empregada, a qual é essencial na compressão distribuída e ordena períodos por ordem de similaridade [17].

No trabalho de Wu, Tsai e Chang [14], aplicado à compressão de sinais de ECG, uma técnica de pré-processamento semelhante à utilizada por Lima Filho foi empregada [17]. Além disso, a sua máquina de codificação é baseada na técnica de quantização vetorial proposta por Sun e Tai [22], com treinamento de diferentes sinais de ECG para a criação de um cabeçalho utilizado na reconstrução de sinais quantizados. O CODEC utilizado foi do tipo códigos turbo e os experimentos apresentados utilizaram a base de dados disponibilizados pelo MIT-BIH [23].

Outro trabalho interessante e fundamentado em DSC aplicada à vídeo, é o dos autores Lee, Tillo e Jeon [24], que implementaram uma metodologia capaz de eliminar o canal de retorno. Para isso, esses autores basearam-se nos princípios descobertos por Claude Shannon e a sua teoria de entropia da informação.

A metodologia proposta neste trabalho utiliza algumas técnicas comentadas nos parágrafos anteriores e sumarizadas na Tabela 2.1, ao mesmo tempo que propõe novas metodologias adaptadas a sinais de ECG. Nesta dissertação, uma metodologia semelhante à utilizada pelos autores Aaron, Zhang e Girod [7] e baseada no domínio da amostra foi implementada, a qual utiliza códigos turbo e implementa a etapa de DCT de Aaron, Rane e Girod [13], possibilitando assim maior compactação de energia. Sendo isso muito interessante para sinais de ECG, pois estes possuem a maior parte da sua energia concentrada em frequências baixas. Nesta dissertação, a técnica de perfuração criada por Brites, Ascenso e Pereira [9] foi utilizada, a qual seleciona a informação de paridade e funciona através da solicitação de *bits* por um canal de retorno.

Técnicas de pré-processamento desenvolvidas por Lima Filho *et al.* [26] também foram utilizadas, as quais se baseiam em normalização, equalização e reordenação de segmentos de ECG e tem o potencial de aumentar o desempenho global da compressão. Outra técnica utilizada foi a de segmentação de informação em GoS de tamanhos diferentes [13], com a escolha de duas larguras de grupos que experimentalmente apresentaram bons resultados, de forma diferente de Vijayanagar *et al.* [10]. De fato, este último implementou grupos de segmentos com tamanhos adaptados, de acordo com estatísticas de quadros inseridos no codificador.

Em resumo, o presente trabalho propõe uma metodologia base parecida com a utilizada

Tabela 2.1: Sumário dos trabalhos relacionados de codificação distribuída.

Referência	Aplicação	Codificação	Métodos
Aaron, Zhang e Girod [7]	Compressão de vídeo no domínio da amostra	Quadros Wyner-Ziv: códigos turbo; Quadros-chaves: reconstrução perfeita (Não foi utilizado codificação).	Segmentação: Quadros pares e ímpares; Quantização uniforme; Utiliza canal de retorno; Etapa de perfuração.
Aaron, Rane e Girod [13]	Compressão de vídeo no domínio da transformada	Quadros Wyner-Ziv: códigos turbo; Quadros-chaves: H.263.	Transformada DCT; Segmentação: tamanhos diferentes; Informação lateral refinada através de hash bits ; Quantização uniforme ; Utiliza canal de retorno; Etapa de perfuração.
Brites, Ascenso e Pereira [9]	Compressão de vídeo no domínio da amostra	Quadros Wyner-Ziv - códigos turbo; Quadros-chaves H.263.	Códigos turbo ; Segmentação: quadros pares e ímpares; Modelagem adaptativa do ruído de correlação; Quantização uniforme ; Utiliza canal de retorno; Etapa de perfuração.
Vijayanagar <i>et al.</i> [10]	Compressão de vídeo no domínio da amostra	Quadros Wyner-Ziv : código BCH;	Código BCH;
Lee, Tillo e jeon [25]	Compressão de vídeo	Quadros Wyner-Ziv: H.264; Quadros-chaves: DCT: H.264.	Segmentação dinâmica de quadros. Segmentação: quadros pares e ímpares; Propôs metodologia que elimina canal de retorno.
Lima Filho [17]	Compressão de ECG no domínio da amostra e da transformada	Quadros Wyner-Ziv: códigos Turbo; Quadros-chaves: Sem distorção.	Técnicas de pré-processamento; Segmentação: separação em grupos de segmentos; Quantização uniforme (domínio da amostra) e em torno de zero (domínio da transformada); Transformada DCT inteira.
Wu <i>et al.</i> [14]	Compressão de ECG no domínio da amostra	Quadros Wyner-Ziv: códigos turbo; Quadros-chaves: sem codificação; reconstrução perfeita foi considerada.	Pré-processamento (Normalização do período); Quantização vetorial (geração de dicionário através de treinamento). Técnicas de pré-processamento;
Proposta	Compressão de ECG no domínio da amostra e da transformada	Quadros Wyner-Ziv: códigos turbo; Quadros-chaves: sem distorção.	Segmentação: separação em grupos de segmentos; Quantização uniforme (domínio da amostra) e em torno de zero (domínio da transformada); Transformada DCT inteira; Codificação residual; Eliminação do canal de retorno.

---

por Aaron, Zhang e Girod [7], com codificadores no domínio da amostra e da transformada DCT [13], mas com novas abordagens de pré-processamento e codificação (resíduos), remoção de canal de retorno e diversas adaptações e abordagens relacionadas a sinais de ECG.

# Capítulo 3

## O sinal de ECG

Neste capítulo, conceitos sobre sinais de ECG serão apresentados, levando-se em consideração sua origem, estrutura, métodos de captura e uma breve análise do equipamento portátil *Holter* de ECG, o qual tem a função de gravar sinais de ECG durante longos períodos, geralmente entre 24 a 48 horas.

### 3.1 A origem do sinal de ECG

Um dos idealizadores da estrutura do sinal de ECG foi Willem Einthoven (1860-1927), o qual tinha o objetivo de mostrar a atividade elétrica originária do coração, utilizando um equipamento chamado galvanômetro [27]. Desde então, inúmeros aparelhos conhecidos como eletrocardiógrafos surgiram, nesses mais de 100 anos que passaram. Além disso, tais sistemas foram modernizados e simplificados, o que tornou o seu uso extremamente útil e fundamental para a interpretação de patologias cardiocirculatórias [27]. Em resumo, o eletrocardiógrafo é um equipamento que mede a diferença de potencial entre dois pontos ou mais, ou seja, pequenas variações de correntes em eletrodos aplicados a pontos estratégicos [3] do corpo humano, enquanto registra tais variações elétricas, ao longo do tempo.

O coração é basicamente um músculo especializado em movimentar o sangue presente no corpo humano, através de veias e artérias. Ele é composto por células cardíacas semelhantes a todas as outras células do organismo, que tem, em repouso, o meio intracelular negativo em relação ao extracelular, conforme mostrado na Figura 3.1 [28]. Essa distribuição de cargas (positivas no exterior e negativas no interior) é uniforme e, devido a isso, a célula cardíaca

normal em repouso é chamada de "célula polarizada", sendo esta rica em Potássio, na sua parte interna. Quando uma ativação elétrica originária do nódulo sinoatrial ocorre, trocas iônicas também acontecem e a polaridade dessas células é invertida, o que gera uma frente de onda de despolarização, que é propagada pelo sistema de condução do coração e células adjacentes. Uma pequena parte dessa corrente se propaga até a superfície do corpo humano, onde é captada por sensores de um equipamento eletrocardiográfico [28].

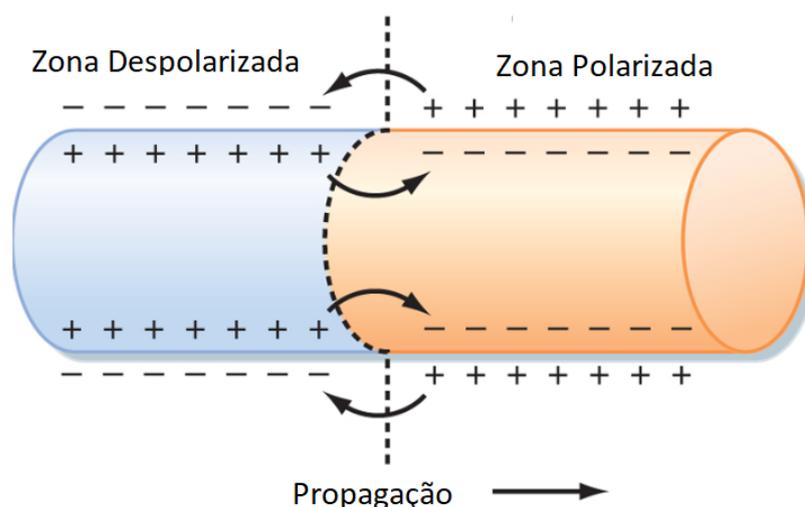


Figura 3.1: Ilustração da propagação elétrica em uma célula cardíaca  
FONTE: Berne e Levy (2009) [29].

O ciclo cardíaco é responsável por manter o funcionamento do coração e é composto por dois movimentos básicos: a sístole, que expelle o sangue dos ventrículos, e a diástole, onde os ventrículos recebem sangue dos átrios. A sístole e a diástole são controladas através de polarização e despolarização de células cardíacas, com impulsos elétricos originários do nódulo sinoatrial (NSA) [28], conhecido também como marcapasso natural (Figura 3.2). Ele gera um estímulo elétrico com frequência entre 60 a 100 impulsos por minuto, que se propaga pelos átrios e provoca a polarização e consequente contração destes, seguindo até o nó atrioventricular (AV), o qual atrasa o estímulo elétrico em aproximadamente 0,08 a 0,1s. Depois, esses estímulos prosseguem mais abaixo do coração, pela rede de *Purkinje* e através do Feixe de *His* [27], fazendo com que os ventrículos se contraíam de baixo para cima. A principal função do nó AV é transmitir estímulos elétricos dos átrios para os ventrículos, de forma sincronizada, separando assim a sístole atrial da sístole ventricular e evitando arritmias atriais ventriculares [27]. Em seguida, todo este processo é repetido. O sistema condutor do coração e seus potenciais elétricos estão ilustrados na Figura 3.2.

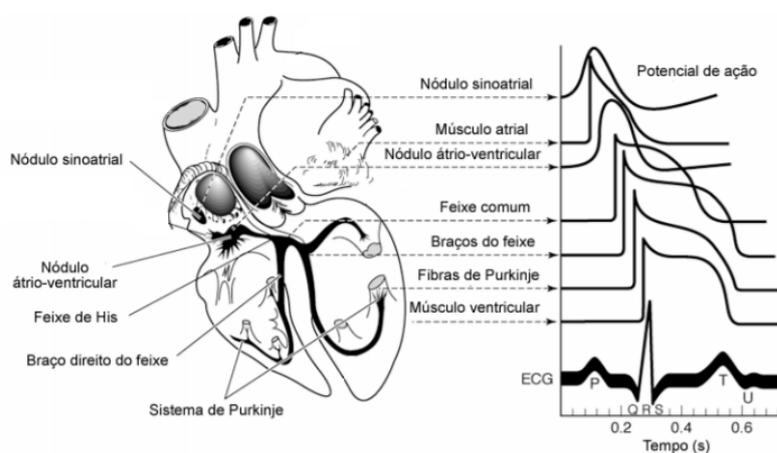


Figura 3.2: Sistema de condução do coração e os potenciais de ação típicos de cada parte do coração e a sua correlação com a atividade elétrica adquirida na pele (ECG).

FONTE: Figura adaptada de Barret *et al.* (2014) [30].

Os ramos de feixes de *His* são a continuação do nó AV e penetram o corpo fibroso pelo interior do coração, até a rede de *Purkinje*, sendo responsáveis por causar a despolarização dos ventrículos e a sua consequente contração, o que transmite o estímulo elétrico criado no nó sinoatrial [27]. O ventrículo direito e o átrio esquerdo são interligados ao pulmão, que expelle o dióxido de carbono e injeta oxigênio no sangue. O ventrículo esquerdo e o átrio direito, por outro lado, são conectados ao resto do corpo humano. Toda essa estrutura do coração é contraída e relaxada de forma sincronizada, como visto anteriormente, através do sistema de condução cardíaco [27].

## 3.2 Morfologia do sinal de ECG

Um ciclo completo de um batimento cardíaco é composto pelas ondas *P*, *Q*, *R*, *S* e *T*, conforme a Figura 3.3. Além disso, outro fator importante a se considerar é a frequência cardíaca, que é considerada normal quando está em torno de 72 batimentos por minuto (BPM). Entretanto, em situações anômalas, esta pode aumentar ou diminuir: quando acima de 100 BPM, considera-se que o respectivo sujeito está em estado de taquicardia e, quando abaixo de 60 BPM, considera-se que há braquicardia. A onda *P*, que é a primeira parte do seguimento completo, representa a despolarização do átrio direito, ao passo que a segunda parte compõe a despolarização do átrio esquerdo. A função desta despolarização é contrair os átrios, até que os ventrículos fiquem repletos de sangue da cavidade atrial [4]. Já a polarização dos átrios não é mostrada no ECG, pois ocorre ao mesmo tempo que a despolarização ventricular, a qual possui

maior amplitude, devido à sua maior massa muscular. A onda *P* possui amplitude máxima de 2,5 mm e duração inferior a 110 ms [4]. O intervalo *PR*, que é o seguimento do final da onda *P* até o início do complexo *QRS*, representa o tempo em que ocorre a ativação atrial e a ativação ventricular, tendo duração de 0,12s a 0,20s. O intervalo *PR* varia de acordo com idade e frequência cardíaca [4].

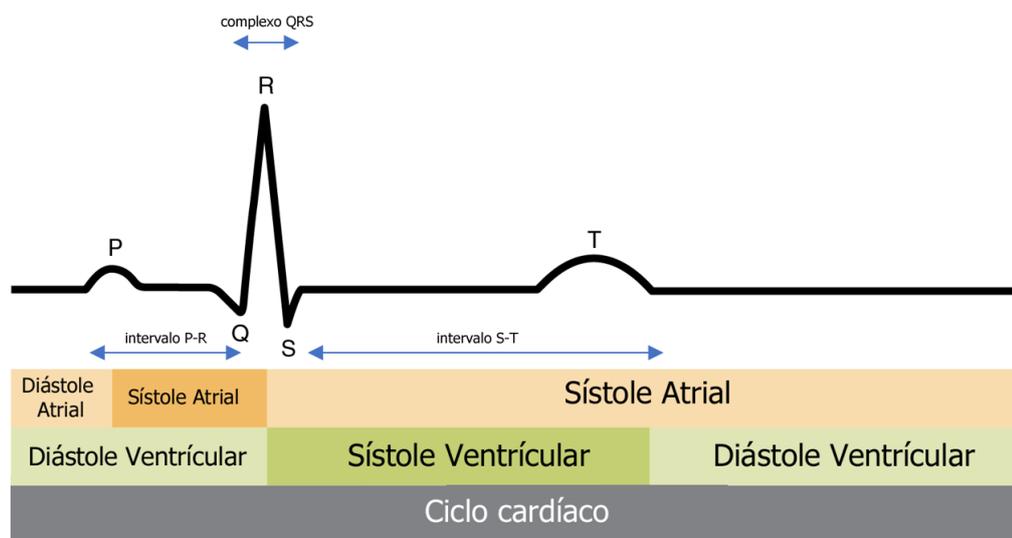


Figura 3.3: Período completo de um sinal de ECG.

FONTE: adaptado de mit.edu [23].

O complexo *QRS* representa três ondas originárias da contração ventricular, ou seja, a ativação ventricular, e produz deflexões de tamanho e polaridade variável [31]. A duração do complexo *QRS* tende a ser maior quanto mais idoso é o indivíduo e menor é a frequência cardíaca, assim como acontece com a onda *P*, sendo também maior em atletas e pessoas que apresentam certo retardo na porção final do complexo *QRS* [31]. O seguimento *QRS* é considerado normal quando a sua duração for menor que 120 ms e a sua amplitude estiver entre 5 e 20 mm, nas derivações periféricas, e entre 10 e 30 mm, nas derivações precordiais [4].

O segmento *ST* inicia após o complexo *QRS* e vai até o início da onda *T*, sendo nomeado independentemente do complexo ventricular acabar na onda *R* ou na onda *S*. O início da onda *T*, em traçados normais, quase não é notada e por isso há dificuldade em se medir a sua duração. Esse segmento representa o tempo que decorre desde o fim da despolarização até o início da repolarização dos ventrículos; porém, desvios pequenos são frequentes e, quando não ultrapassam 0,05 ou, no máximo, 0,1 mV, são considerados normais [31]. A onda *T* representa a repolarização ventricular e, em geral, apresenta polaridade igual à maior onda do complexo *QRS*. Ela não

é medida isoladamente e o seu valor está geralmente incluso no cálculo do intervalo  $QT$  [31].

### 3.3 Derivações

Os impulsos elétricos que percorrem o sistema de condução do coração são captados através de até 12 derivações básicas, que são as diferentes posições dos eletrodos em contato com a superfície da pele humana. Tais derivações podem ser unipolares, bipolares e pré-cordiais, dependendo das posições dos sensores [28].

As derivações bipolares medem a diferença de potencial entre membros ou determinadas partes do corpo e o coração [32]. Em resumo, colocam-se pares de eletrodos em cada braço e também um par na perna esquerda, dando origem assim ao triângulo de *Einthoven* [28], conforme a Figura 3.4, e na perna direita, colocando-se um fio terra para estabilização. O sentido da polaridade, seja ela positiva ou negativa, é determinada pelas derivações DI, DII e DIII [32].

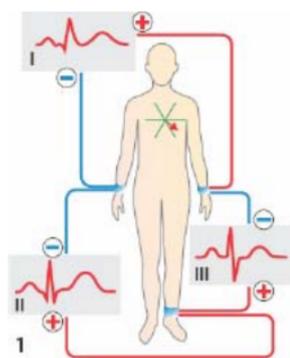


Figura 3.4: Sistema de derivações bipolares. Representação visual das formas de onda.  
FONTE: adaptado de DESPOPOULOS e SILBERNAGL (2003) [33].

O sistema unipolar, ilustrado na Figura 3.5 e proposto por Emanuel Goldberger, em 1942, é composto por 3 derivações, ou seja, aVR, aVL e aVF, cujas letras iniciais "a" indicam que as tensões "V" foram aumentadas, ou seja, sofreram um ganho de até 50%, e as letras R,L e F são referentes respectivamente a: *right* (braço direito), *left* (braço esquerdo) e *foot* (pé) [32].

Outro tipo de derivação unipolar, também conhecido como derivação pré-cordial, tem o intuito de mostrar a forma mais fidedigna do sinal de ECG. Frank Wilson, que propôs tal ideia, por volta de 1930, apresentou seis derivações pré-cordiais [32]: V1, V2, V3, V4, V5 e V6. Este sistema e suas respectivas formas de ondas estão mostrados na Figura 3.6.

Cada ponto do sinal de ECG é resultado de polarização ou despolarização de células do coração, em determinadas áreas deste [32]. A derivação aVR reflete a atividade dos ventrícu-

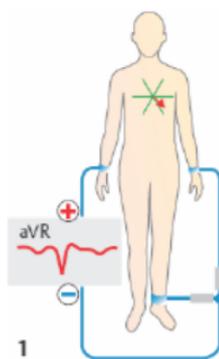


Figura 3.5: Sistema de derivações unipolares. Exemplo da obtenção da derivação aVR.  
 FONTE: adaptado de DESPOPOULOS e SILBERNAGL (2003) [33]

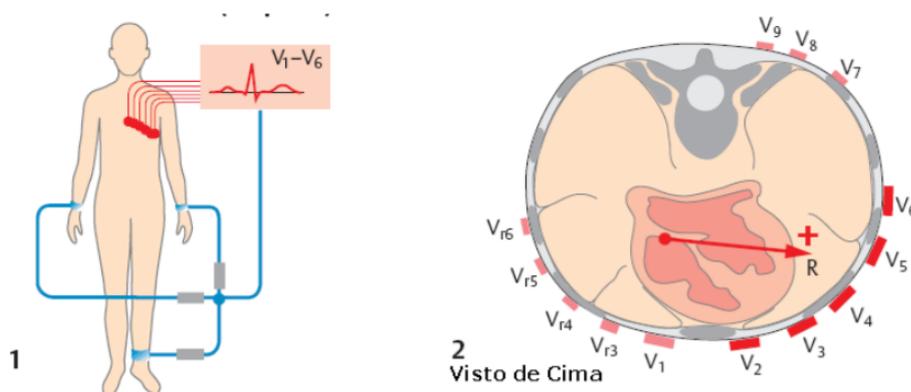


Figura 3.6: Sistema de derivações pré-cordiais: (1) eletrodos posicionados no peito e terminal central conectando os membros.

FONTE: adaptado de DESPOPOULOS e SILBERNAGL (2003) [33].

los, ao passo que a despolarização dos átrios, a despolarização ventricular e a repolarização ventricular estão em oposição a este eletrodo, o que resulta em ondas *P*, *T* e complexo *QRS* negativados [32]. As derivações aVF e aVL mostram as atividades dos ventrículos e, desse modo, a deflexão associada é positiva ou bifásica. Já nas derivações V1 e V2, a onda *Q* não aparece e o complexo *QRS* ocorre apenas como uma única elevação do sinal, devido ao fato da despolarização ventricular mover-se da esquerda para direita, no eletrodo. As derivações V4, V5 e V6 mostram uma pequena onda *Q* e uma grande onda *R* [32].

A Figura 3.7 mostra as ondas geradas em cada uma das 12 derivações, porém, a derivação denominada MLII, que é uma DII modificada, é a mais utilizada pela literatura, assim como a V1, pois esta é a mais indicada para a realização de leituras durante longos períodos de tempo, em pacientes acamados ou com dispositivo *Holter* [34]. De fato, como o presente trabalho emprega a base de dados MIT-BIH [23], os registros mais utilizados aqui são do tipo MLII e V1.

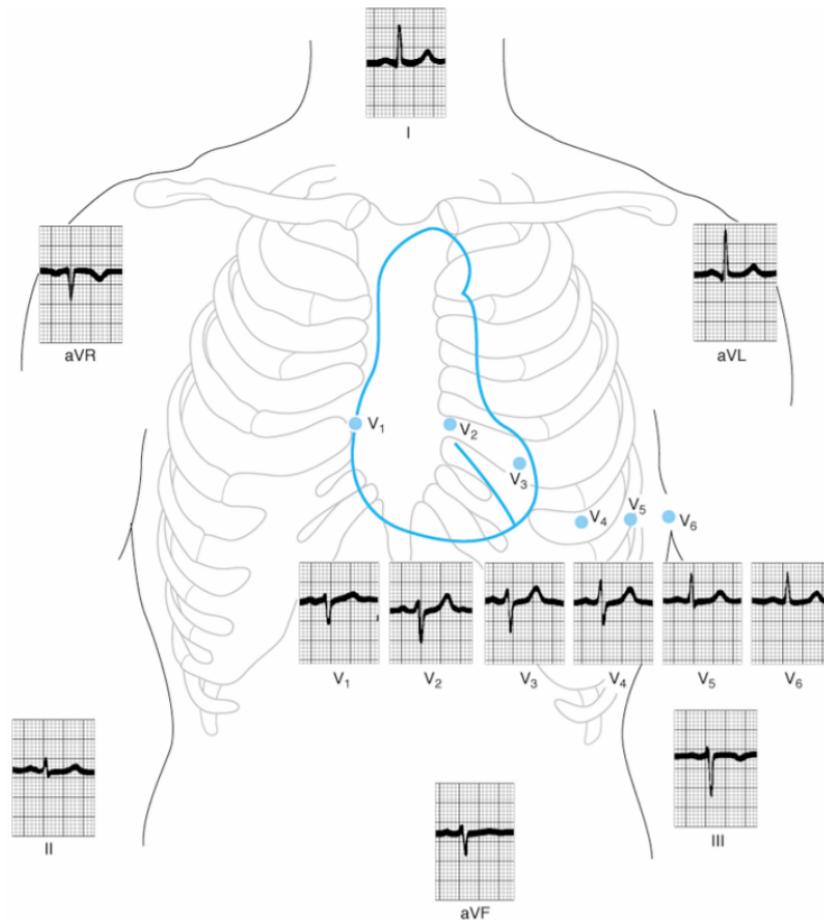


Figura 3.7: Formas de onda de 12 derivações de ECG.  
 FONTE: Barret *et al.* (2014) [30].



Figura 3.8: Ondas de ECG das derivações MLII e V1.  
 FONTE: Physionet.org.

### 3.4 O dispositivo *Holter*

Pacientes com problemas cardíacos podem ter momentos com anomalias em seus registros de ECG, pois estas patologias são muitas vezes intermitentes. Sendo assim, exames de curta duração realizados em consultórios podem não detectar tais condições [35], o que leva à necessidade de se medir a atividade elétrica do coração por um período de tempo maior, geralmente de 24 ou 48 horas, através de um aparelho portátil conhecido como *Holter* de ECG. Este pequeno aparelho, que opera com bateria, registra sinais de ECG em formato digital, durante toda a atividade diária de um paciente [35], o que demanda muita memória, em virtude da quantidade de tempo e do número de derivações utilizados.

# Capítulo 4

## Compressão de sinais de ECG

O capítulo anterior apresentou a origem do sinal de ECG, bem como a sua estrutura, o porquê dele ser útil e a necessidade de se executar a sua compressão. Como consequência natural, este capítulo apresentará o conceito de compressão de dados, o que significa comprimir um sinal e como obtemos essa nova representação, assim como mostrará os dois tipos de compressão básicos existentes e as técnicas específicas utilizadas neste trabalho.

### 4.1 Compressão de Dados

Algoritmos de compressão são utilizados para se reduzir o número de *bits* requeridos para a representação de um dado sinal [36], como, por exemplo, uma imagem, uma sequência de vídeo, áudio e sinais biomédicos, o que inclui registros de ECG. A motivação para a compressão de dados reside na limitação de armazenagem em formato digital, através de memórias e discos rígidos, e também na limitação de banda para transmissão. Mesmo com os grandes avanços observados em sistemas de transmissão de dados, que ocorreram nos últimos anos, a compressão busca reduzir banda e destinar recursos para outros serviços ou usuários, através da obtenção de taxas de compressão maiores, apresentando também menor distorção [36].

Existem vários métodos de compressão, os quais são baseados em diferentes ideias e úteis para diferentes tipos de dados, porém, todos, de uma forma ou outra, baseiam-se no mesmo princípio, ou seja, comprimir informação através da retirada de redundância presente nos dados originais, o que está mostrado na Figura 4.1 [5]. Toda fonte de dados não aleatória tem uma estrutura básica, a qual pode ser explorada para a obtenção de uma representação mais compacta.

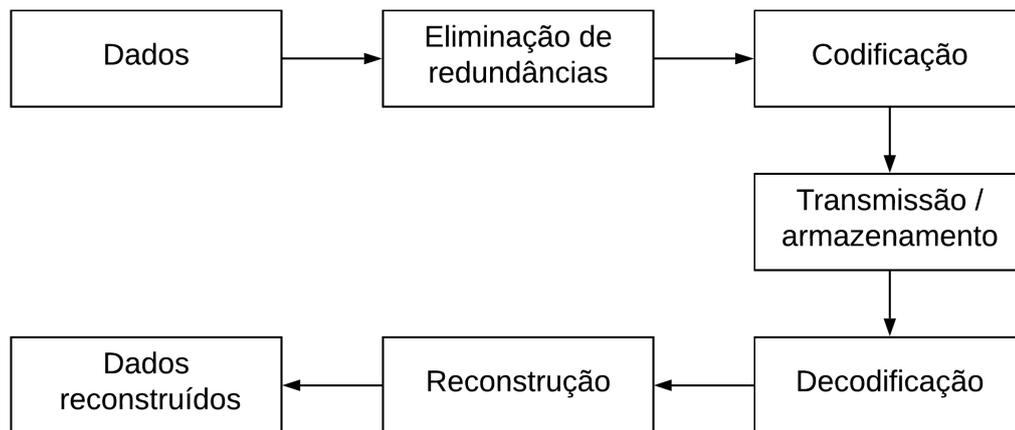


Figura 4.1: Esquema geral um sistema de compressão de dados.  
FONTE: adaptado de Faria, Fonseca e Costa (2012) [37].

É importante ressaltar que, para se melhorar a eficiência de um algoritmo de compressão, a natureza dos dados deve ser levada em consideração, bem como a sua base estatística [36]. De modo a comprimir uma sequência de dados, um algoritmo deve examiná-los, achar redundâncias e tentar retirar estas últimas; porém, tais elementos dependem da informação propriamente dita e de como ela está organizada (*e.g.*, texto, imagens, vídeo, sinais biomédicos, etc.) ou é inicialmente processada, o que explica o fato de alguns algoritmos possuírem melhor desempenho que outros, para determinado tipo de dado [36].

A literatura divide a compressão de dados em duas vertentes básicas: compressão com e sem perdas [5]. Estas, por sua vez, são subdivididas em outros modelos de compressão, sendo a compressão com perdas a mais explorada, devido ao seu maior poder de redução de tamanho [36]. Existem diversas formas de se classificar os tipos existentes de compressão de dados, o que leva em consideração diferentes metas ou aspectos de sinais explorados durante a obtenção uma forma compacta. Na compressão sem perdas, não há descarte de informação, ou seja, se essa técnica foi utilizada, então os dados originais podem ser recuperados a partir dos comprimidos [5]. Um exemplo de aplicação desse paradigma é compressão de texto, pois pequenas diferenças podem mudar completamente o significado de uma frase, o que leva à necessidade de recuperação integral da informação. No caso de compressão com perdas, existe descarte de informação e os dados originais não podem ser exatamente recuperados. Entretanto, se essa diferença puder ser tolerada, geralmente obtém-se uma compressão muito maior que a observada em sistemas sem perdas. Conceitualmente, na compressão sem perdas,

exploram-se dados redundantes e, na compressão com perdas, exploram-se dados irrelevantes. De fato, a perda de informação, na compressão com perdas, não é um problema para diversas aplicações [5], como em transmissão de sinais de áudio ou voz, pois nesta não é necessário preservar o valor exato de cada amostra recuperada, de modo a se entender o que foi dito por uma pessoa.

Existem muitas técnicas de compressão de ECG, as quais podem ser caracterizadas como: diretas, paramétricas ou baseadas em transformada [38]. Na forma direta, amostras originais são diretamente processadas para compressão [38]. Um exemplo deste tipo é o *amplitude zone time epoch coding* (AZTEC) [39], que emprega técnicas de predição para o cálculo de amostras previamente codificadas, após quantização. A técnica paramétrica explora dados para a extração de características que, na reconstrução, são utilizadas para formar um dado sinal, através de um modelo [40]. Por fim, algoritmos baseados em transformada manipulam a distribuição espectral de energia de conjuntos de dados [41].

## 4.2 Compressão direta de ECG

A compressão direta de ECG depende de algoritmos de previsão e interpolação, os quais exploram a redução de elementos repetitivos, em sucessivas amostras [41]. O algoritmo de previsão recorre ao conhecimento *a priori* de amostras anteriores, ao passo que o de interpolação explora o conhecimento *a priori* de amostras anteriores e futuras. Os principais algoritmos que empregam tal técnica são AZTEC [41], técnica de desvio de pontos (TP) [42], *co-ordinate reduction time encoding system* (CORTES) [43] e *Fan* [44].

A Figura 4.2 representa a comparação entre os três métodos citados anteriormente (TP, AZTEC e CORTES). Nessa ilustração, observa-se o desempenho da compressão de cada uma dessas metodologias, de maneira que o método TP apresenta descontinuidade no complexo *QRS*, o algoritmo AZTEC mostra distorções significantes e o CORTES resulta em menor distorção.

## 4.3 Compressão Paramétrica de ECG

Os métodos de compressão paramétricos utilizam algoritmos que extraem características ou parâmetros dominantes do sinal de ECG [45]. Exemplos incluem, dentre outros, redes

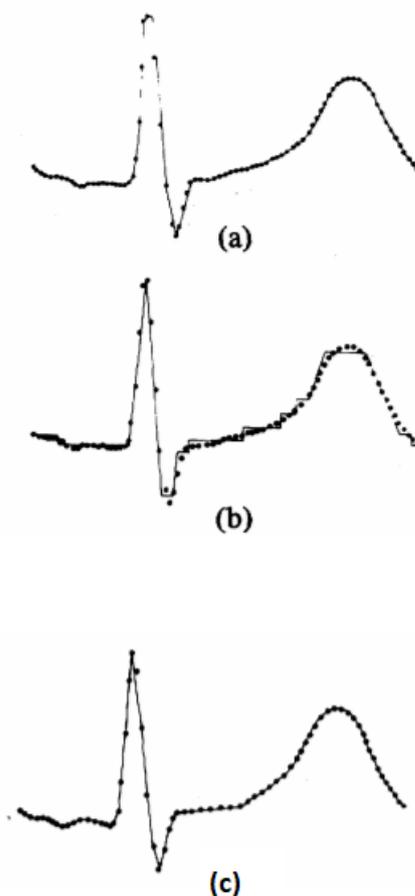


Figura 4.2: Comparação entre os métodos: (a) TP, (b) AZTEC e (c) CORTES  
FONTE: adaptado de ABENSTEIN, (1982) [43].

neurais [46], método de extração de picos [47] e algoritmos de quantização vetorial [48].

Por exemplo, o método de compressão de ECG utilizando redes neurais [49] usa, como base, neurônios artificiais distribuídos em camadas, com uma de entrada e outra de saída, além de camadas ocultas entre estas. No caso do trabalho de Zhang *et al.* (2017) [49], o qual utilizou três camadas, cada uma com certa quantidade de neurônios, que por sua vez possuem um peso atrelado. Cada peso é determinado por um algoritmo chamado de *backpropagation*, que é responsável por ajustar polaridades e pesos. Zhang *et al.* utilizaram 70 elementos na camada inicial, por onde sinais de ECG são inseridos na rede neural, assim como 70 na camada de saída. A camada oculta do meio é responsável por realizar compressão, por meio da técnica de multi-objetivos [49].

## 4.4 Compressão de ECG utilizando transformadas

Técnicas baseadas em transformadas são muito utilizadas em sistemas eletrocardiógrafos [41]. Essas arquiteturas geralmente necessitam de detecção de períodos de ECG e geralmente há um pré-processamento no sinal de entrada, além da utilização de transformações ortogonais, de onde posteriormente serão descartados dados desnecessários para a representação de sinais. Na reconstrução, a transformada inversa é calculada e o sinal original é recuperado, com algum erro tolerável [41].

Muitas técnicas de transformação vêm sendo utilizadas em pesquisas [45] e, dentre elas, é possível destacar a DCT.

### 4.4.1 A transformada discreta dos cossenos na compressão de sinais de ECG

A transformada discreta dos cossenos, cujo acrônimo é DCT, vem sendo amplamente utilizada, devido à sua capacidade de compactação de energia. Existem diferentes algoritmos com base nessa arquitetura, para compressão de sinais ECG, que apresentam bom desempenho, desde que aliados a alguma técnica de quantização e métodos de codificação, visando explorar particularidades da DCT [50].

A DCT representa um sinal como uma soma de variáveis de frequência e magnitude e é, provavelmente, a principal técnica de compressão de dados utilizada na atualidade, sendo empregada para a codificação de diversos tipos de sinais, como imagens, vídeo, áudio e ECG. A DCT tem como característica principal alto poder de compactação de energia, além de apresentar alta correlação. De fato, um sinal comprimido utilizando DCT, pode ser restaurado com um pequeno número de coeficientes [45].

A DCT  $v = [v_0 v_1 \dots v_{N-1}]^T$  de um vetor  $x$  é calculada através de

$$v_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n, \quad (4.1)$$

e

$$v_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \frac{\cos(2n+1)k\pi}{2N}, \quad k = [1, 2, \dots, (N-1)], \quad (4.2)$$

onde  $v_k$  é um coeficiente da DCT e  $x_n$  é o vetor que representa o sinal de ECG. A DCT inversa (*inverse discrete cosine transform* - IDCT) é dada por

$$x_k = \frac{1}{\sqrt{N}}v_0 + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} v_k \frac{\cos(2n+1)k\pi}{2N}, \quad n = [0, 1, 2, \dots, (N-1)]. \quad (4.3)$$

Assim como existem algoritmos rápidos para o cálculo da transformada de Fourier, também existe um modo rápido de calcular a DCT, através da aplicação da técnica de desdobramentos sucessivos. Uma DCT de um sinal expressa uma sequência finita de dados, em termos de uma soma de funções cosseno oscilando em diferentes frequências. No caso de sinais de ECG, que apresentam um espectro concentrado com componentes em baixas frequências (sinal passa-baixas), a DCT é a abordagem ideal, pois ela descarta as componentes de alta frequência que pouco contribuem com o sinal e permite compressão com resultados satisfatórios, conforme o trabalho de Sharma e Wadhvani (2011) [51].

A Figura 4.3 ilustra a metodologia denominada *Joint Photographic Experts Group* (JPEG), a qual emprega a DCT em sua estrutura, seguida por uma etapa de quantização e um codificador de entropia que no JPEG é o codificador *Huffman*. O processo de reconstrução é implementado com os respectivos processos inversos do decodificador de entropia, quantizador e transformada DCT inversa (IDCT).

## 4.5 Uma nova classe de compressão de ECG baseada na compressão distribuída

Neste capítulo, as principais arquiteturas utilizadas na compressão de ECG e encontradas na literatura foram apresentadas. Tais metodologias apresentam diferentes resultados de compressão, sendo as baseadas em transformadas as com melhores desempenhos. Contudo, tais métodos precisam realizar muitos cálculos no codificador, assim como também os métodos paramétricos e diretos, o que não é ideal para dispositivos embarcados que carecem de bateria, assim sendo, uma nova classe surgiu baseada em fundamentos de codificação distribuída, que possuem potencial de implementação de codificadores mais simples.

A codificação distribuída foi fundamentada na década de 70, por Slepian e Wolf [18], para compressão sem perdas, o que foi posteriormente estendido por Wyner e Ziv [15], para compressão com perdas. A arquitetura de codificação distribuída é baseada no princípio da

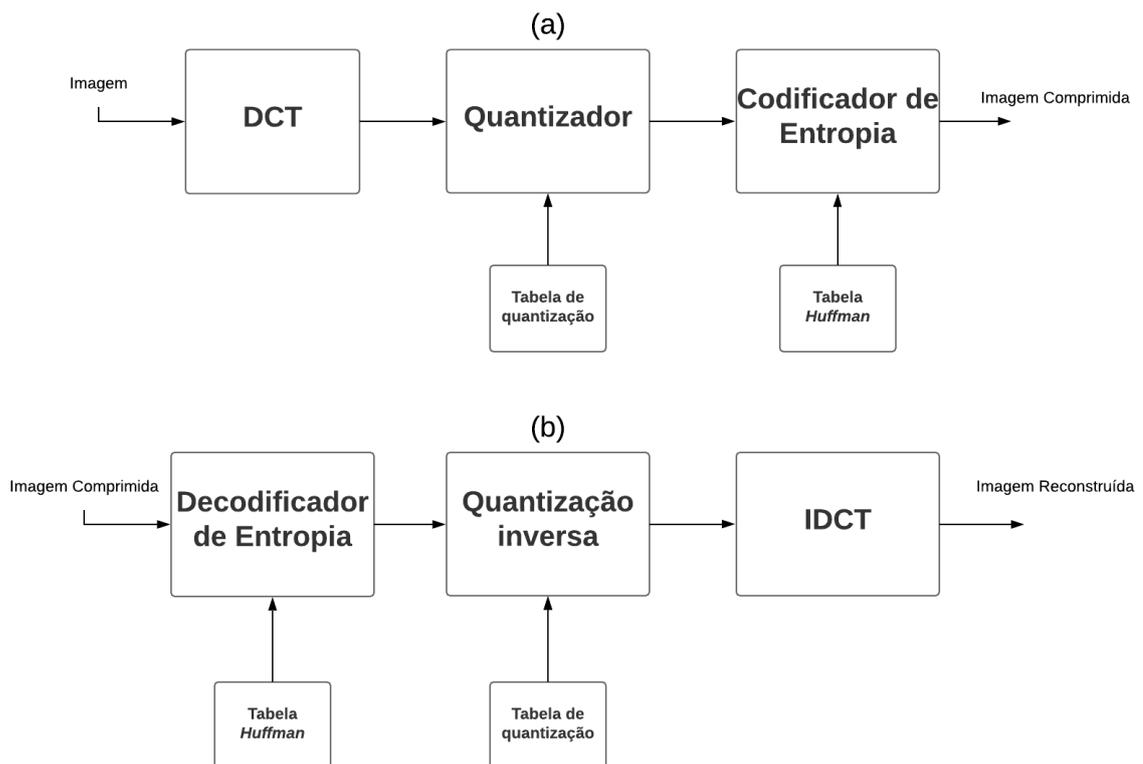


Figura 4.3: Arquitetura da metodologia de compressão JPEG, (a) codificador, (d) decodificador  
 FONTE: adaptado de Kurniawan, Purboyo e Prasasti (2017) [52].

codificação de fontes aleatórias, de forma independente, e decodificação conjunta. Essas ideias de Wyner e Ziv foram demonstradas apenas na teoria, devido a limitação de sistemas computacionais da época, porém, esta técnica de codificação tem sido abordada em pesquisas mais atuais.

Wu, Tsai e Chang [14] são pioneiros na compressão de ECG utilizando fundamentos de DSC, apresentaram no ano de 2014 uma arquitetura DSC fundamentada na teoria de Slepian e Wolf e tem como integrante de sua metodologia a utilização de códigos turbo. A etapa de codificação é antecedida de pré-processamento do sinal de ECG que busca deixar o sinal na forma bidimensional, ou seja, todos períodos são identificados e normalizados, sendo posteriormente colocados em uma linha cada, formando uma matriz. Além disso, emprega técnica de quantização vetorial baseada em dicionário. O trabalho de Wu, Tsai e Chang mostrou que é possível a codificação distribuída de ECG, abrindo caminho para novas pesquisas nesta nova ramificação de algoritmos de compressão de ECG.

Em relação a compressão de sinais de ECG utilizando DSC, ainda tem muito a se explorar. Pensando nisso este trabalho busca utilizar a mesma técnica de pré-processamento de Wu,

Tsai e Chang, com adicionais de equalização DC e ordenação por similaridade de segmentos de ECG, o que permite mais similaridade entre segmentos adjacentes, e desta forma, melhora o desempenho de compressão. Além disso, a técnica de quantização uniforme é utilizada, em vez de vetorial, eliminando assim a necessidade de treinamento e geração de dicionário.

As técnicas apresentadas nas seções anteriores são normalmente utilizadas de forma que os respectivos codificadores explorem redundâncias e estatísticas das fontes, de forma intensiva, e resultem em representações compactas, que apenas são remapeadas em informações semelhantes ou idênticas às originais, dependendo da abordagem geral (com ou sem perdas). Entretanto, o presente trabalho propõe uma metodologia que apenas reduz a resolução de representações, como se erros tivessem sido introduzidos no sistema, deixando para os decodificadores a tarefa de explorar estatísticas e redundâncias, ou seja, são tarefas que mais exigem trabalho computacional e realmente caracterizam uma nova classe de codificadores.

Vale a pena mencionar que, apesar das técnicas apresentadas nas seções anteriores estarem inseridas em arquiteturas tradicionais, elas também podem ser utilizadas em esquemas baseados em compressão distribuída, desde que apenas utilizadas para facilitar ou otimizar modificações nos sinais originais, excluindo-se etapas complexas de codificação. Um exemplo é a DCT inteira [53], que foi incorporada à metodologia proposta e permitiu melhores resultados.

Finalmente, a principal modificação proposta não reside exatamente em técnicas de transformação de sinais ou codificação de dados, mas sim o local onde as estatísticas dos sinais são exploradas, o que foi possível com a utilização de códigos turbo [54].

# Capítulo 5

## Codificação distribuída de sinais

Neste capítulo, a técnica de codificação de fontes distribuídas, que proporciona codificação separada e decodificação conjunta, de fontes de sinais, será apresentada. Em especial, o procedimento de codificação distribuída será descrito, com ênfase em codificação distribuída sem perdas, codificação distribuída com perdas e, por fim, códigos turbo.

### 5.1 A codificação distribuída

A metodologia DSC é um problema importante na teoria da informação. A sua estrutura baseia-se em compactação de fontes de informações correlacionadas, que podem ou não se comunicar. Ao modelar a correlação entre fontes, no lado do decodificador e juntamente com algum código de canal (normalmente, códigos corretores de erros são utilizados), o DSC é capaz de mudar a complexidade computacional do codificador para o decodificador. Portanto, a DSC fornece estruturas apropriadas para aplicações com emissores que apresentam restrições de complexidade, como, por exemplo, dispositivos embarcados e, especificamente, o *Holter* de ECG. Uma das principais propriedades da codificação de fontes distribuídas é que a carga computacional em codificadores é deslocada para um decodificador comum. É possível desmembrar a teoria DSC em duas partes: codificação sem ou com perdas.

### 5.2 A codificação distribuída sem perdas

Na compressão sem perdas tradicional, conforme mencionado anteriormente, a informação restaurada na decodificação é exatamente igual a original. Existem várias técnicas de

compressão sem perdas, nas quais codificadores exploram sinais originais e, dentre elas, é possível destacar: codificação *Run-length* [55], codificação *Huffman* [56], codificação aritmética [57] e codificação de *Lempel Ziv* [58]. Entretanto, recentemente, uma ramificação completamente nova surgiu, a qual é baseada em conceitos de Slepian e Wolf [18] e permite exploração de redundâncias em decodificadores, através de codificação separada e decodificação conjunta de fontes.

Conceitos da teoria de codificação distribuída de Slepian e Wolf tem base nas descobertas de Claude E. Shannon [59], que idealizou a definição de auto-informação média conhecida como *entropia*, a qual, matematicamente, pode ser expressa da seguinte forma:  $X$  é uma fonte com alfabeto  $X = [x_1, x_2]$  e  $p(X) = [p_1, p_2]$  é a probabilidade de ocorrência de  $x_1$  e  $x_2$ . Como consequência, a auto-informação de Shannon associada a cada um desses símbolos é dado por

$$h_1 = \log_2\left(\frac{1}{p_1}\right) \quad (5.1)$$

e

$$h_2 = \log_2\left(\frac{1}{p_2}\right). \quad (5.2)$$

Na prática, geralmente não se está preocupado em saber a imprevisibilidade de um valor de uma variável aleatória, mas sim em conhecer todos os valores possíveis dessa variável aleatória  $X$ . Logo, Shannon definiu entropia como

$$H(X) = p_1 h_1 + p_2 h_2 = p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right) = \sum_{i=1}^2 p_i \log_2\left(\frac{1}{p_i}\right), \quad (5.3)$$

que é a informação de Shannon média. A entropia de uma fonte estabelece a quantidade mínima de *bits*, por símbolo, para representação de uma determinada fonte de símbolos binária, sem saber qual tipo de codificação foi usada. De fato, tal abordagem não é construtiva, pois apenas mostra o que é possível, sem dizer exatamente como isso poderia ser feito, e, devido a isso, algoritmos de compressão são sempre desenvolvidos buscando-se aproximação desse limiar de entropia de  $X$  [59]. A entropia média para os símbolos  $x_1$  e  $x_2$  é definida na equação (5.3). Caso  $X$  tenha  $m$  símbolos, essa expressão pode ser definida como

$$H(X) = - \sum_{i=0}^m p_i \log_2(p_i). \quad (5.4)$$

Ainda pelo trabalho de Shannon, a entropia conjunta de duas fontes distintas, onde  $H(X, Y)$  representa a entropia conjunta das fontes  $X$  e  $Y$ , foi definida por Shannon como

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2(p(x, y)). \quad (5.5)$$

Um resultado muito interessante é que a entropia conjunta pode ser escrita em função da entropia condicional, através de

$$H(X, Y) = H(X) + H(Y|X), \quad (5.6)$$

onde a entropia condicional  $H(Y|X)$  é calculada com

$$H(Y|X) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2(p(y|x)). \quad (5.7)$$

Compressores tradicionais, conforme mostrado na Figura 5.1, codificam e decodificam separadamente duas fontes correlacionadas  $X$  e  $Y$ . Entretanto, Slepian e Wolf definiram que, teoricamente, existe uma região de taxas maiores ou iguais à entropia conjunta  $H(X, Y)$ , quando a codificação de  $X$  e  $Y$  é realizada de forma separada [18] e utilizando-se informação auxiliar no decodificador, conforme mostrado na Figura 5.2, o que pode ser expresso por

$$R_X + R_Y \geq H(X, Y). \quad (5.8)$$

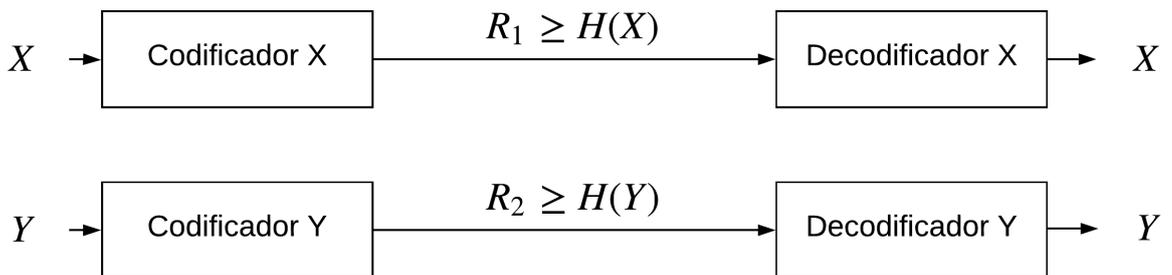


Figura 5.1: Ilustração de esquema de codificação tradicional.

A Figura 5.3 mostra as regiões atingíveis e não atingíveis com a codificação de fontes separadas, ao passo que a equação (5.8) deixa claro que esta codificação alcança taxas maiores ou iguais à codificação conjunta  $H(X, Y)$ , desde que haja uma pequena probabilidade de erro residual na recuperação de  $X$  e  $Y$  [18]. A exploração de informação auxiliar pode acontecer

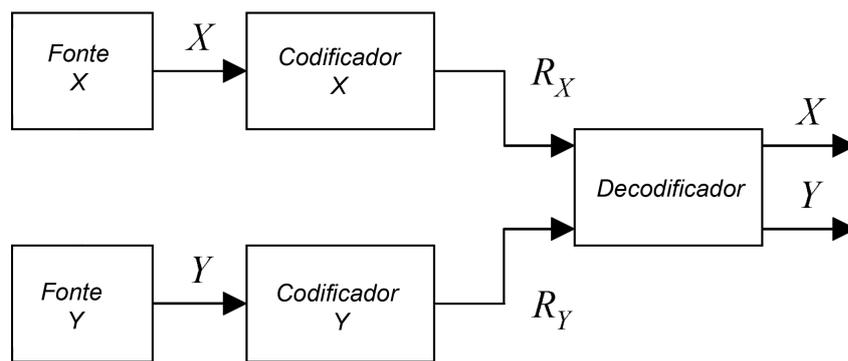


Figura 5.2: Codificação distribuída de duas fontes (*i.i.d*)  $X$  e  $Y$ .  
 FONTE: adaptado de Girod *et al.* (2005) [60]

tanto no codificador quanto no decodificador, ou totalmente no decodificador. Portanto, conforme Slepian e Wolf,  $X$  pode ser codificado usando a entropia condicional  $H(X|Y)$ , de maneira que se tenha conhecimento prévio de  $Y$  e, posteriormente, decodificado com a informação auxiliar  $Y$ , presente no decodificador [18].

É possível entender então, pela teoria de Slepian e Wolf, que a codificação de duas fontes  $X$  e  $Y$  pode ser realizada por dois codificadores, um que codifica a fonte  $Y$ , a uma taxa não menor que  $H(Y)$ , e outro que codifica  $X$ , a uma taxa não menor que  $H(X|Y)$ , o que possibilita a codificação distribuída e a transferência da exploração estatística para o decodificador. Os pontos A e B na Figura 5.3 identificam a compressão de duas fontes  $X$  e  $Y$  com codificação separada e decodificação conjunta, o que é conhecido como abordagem assimétrica. O ponto C representa a entropia máxima para codificação e decodificação separada. Portanto, conforme a Figura 5.3, percebe-se que a decodificação conjunta alcança taxas melhores que a decodificação separada.

A Figura 5.4 descreve um exemplo de codificação distribuída sem perdas abordado por Pradhan, Sandep e Ranchandram [19]. Supondo-se duas fontes  $X$  e  $Y$  equiprováveis e com largura de 3 *bits*, de maneira que a diferença entre ambas seja de, no máximo, 1 *bit*, se  $X = 010$ , então  $Y$  pode ser (010,011,000,110). O detalhe mais importante na codificação separada está no conhecimento da estatística conjunta de  $X$  e  $Y$ , que nesse caso é a diferença entre  $X$  e  $Y$ , ou seja,  $D_{Hamming}(X,Y) \leq 1$ , permitindo assim a codificação de  $X = x_1, x_2, x_3, \dots, x_n$  sem conhecimento de  $Y$ . O espaço amostral de  $X$  é composto por valores iguais a (000,001,010,011,100,101,110,111), que por sua vez é subdividido em quatro conjuntos distintos (000,111),(001,110),(010,101) e (011,100), de forma que a distância entre os *bits*

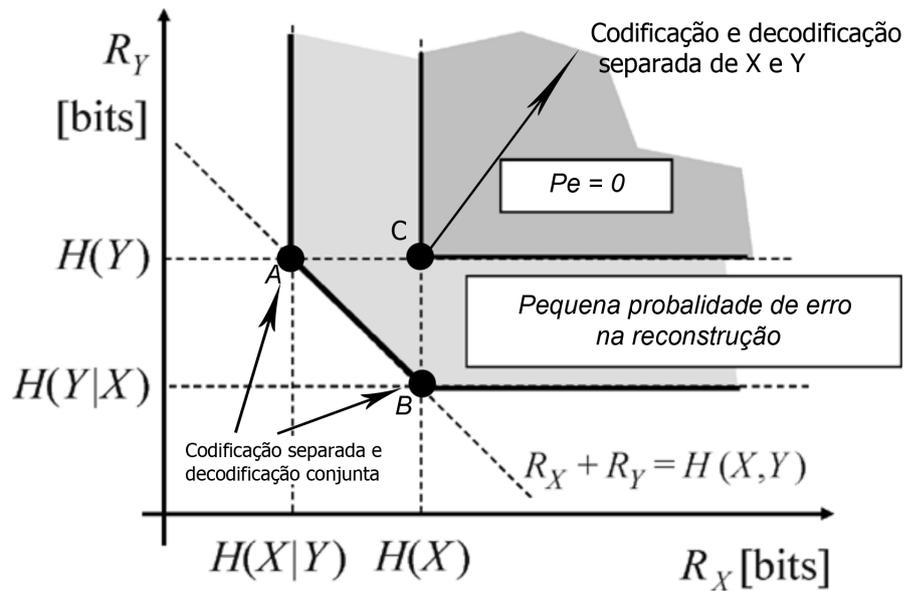


Figura 5.3: Região de taxas atingíveis da codificação de Slepian Wolf.  
 FONTE: adaptado de Slepian Wolf (1973) [18].

de cada conjunto seja igual a 3. Cada um deles é classificado com um índice, que neste exemplo pode ser (00,01,10,11), pois foram criados 4 conjuntos e a fonte  $Y$  envia o valor bruto de 3 bits. Em resumo,  $X$  envia o índice referente ao seu conjunto e  $Y$  envia os dados brutos que ajudam na sua decodificação (utilizar-se-á aquele mais próximo de  $Y$ , dentre os dois do conjunto). Com isso, a entropia conjunta é igual a  $H(X, Y) = H(Y) + H(X | Y) = 3 + 2 = 5$  e a codificação conjunta utiliza 5 bits, ao invés de 6, o que leva ao atingimento da região do ponto A, na Figura 5.3.

Índices	{00}	{01}	{10}	{11}	
Conjuntos	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	
	conjunto 0	conjunto 1	conjunto 2	conjunto 3	
$Y = \{100, 111, 100, 000, 101\}$					
$X = \{101, 011, 000, 001, 111\}$					
Codificando X e Y temos: $\{(100)10\}$ $\{(111)11\}$ $\{(100)00\}$ $\{(000)01\}$ $\{(101)00\}$					
decodificando:	$\overbrace{\{010, 101\}}^{10}$	$\overbrace{\{100, 011\}}^{11}$	$\overbrace{\{000, 111\}}^{00}$	$\overbrace{\{001, 110\}}^{01}$	$\overbrace{\{000, 111\}}^{00}$
Y =	100	111	100	000	101
X =	101	011	000	001	111

Figura 5.4: Exemplo de codificação distribuída sem perdas.  
 FONTE: Pradham e Ramchandran (2003) [61].

O exemplo da Figura 5.4 pode ser visto como um código corretor de erros que utiliza a informação auxiliar  $Y$  para reconstruir  $X$ . Os índices criados para cada subconjunto de  $X$  são também chamados de síndromes e podem ser modelados matematicamente pela expressão  $s^T = Hc^T$ , onde  $H$  é a matriz de checagem de paridade,  $c^T$  são *bits* do conjunto de entrada,  $T$  denota a transposta e  $s$  e  $c$  são vetores. Portanto, cada linha de um conjunto tem uma única síndrome [61].

Sendo assim, o problema de codificação distribuída pode ser resolvido utilizando-se códigos corretores de erros. A ideia principal consiste na formação de conjuntos chamados de *cosets*, a partir de um espaço amostral, de maneira que haja correlação entre  $X$  e  $Y$ , como abordado na Figura 5.4.

### 5.3 A codificação distribuída com perdas

Três anos após a publicação de Slepian e Wolf, Wyner e Ziv generalizaram o seu teorema para codificação com perdas. Considerando duas fontes aleatórias  $X$  e  $Y$  *i.i.d.*, que respectivamente representam a fonte e a informação auxiliar, com  $X$  sendo codificado sem acesso à informação auxiliar  $Y$ , conforme mostrado na Figura 5.5, um decodificador recupera  $X$  utilizando a informação prévia de  $Y$ , a partir de uma curva taxa-distorção  $D = E[d(X, \hat{X})]$ , sendo que a função de Wyner-Ziv  $R_{X|Y}^{WZ}(D)$  será o limite inferior de taxa para se obter uma distorção máxima  $D$ . A função taxa-distorção  $R_{X|Y}(D)$  é a taxa requerida quando a informação auxiliar está presente no codificador [15].

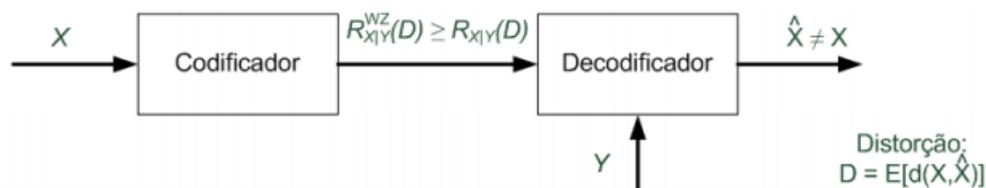


Figura 5.5: Esquema de codificação Wyner-Ziv.  
FONTE: Adaptado de Wyner-Ziv (1976) [15].

Wyner e Ziv chegaram à conclusão de que uma perda de taxa

$$R_{X|Y}^{WZ}(D) - R_{X|Y} \geq 0, \quad (5.9)$$

acontece quando o codificador não tem acesso à informação auxiliar. Contudo, eles também provaram que  $R_{X|Y}^{WZ}(D) - R_{X|Y} = 0$ , no caso de fontes gaussianas sem memória [15]. Em geral, um codificador Wyner-Ziv pode ser constituído com um quantizador seguido de um codificador de Slepian-Wolf, conforme a Figura 5.6, que mostra esse esquema de codificação (enCoder - DECoder - CODEC), onde o quantizador  $Q$  divide o espaço amostral do sinal, em células com índice  $q$ , e envia índices ao codificador de Slepian Wolf.

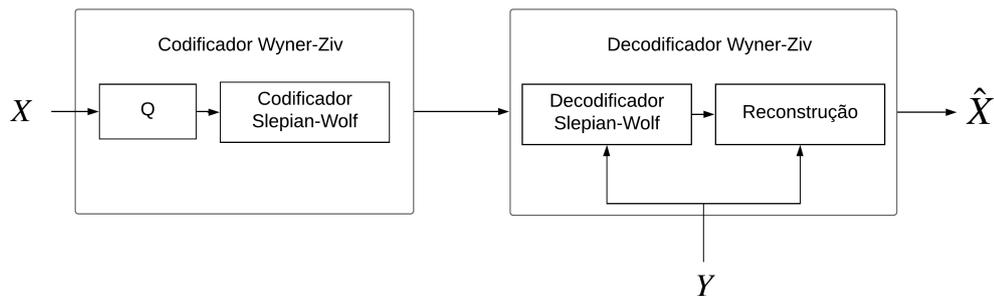


Figura 5.6: CODEC Wyner-Ziv.

FONTE: Adaptado de Aaron, Zhang e Girod (2003) [62]

Existem vários codificadores práticos Wyner-Ziv, sendo o apresentado por Girod *et al* [60], um exemplo interessante. Conhecido também como arquitetura de *Stanford*, que está mostrada na Figura 5.7, ela foi desenvolvida inicialmente para codificação de vídeo, utilizando somente um quantizador, um codificador de Slepian-Wolf e um decodificador baseado em códigos turbo [62–64], sendo posteriormente aperfeiçoada, com a inclusão da DCT [13].

Nos exemplos de Girod *et al.* [13, 60], com e sem transformada DCT, um conjunto de quadros de vídeos é dividido em duas partes: quadros Wyner-ziv e quadros-chaves. O primeiro é composto por quadros pares do conjunto total, ao passo que os quadros-chaves são os ímpares. Ambos são codificados com a técnica *intraframe*, porém, quadros Wyner-ziv são codificados com um codificador de Slepian-Wolf e decodificados com a ajuda de quadros-chave codificados com um compressor convencional, utilizado para produzir a informação lateral  $Y_k$ .

## 5.4 Códigos turbo

Como já foi comentado na seção anterior, códigos corretores de erros são candidatos à codificação de Slepian e Wolf, sendo os códigos turbo muito explorados para tal propósito.

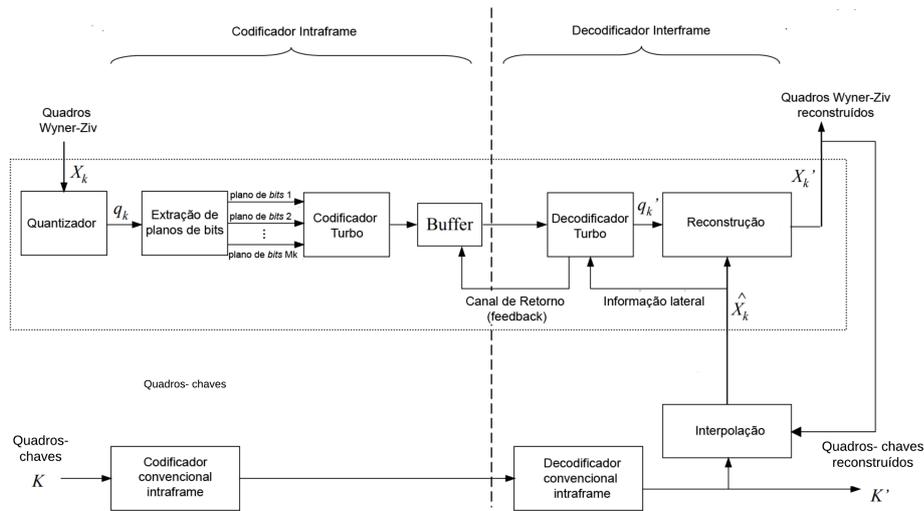


Figura 5.7: Arquitetura básica de codificação Wyner-Ziv .

FONTE: Adaptado de Aaron e Girod (2002) [65].

Nesta seção, a técnica de codificação turbo será abordada, assim como o seu funcionamento e as suas principais características.

### 5.4.1 Codificação Turbo

O codificador turbo de Berrou, Glaviex e Thitimajshima (Figura 5.8) [54], ilustrado na Figura 5.8 , é composto por dois codificadores concatenados com taxa binária de 1/2 cada, sendo a taxa total na saída do codificador igual a 1/3. O conjunto de bits sistemáticos " $u$ " é enviado aos dois codificadores, sendo que o segundo recebe uma versão entrelaçada. Em cada codificador, somas módulo-2 são realizadas e, dessa forma, saídas de *bits* de paridade  $x_{kp}^1$  e  $x_{kp}^2$  são geradas.

A operação de codificação convolucional pode ser descrita por um processo de Markov de estado finito [66], que por sua vez é representado por um diagrama de estados ou treliças, onde uma entrada  $u$  gera uma saída  $x_{kp}^{(1)}$ , fazendo assim o sistema transicionar de um estado  $s$  para outro  $s'$ .

A função do entrelaçador P mostrado na Figura 5.8 é descorrelacionar a entrada do codificador 2 em relação ao 1. Dessa forma, os *bits* que não conseguem ser corrigidos pelo primeiro decodificador tendem a ser corrigidos pelo segundo. Um entrelaçador de largura  $N$  é suficiente para descorrelacionar a entrada [54]. De fato, esse elemento é o que confere aleatoriedade ao

processo e o seu objetivo é atender à uma das descobertas de Shannon, que postulou sobre a superioridade de códigos com características aleatórias [67]. Os *bits* de paridade  $x_{kp}^{(1)}$  e  $x_{kp}^{(2)}$  são enviados ao decodificador, onde suas versões recebidas são denominadas  $y_{kp}^{(1)}$  e  $y_{kp}^{(2)}$ . Os *bits* sistemáticos  $x_{kl}$  não são enviados, pois o decodificador utiliza *bits* sistemáticos gerados pela informação lateral.

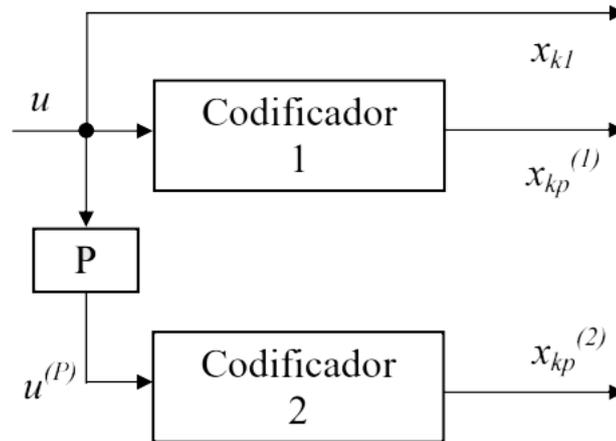


Figura 5.8: Codificador Turbo de taxa 1/3.

FONTE: Adaptado de Berrou, Glavieus e Thitimajshima (1993) [54].

### 5.4.2 Decodificação Turbo

A Figura 5.9 representa um decodificador turbo e é possível perceber que a sua arquitetura é semelhante ao codificador, pois há dois decodificadores em paralelo, além de operações direta e inversa de entrelaçamento, denominadas  $P$  e  $P^{-1}$ .

A última saída  $L(u_k|y)$  do decodificador mostrado na Figura 5.9 é denominada *log-likelihood ratio* (LLR), sendo definida pela probabilidade *a posteriori*

$$L(u_k|y) = \ln \frac{P(u_k = +1|y)}{P(u_k = -1|y)}. \quad (5.10)$$

A equação (5.10) utiliza probabilidades condicionais *a posteriori*, ou seja, as probabilidades, no instante  $k$ , do *bit* de entrada do codificador ter sido  $u_k = +1$  ou  $u_k = -1$ , dado que o decodificador tem conhecimento de toda sequência recebida. É importante ressaltar que *bits* com valores iguais a zero ( $u_k = 0$ ) são mapeados para serem iguais a  $u_k = -1$ . Esse algoritmo

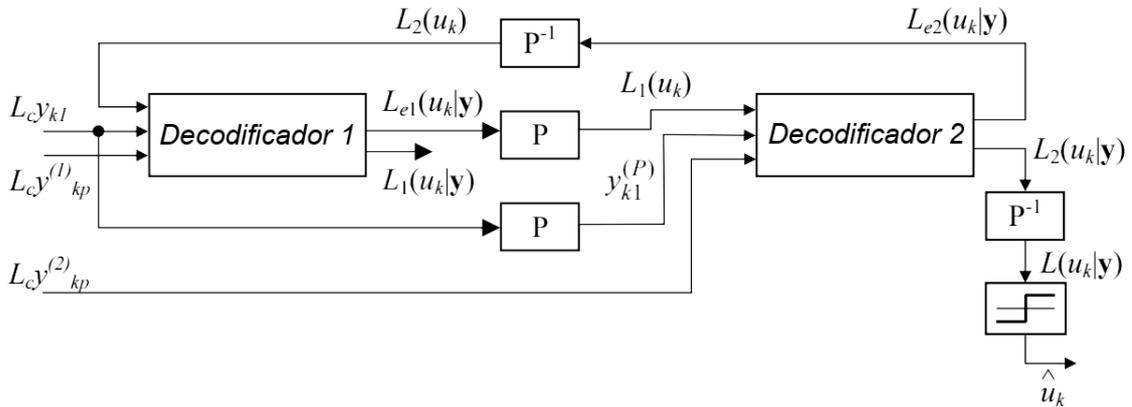


Figura 5.9: Decodificação turbo.

FONTE: Adaptado de Berrou, Glaviex e Thitimajshima (1993) [54].

é conhecido como *Máximo a Posteriori Probability* (MAP) [54], o qual realiza cálculos de probabilidades *a posteriori*, e no final destes cálculos, se  $L(u_k|y) > 0$ , estima-se que foi enviado o bit  $u_k = 1$ , caso contrário, estima-se  $u_k = -1$ . O algoritmo MAP calcula a LLR *a posteriori* (equação (5.10)) e indica o *bit* de saída do decodificador, onde o sinal representa o bit enviado e o seu valor absoluto informa a confiança no *bit* calculado [54].

Na decodificação iterativa, uma sequência de  $N$  *bits* é decodificada por vez e, dessa forma, torna-se conveniente trabalhar com treliças [68], conforme mostrado na Figura 5.10, pois mostram todos os estados e as saídas possíveis, conforme os *bits* de entrada. Por exemplo, considerando-se um codificador convolucional com taxa de  $1/2$ , ou seja, para cada *bit* enviado há dois na saída, com quatro estados 0,1,2,3, uma treliça que mostra dois instantes de tempo consecutivos  $(k-1, k)$ , onde uma linha tracejada representa que o bit na entrada foi (+1), uma contínua que este foi (-1) e o estado atual é  $S_k = s$ , o estado anterior é  $S_{k-1} = s'$  e o símbolo recebido no decodificador é  $y_k$ . Antes e depois deste símbolo já foram recebidos, respectivamente,  $(N-k)$  e  $(N+k)$  símbolos. Portanto, a sequência completa recebida pelo decodificador pode ser dividida em três partes, representando o passado, presente e futuro [68]. Com base nisso, a LLR *a posteriori* (equação (5.10)) pode ser dividida em três subsequências, conforme a equação

$$L(u_k|y) = \ln \frac{P(u_k = +1|y)}{P(u_k = -1|y)} = \ln \frac{\sum_{R_1=0}^n P(s', s, y)}{\sum_{R_0=0}^n P(s', s, y)} = \ln \frac{\sum_{R_1=0}^n \alpha_{k-1}(s') \gamma(s', s) \beta_k(s)}{\sum_{R_0=0}^n \alpha_{k-1}(s') \gamma(s', s) \beta_k(s)}, \quad (5.11)$$

onde a probabilidade conjunta  $P(s', s, y)$  pode ser calculada como o produto de três probabilidades.

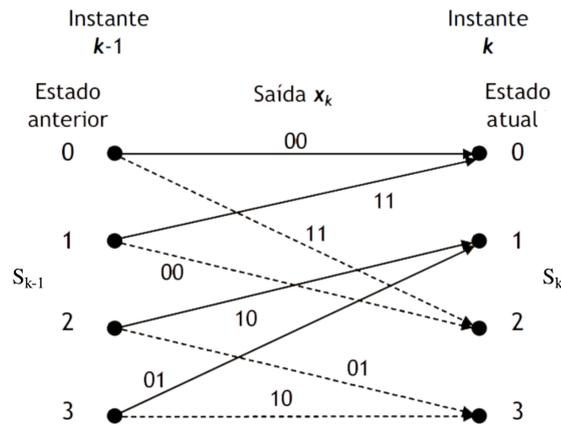


Figura 5.10: Exemplo de uma treliça de codificação turbo.  
 FONTE: Adaptado de Abrantes (2010) [68].

No exemplo de treliça da Figura 5.10, a estimativa do *bit* enviado, conforme algoritmo MAP, é realizada com

$$L(u_k|y) = \ln \frac{\sum_{R_1=0}^3 P(s', s, y)}{\sum_{R_0=0}^3 P(s', s, y)} = \ln \frac{p(0, 2, y) + p(1, 2, y) + p(2, 3, y) + p(3, 3, y)}{p(0, 0, y) + p(1, 0, y) + p(2, 1, y) + p(3, 1, y)}, \quad (5.12)$$

onde o numerador representa as possíveis transições de estado entre os instantes  $k-1$  e  $k$ , caso o *bit* na entrada no codificador seja 1, e o denominador representa esse mesmo cálculo para o *bit* zero.

No instante  $k$ , as probabilidades  $\alpha$ ,  $\gamma$  e  $\beta$  estão associadas ao passado, ao presente e ao futuro da sequência  $y$ , respectivamente. A parcela  $\gamma(s', s) = P(y_k, s|s')$  é a probabilidade condicional de, no instante  $k$ , o símbolo recebido ser  $y_k$  e o estado atual ser  $S_k = s$ , sabendo-se que o estado de origem foi  $S_{k-1} = s'$ . Portanto,  $\gamma$  é calculado pelo produto das probabilidades

$$\gamma(s', s) = P(y_k|y_k)P(u_k), \quad (5.13)$$

onde a parcela  $P(y_k|u_k)$  representa a probabilidade *a posteriori* e  $P(u_k)$  a probabilidade *a priori*. Em outras palavras, a primeira parcela representa a probabilidade condicional de  $y$  ser recebido dado o símbolo  $x_k$ , ao passo que a parcela  $P(u_k)$  representa o conhecimento prévio da probabilidade de um *bit* ser 0 ou 1.

As probabilidades  $\alpha$  e  $\beta$  são recursivamente calculadas, sendo a primeira computada do início da treliça, ou seja,  $k-1$ , até o final desta, ou seja,  $k$ , a cada instante, e a segunda obtida do último instante, ou seja,  $k$  até o primeiro, ou seja,  $k-1$ , conforme

$$\alpha_k(s) = \sum_{s'=0}^n \gamma(s', s) \alpha_{k-1}(s) \quad \text{sendo} \quad \alpha_0(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0 \end{cases} \quad (5.14)$$

e

$$\beta_{k-1}(s') = \sum_{s=0}^n \gamma(s', s) \beta_k(s) \quad \text{sendo} \quad \beta_N(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0. \end{cases} \quad (5.15)$$

### 5.4.3 O algoritmo MAP e a decodificação iterativa

Como visto anteriormente, códigos turbo funcionam de forma iterativa. Se houver um codificador convolucional sistemático de taxa 1/3, como mostrado na Figura 5.8, onde um primeiro *bit* codificado  $x_{k1}$  é igual ao bit de informação  $u_k$  e os demais ( $x_{kp}^{(1)}$   $x_{kp}^{(2)}$ ) são chamados de *bits* de paridade [68], a razão LLR *a posteriori*  $L(u_k|y)$  pode ser decomposta conforme o teorema de Bayes, de acordo com

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)}, \quad (5.16)$$

que aplicado à equação (5.10) resulta em

$$L(u_k|y) = \ln \left( \frac{p(y|u_k=+1)p(u_k=+1)}{p(y|u_k=-1)p(u_k=-1)} \right) = \ln \left( \frac{p(y|u_k=+1)}{p(y|u_k=-1)} \right) + \ln \left( \frac{p(u_k=+1)}{p(u_k=-1)} \right). \quad (5.17)$$

A parcela  $L(u_k|y)$  pode ser decomposta em três partes:  $\alpha$ ,  $\beta$  e  $\gamma$ , conforme demonstrado na equação (5.10). Contudo,  $\alpha$  e  $\beta$  podem sofrer instabilidade numérica e precisam ser normalizadas, conforme [68]

$$\alpha'_k(s) = \frac{\sum_{s'=0}^n \alpha_{k-1}(s') \gamma_k(s', s)}{\sum_{s=0}^n \alpha'_k(s)} \quad (5.18)$$

e

$$\beta'_{k-1}(s') = \frac{\sum_{s=0}^n \beta_k(s) \gamma_k(s', s)}{\sum_{s'=0}^n \beta'_{k-1}(s')}, \quad (5.19)$$

respectivamente.

A probabilidade *a priori*  $P(u_k)$  é definida por

$$P(u_k = \pm 1) = e^{\frac{u_k^2 \text{apriori}}{L}}, \quad (5.20)$$

ao passo que  $L^{apriori}$  é dada por

$$L^{apriori} = \frac{P(u_k = +1)}{P(u_k = -1)}. \quad (5.21)$$

No caso especial de ruído gaussiano,  $\gamma$  pode ser calculado por

$$\gamma_k(s', s) = D_k e^{u_k L(u_k)/2} e^{\left(\frac{L_c}{2} \sum_{l=1}^n y_{kl} x_{kl}\right)}, \quad (5.22)$$

onde  $D_k$  representa um fator que estará presente no numerador e no denominador da equação (5.11), desaparecendo no cálculo da LLR  $L(u_k|y)$ . A parcela  $L_c$  representa a medida de fiabilidade do canal e, no caso da modulação BPSK ser utilizada, é dada por

$$L_c = \frac{E_c}{N_0/2}, \quad (5.23)$$

onde  $N_0/2$  representa a densidade espectral do ruído branco e  $E_c = rE_b$  a energia por *bit*, sendo  $r$  a taxa do canal.

Supondo-se que o primeiro *bit*  $x_{k1}$  codificado seja o sistemático  $u_k$ , então a equação (5.22) pode ser desmembrada em

$$\gamma_k(s', s) = \frac{D_k e^{(u_k)^2 [L(u_k) + L_c y_{k1}]} x p^{\frac{L_c}{2} (\sum_{l=2}^n y_{kl} x_{kl})}}{e}, \quad (5.24)$$

onde o cálculo da primeira exponencial é realizado com os *bits* sistemáticos e a probabilidade  $L(u_k)$ . A segunda exponencial utiliza os *bits* de paridade  $y_{kl}$  e pode ser representada por uma nova variável, conforme

$$v_k = e^{\frac{L_c}{2} \sum_{l=2}^n x_{kl} y_{kl}} \quad (5.25)$$

Introduzindo-se  $v_k$  na equação (5.11), obtêm-se

$$L(u_k|y) = \ln \frac{\sum_{R_1=0}^n D_k e^{\frac{u_k}{2} [L(u_k) + L_c y_{k1}]} \alpha_{k-1}(s') v(s', s) \beta_k(s)}{\sum_{R_0=0}^n D_k e^{\frac{u_k}{2} [L(u_k) + L_c y_{k1}]} \alpha_{k-1}(s') v(s', s) \beta_k(s)}, \quad (5.26)$$

a qual pode ser reduzida a

$$L(u_k|y) = L(u_k) + L_c y_{k1} + L_e(u_k), \quad (5.27)$$

sendo que a parcela  $L(u_k)$  refere-se à informação *a priori*,  $L_c y_{k1}$  é calculada usando *bits* sistemáticos,  $L_e(u_k)$  representa a informação extrínseca calculada pela multiplicação  $\alpha v \beta$ , ou

seja, está relacionada aos *bits* de paridade, e  $L_c$  representa a medida de fiabilidade do canal. Como visto na equação (5.26),  $\gamma$  foi desmembrada em duas partes: uma primeira  $L_{c|y_{k1}}$ , onde *bits* sistemáticos são usados, e outra  $L_e(u_k)$ , onde os de paridade são empregados.

A partir desse momento, é possível descrever o funcionamento da decodificação iterativa [68], que procede da seguinte forma:

- Na primeira iteração,  $L(u_k)$  é chamado de informação *a priori*, ou seja, uma prévia da probabilidade dos *bits* enviados, a qual é considerada nula em decodificadores convencionais, e o decodificador 1 da Figura 5.9 calcula a informação extrínseca  $L_{e1}(u_k|y)$ , através da equação (5.27) e utilizando seus *bits* sistemáticos e de paridade do codificador 1;
- Depois do entrelaçamento, a informação  $L_{e1}(u_k|y)$  tornar-se  $L_1(u_k)$ , que é uma estimativa enviada ao decodificador 2, e os *bits* sistemáticos e de paridade do segundo codificador são utilizados para o cálculo de  $L_{e2}(u_k|y)$ , que é entregue ao decodificador 1 como  $L_2(u_k)$ , ou seja, uma estimativa mais confiável. A cada iteração, a estimativa calculada torna-se mais confiável e, depois de  $n$  interações, a saída do segundo decodificador uma decisão é tomada: se  $L(u_k|y) > 0$  estima-se que foi enviado o bit  $u_k = 1$ ; caso contrário,  $u_k = -1$ .

# Capítulo 6

## A compressão distribuída de ECG

Nos capítulos anteriores, todos os referenciais teóricos utilizados nessa dissertação foram revisados. Com base nisso, esse capítulo apresentará a metodologia para compressão de ECG utilizando técnicas de codificação distribuída, em conjunto com outros métodos, com a finalidade de comprimir sinais de eletrocardiograma.

### 6.1 Introdução

Nesse capítulo, duas metodologias de compressão serão apresentadas, sendo a primeira executada no domínio da amostra e a segunda no domínio da transformada. Ambas são implementadas inicialmente com etapa de pré-processamento, que provê maior similaridade entre períodos adjacentes de ECG. A seguir, os períodos resultantes são segmentados em dois grupos, conforme fundamentos da teoria de Slepian e Wolf, onde o primeiro, que compreende segmentos Wyner-Ziv (Wyner-Ziv *segments* - WZSs), é composto por elementos quantizados, codificados e, quando recebidos, decodificados com auxílio de informação lateral (*Side Information* - SI), gerada pela interpolação de segmentos do segundo grupo, denominados de segmentos-chaves (*key segments* - KSs). A codificação é realizada através de códigos turbo, os quais geram *bits* de paridade a partir de WZSs originais, que são enviados como a versão compacta desta informação, ou seja, correspondem aos dados comprimidos. Na recepção, KSs geram versões aproximadas de WZSs, através de interpolação, que são utilizadas como dados corrompidos a serem corrigidos com auxílio de *bits* de paridade fornecidos por WZSs após codificação. Por fim, os segmentos de ECG decodificados são reagrupados conforme suas ordens originais e o

processo inverso do pré-processamento é realizado, onde sinais de ECG são reconstruídos.

No presente momento, a maioria dos métodos de compressão de ECG encontrados na literatura utilizam a abordagem tradicional [38, 69–71]. Entretanto, recentemente surgiu uma nova classe de compressores de ECG, que é baseada em compressão distribuída e utiliza princípios postulados nos trabalhos de Slepian e Wolf e também Wyner e Ziv. Na literatura, existem os trabalhos de Wu, Tsai e Chang [14] e Lima Filho [17], que trabalham com tais metodologias para compressão de ECG, os quais buscam reverter o paradigma de compressão atual e assim transferir a maior parte de exploração estatística para o decodificador. Diferente do trabalho de Wu, Tsai e Chang [14], que utiliza quantização vetorial, essa dissertação emprega quantizações uniforme e em torno de zero do tipo *midtread*, de maneira similar ao trabalho de Lima Filho [17]. Entretanto, a metodologia dessa dissertação apresenta diferenças marcantes em relação à proposta de Lima Filho [17], a qual utiliza apenas a codificação direta de sinais de ECG. Nesse trabalho, por outro lado, realiza-se tanto a codificação direta quanto o processamento de resíduos. Além disso, será apresentada uma metodologia que elimina o canal de retorno, o qual faz solicitação de *bits* de paridade e, com isso, permite o envio único do menor número de bits possíveis.

Basicamente a estrutura dos CODECs implementados nessa dissertação seguem o modelo de *Stanford*, explanado na Seção 5.3, que foi desenvolvida para aplicação de vídeos, sua estrutura é composta por dois codificadores, Wyner-Ziv e convencional, seguidos de seus respectivos decodificadores, sendo que a decodificação Wyner-Ziv é realizada com auxílio de SI gerada por quadros-chaves interpolados. SIs gerados são previsões de WZSs que serão utilizados no decodificador Wyner-Ziv para auxiliar a decodificação de WZSs, portanto, contém erros, pois SIs são versões ruidosas de WZSs, por isso, o código corretor de erros é utilizado, que neste trabalho são os códigos turbo. Toda essa estrutura permite a transferência da exploração estatística do codificador para o decodificador, possibilitando assim a implementação de codificadores mais simples, onde são realizadas simples operações lógicas, o que pode ser utilizado em dispositivos embarcados com menor poder de processamento.

## 6.2 Arquitetura de compressão distribuída de ECG

Na metodologia proposta, ilustrada na Figura 6.1, sinais de ECG passam por uma etapa de pré-processamento, que recebe períodos originais de batimentos cardíacos e tem objetivo de

facilitar a exploração de redundâncias presentes em sinais de ECG. O passo de pré-processamento é dividido em quatro etapas: detecção de períodos de ECG, normalização de períodos, equalização DC e ordenação por similaridade, aplicadas nessa ordem, de maneira similar ao que é feito no trabalho de Lima Filho [17]. A detecção de períodos identifica complexos *QRS*, o que pode ser executado utilizando-se algoritmos de extração de características de registros de ECG, e assim todos períodos de um sinal de entrada são identificados [72]. Em seguida, todos os períodos são organizados, de maneira que o início de cada um comece com a segunda metade do complexo *QRS* atual e termine com a primeira metade do próximo complexo. A seguir, todos os períodos são normalizados, de modo que eles tenham o mesmo tamanho. Então, outra técnica, chamada de equalização DC, retira o nível DC de todos os segmentos de ECG, nivelando assim todos os segmentos. A última parte do pré-processamento é a ordenação por similaridade, que organiza todos os períodos por ordem de semelhança, através da métrica de erro quadrático médio. Essa ordenação procede da seguinte forma: o período com menor menor variância é posicionado na primeira posição e, na segunda, coloca-se o segmento mais semelhante, conforme erro quadrático médio (*Mean Squared Error - MSE*) baseado no primeiro, o que é feito do mesmo modo para todos os segmentos, até o último. Terminado esta última etapa, segmentos adjacentes de ECG tornam-se mais semelhantes, o que é ideal para codificação distribuída.

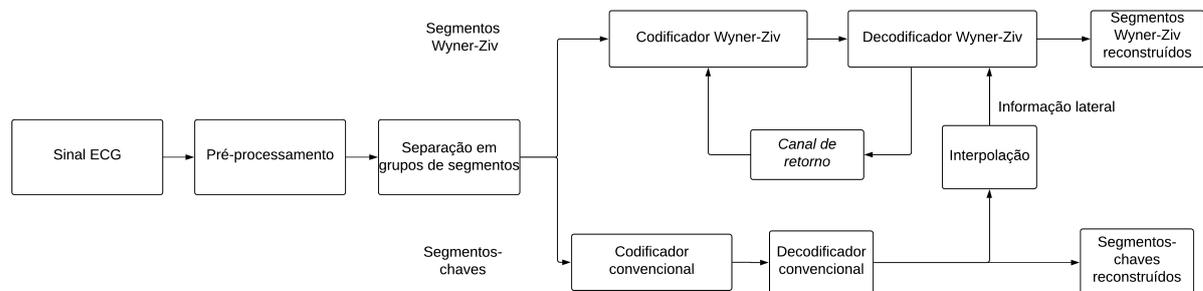


Figura 6.1: Arquitetura da compressão distribuída de ECG.

Após a etapa de pré-processamento,  $X_n$  períodos são formados, os quais são chamados segmentos. Esses elementos  $X_n$  são divididos em dois grupos: KSs e WZSs. Um grupo de  $Z$  segmentos, denominado GoS<sup>1</sup>, com grupos de 3 ou 5 segmentos, é então criado. No caso de GoS-3, a cada três segmentos, o primeiro e último são KSs, sendo o do meio WZS, de maneira que o terceiro é o primeiro segmento da próxima sequência. Para o caso de GoS-5, a estrutura

<sup>1</sup>O conceito de GoS surgiu inspirado nos trabalhos de codificação distribuída de vídeo, com o objetivo de criar uma estrutura com informação a ser comprimida, com base nos conceito de Wyner e Ziv [7, 12].

segue assim  $(KS_1, WZS_1, WZS_2, WZS_3, KS_2)$ , onde o primeiro e último segmentos são KSs, ao passo que os três do meio são WZSs.

WZSs são enviados ao codificador Wyner-Ziv, que é composto por uma etapa de quantização e separação em planos de *bits*, seguidos da etapa de codificação que produz e envia *bits* de paridade ao decodificador, sendo a taxa de envio controlada por uma etapa de perfuração, que permite o envio mínimo de *bits* de paridade, sendo estes recebidos pelo decodificador que realiza a decodificação, ou seja, correção dos *bits* errados inseridos por SIs, utilizando para isso o algoritmo MAP, que faz a estimativa de todos *bits* realizando decodificação de forma iterativa. No final da última iteração, uma decisão é tomada com base na quantidade de erros da informação decodificada, sendo estes maior que 0,001, então, mais *bits* são solicitados pelo canal de retorno, caso contrário, todos *bits* são considerados decodificados. Após a decodificação de todos os planos de *bits*, o processo inverso da quantização é realizado gerando uma matriz com todos os segmentos quantizados, sendo então reconstruídos com auxílio de SIs e dados provenientes da etapa de pré-processamento. Assim todos os segmentos são considerados reconstruídos e prontos para análise.

Nas seções seguintes todas etapas serão abordadas detalhadamente, desde o início que é a etapa de pré-processamento até o último passo que é a reconstrução.

### 6.3 Pré-processamento

Sinais de ECG podem sofrer alterações durante exames, seja por condições de pacientes, como respiração, ruídos causados por mal contato de sensores ou até mesmo anomalias do próprio coração. Portanto, existe a necessidade de pré-processamento nos sinais de ECG, de maneira que as sequências de períodos se tornem mais semelhantes, melhorando assim o desempenho de esquemas de compressão [73].

De fato, técnicas de pré-processamento aliadas a compressores levam ao aumento de qualidade na reconstrução de sinais de ECG [17]. Conforme mencionado, tal etapa, na sua forma mais completa, pode ser dividida em quatro partes: detecção de períodos de ECG, normalização de períodos, equalização DC e ordenação por similaridade. A etapa de pré-processamento utilizada está ilustrada na Figura 6.2.

A detecção de períodos de ECG, mostrada na Figura 6.3, identifica complexos *QRSs* e pode ser executada utilizando-se algoritmos de extração de características de sinais de ECG.

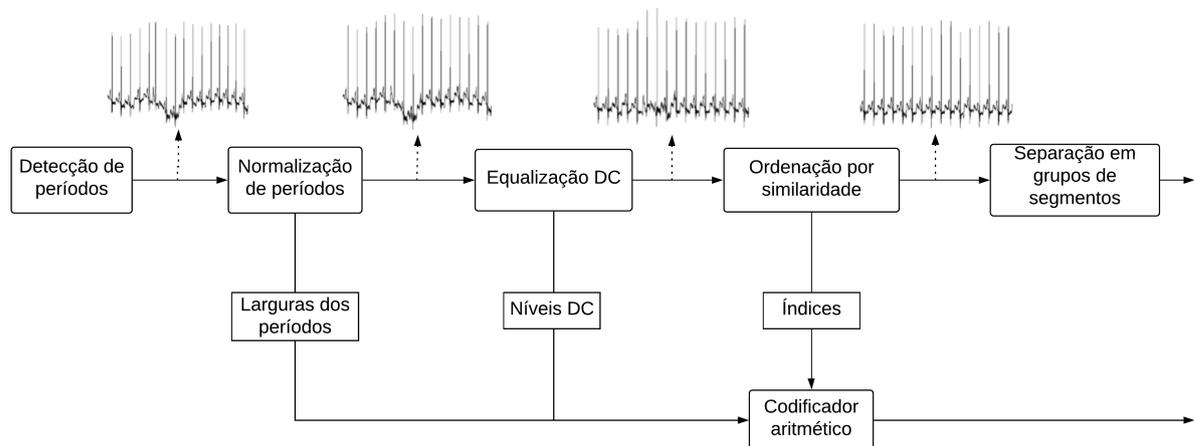


Figura 6.2: Etapa de pré-processamento executada em codificadores.

Dentre esses, existem detectores de complexos *QRS*s, que os identificam em todo o sinal de ECG [72]. Tal algoritmo basicamente encontra picos e larguras de complexos *QRS*, onde a onda *Q* é geralmente a maior de todo o período, possuindo largura  $l$ , e o pico invertido da onda *R* é subsequente a esta [72].

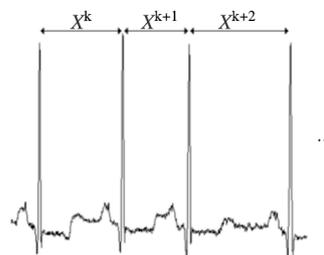


Figura 6.3: Detecção de períodos de ECG.

Após a detecção de períodos, a técnica de normalização é aplicada. Dado que, normalmente, períodos de ECG tem tamanhos diferentes, eles precisam ser normalizados para um tamanho comum, usando-se a técnica conhecida como *spline*<sup>2</sup>, com o objetivo de facilitar a exploração de redundâncias. Há várias técnicas de normalização, como, por exemplo, a conversão de larguras de períodos para o valor da maior ou da menor largura, conforme encontrado em segmentos de ECG originais, mas a opção utilizada nesse trabalho foi a que emprega a

<sup>2</sup>Segundo esse procedimento, o sinal de ECG com um período igual a  $X [x(0)x(1)x(2) \dots x(N)]$  pode ser convertido para um período normalizado igual a  $X_n = [x_n(1)x_n(2) \dots x_n(M)]$  [74].

média de todos segmentos, pois esta foi a que apresentou os melhores resultados. O efeito da normalização é mostrado nas Figuras 6.5 (b), 6.7 (b) e 6.6 (b).

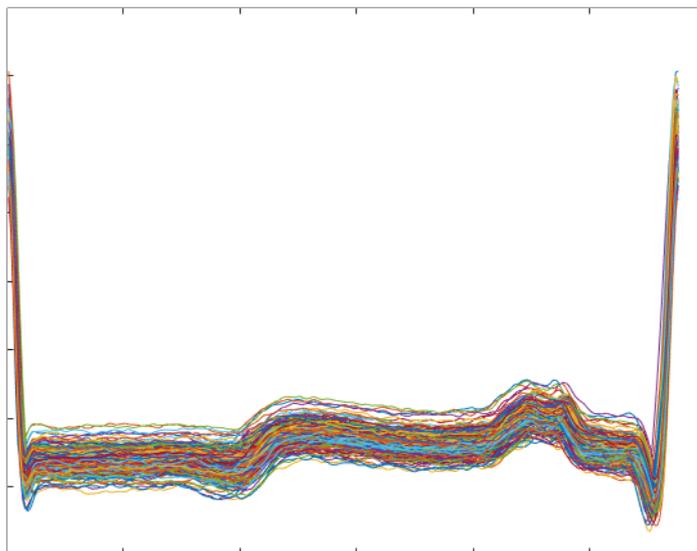


Figura 6.4: Períodos de ECG normalizados.

Geralmente, depois da etapa de normalização, percebe-se que períodos ainda não apresentam o mesmo nível médio, conforme pode ser visto na Figura 6.5 (b). Isso acontece porque as condições que um dado paciente apresenta, durante um exame, podem alterar níveis de sinais de ECG [26], assim como fatores técnicos. Portanto, uma equalização desses níveis é necessária e pode ser realizada através da fórmula

$$x_{i,k}^{DC} = x_{i,k} - (DC_k - DC_{min}), \quad (6.1)$$

onde  $x_{i,k}^{DC}$  são amostras equalizadas  $i$  do  $k$ -ésimo período,  $x_{i,k}$  são amostras originais,  $DC_k$  é o nível DC desse período e  $DC_{min}$  é o mínimo valor DC possível. Resultados da equalização DC podem ser observados nas Figuras 6.5 (c), 6.6 (c) e 6.7 (c).

Depois da equalização DC, períodos adjacentes de ECG continuam muito diferentes, principalmente devido a variações de componentes e dos próprios pacientes, conforme já comentado. De fato, pode haver períodos distantes no tempo e ainda muito semelhantes, por todo o registro de ECG, e, se eles estivessem próximos, redundâncias seriam melhor exploradas por codificadores. Tal problema pode ser resolvido com aplicação da métrica de  $MSE$ , o qual é calculado para cada período usando como referência o segmento que apresenta menor variância. Os períodos são organizados de maneira que o período que tem menor variância é colocado na primeira posição e os demais nas posições restantes, de acordo o aumento do  $MSE$ , conforme

$$MSE_k = \frac{1}{L_N} \sum_{i=1}^{L_N} (x_{i,k}^{DC} - x_{i,\sigma min})^2, \quad (6.2)$$

onde  $x_{i,k}^{DC}$  são amostras  $i$  do  $k$ -ésimo período e  $x_{i,\sigma min}$  são amostras do período com menor variância. As Figuras 6.5 (d), 6.6 (d) e 6.7 (d) mostram o efeito da etapa de ordenação por similaridade, de acordo com técnicas apresentadas aqui. Entretanto, ao término da última etapa de pré-processamento, no decodificador a etapa de reconstrução necessita das informações de larguras originais, níveis DC e também a ordem original dos segmentos, sendo assim, essas informações são enviadas como informação auxiliar ao decodificador através de um codificador aritmético, conforme mostrado anteriormente na Figura 6.2. Depois desse passo, o sinal está pronto para a próxima parte, que é a separação em grupos de segmentos.

As Figuras 6.6 e 6.7 representam os efeitos da etapa de pré-processamento nos registros 117 e 119 da base de dados do MIT-BIH [23], o resultado final da etapa de pré-processamento na Figura 6.6 mostra uma imagem mais suave e regular e pode ser comprimida com eficiência. No caso do registro 119 mostrado na Figura 6.7, o resultado final do pré-processamento mostra que há três regiões distintas, o que é resultado da ordenação por similaridade, a qual permite que segmentos adjacentes sejam mais similares, o que garante melhor exploração de redundâncias de sinais de ECG.

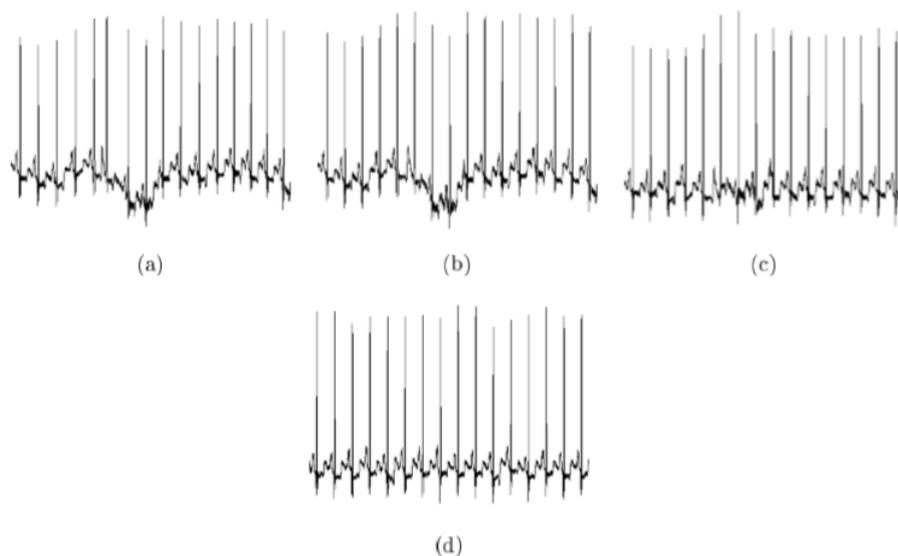


Figura 6.5: Etapa de pré-processamento. (a) Períodos originais, (b) Períodos normalizados, (c) Sinal com equalização DC, (d) Sinal com ordenação por similaridade.

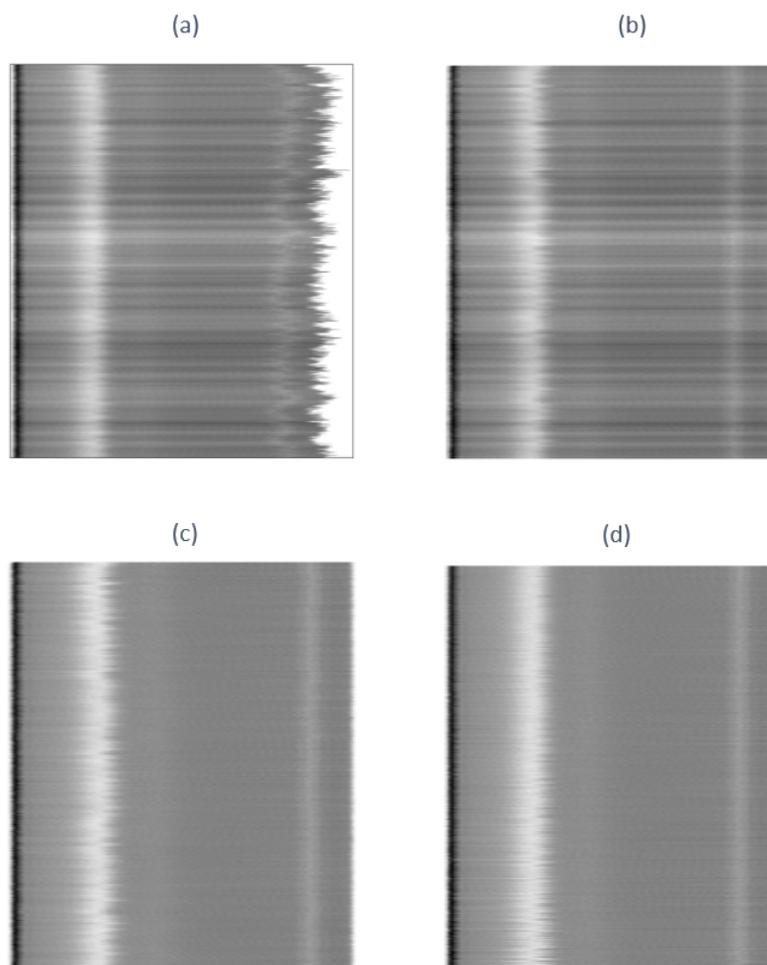


Figura 6.6: Registro 117 do MIT-BIH [23] após etapa de pré-processamento. (a) Períodos originais, (b) Períodos normalizados, (c) Sinal com equalização DC, (d) Sinal com ordenação por similaridade.

Após pré-processamento o sinal de ECG precisa ser adaptado a estrutura do DSC que é constituído por dois codificadores, sendo assim, todos segmentos pré-processados são divididos em dois grupos chamados de WZSs e KSs. Essa separação em GoS surgiu em arquiteturas de codificação distribuída de vídeo [7, 12], com o objetivo de adequar a sequência de quadros ao DSC, portanto, de forma análoga ao DSC de vídeo, esta ideia é adaptada na metodologia de codificação distribuída proposta, aplicada à compressão de ECG.

O GoS é constituído por uma certa quantidade de WZSs e KSs, sendo sua largura definida pela quantidade de WZSs, onde o primeiro e último são KSs, sendo WZS ou WZSs colocados entre segmentos KSs. Por exemplo, seja a sequência  $(KS_1, WZS_1, KS_2)$ , nesse caso a largura do GoS é 3 (GoS-3), conforme ilustrado na Figura 6.8. Para fins de seleção de qual tamanho de GoS utilizar, larguras de tamanho 3, 5, e 7 foram experimentalmente analisadas,

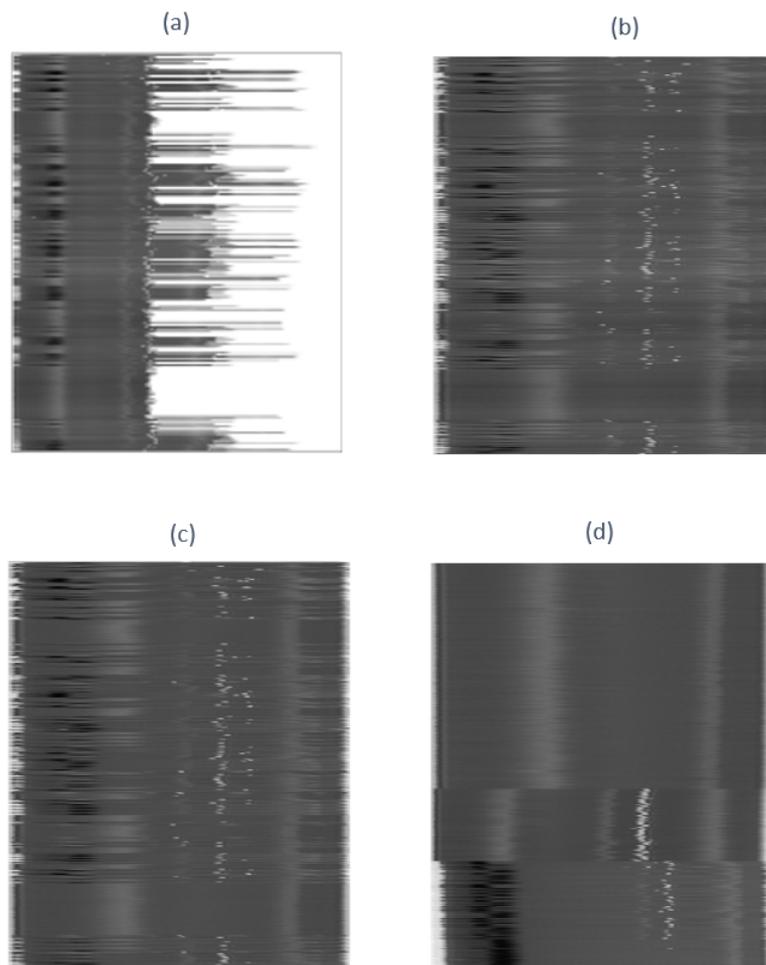


Figura 6.7: Registro 119 do MIT-BIH [23] após etapa de pré-processamento. (a) Períodos originais, (b) Períodos normalizados, (c) Sinal com equalização DC, (d) Sinal com ordenação por similaridade.

onde GoS de tamanho 3 que é o menor agrupamento possível mostrou resultados superiores a maiores larguras, de maneira que, à medida que o tamanho de GoS aumenta, o desempenho de compressão diminui. Por essa razão, GoS-3 e GoS-5 são as larguras utilizadas no CDA e demais CODECs implementados nessa dissertação.

A qualidade da reconstrução de WZSs está interligada a qualidade de sua predição SI, que é criada pela interpolação de KSs. Conforme ilustrado na Figura 6.8, no caso de GoS-3, a interpolação é realizada calculando a média de KSs, ou seja, o primeiro SI é gerado pela equação  $SI_1 = (KS_1 + KS_2)/2$ , diferente de GoS-5 que utiliza média ponderada, onde SIs são calculados utilizando o peso de KSs, ou seja,  $SI_n = p_1KS_n + p_2KS_{n+1}$ , sendo  $(p_1, p_2)$  referentes a pesos da média ponderada e possuem valores diferentes de acordo com  $SI_n$ . Pesos com valores iguais a (3,1), (1,1) e (1,3) foram utilizados, referentes, respectivamente, à  $SI_n$ ,  $SI_{n+1}$  e  $SI_{n+2}$ .

Diferentes valores de pesos ( $p_1, p_2$ ) foram experimentalmente analisados, de maneira que os valores comentados anteriormente foram os apresentaram melhor desempenho, e por isso foram utilizados.

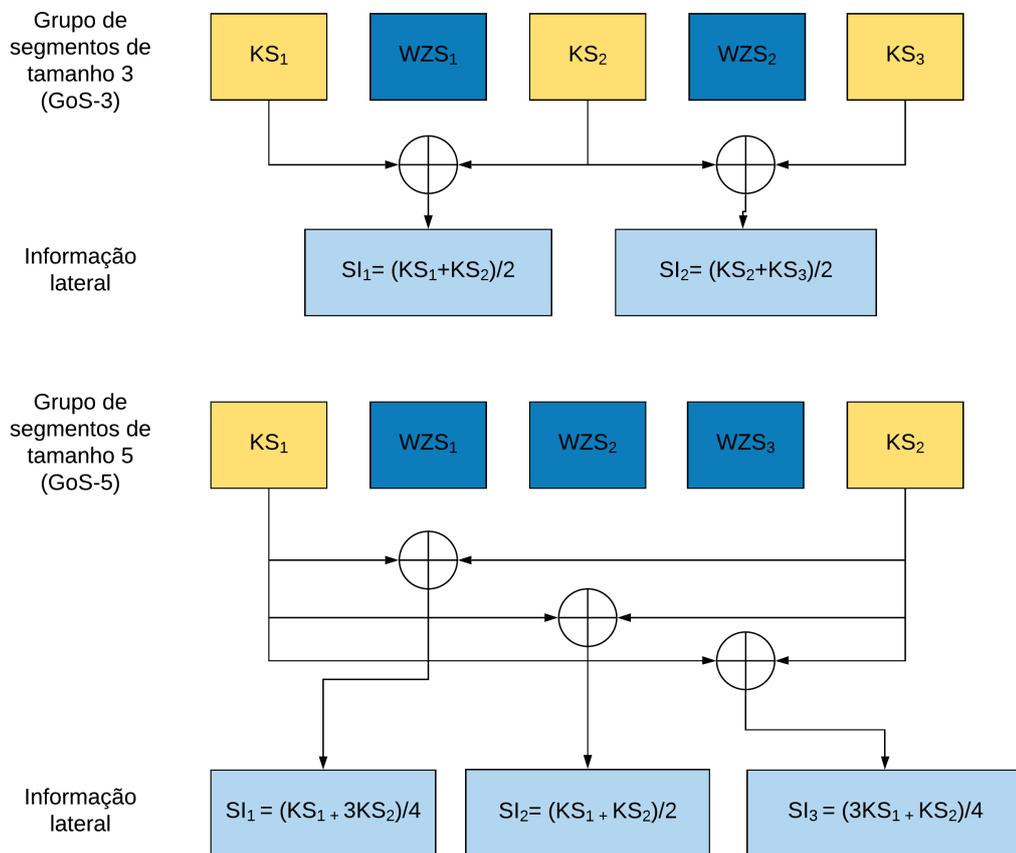


Figura 6.8: Esquema da predição de WZSs utilizando método de interpolação de KSs.

Nesse trabalho, a técnica de separação de segmentos em grupos é utilizada (Figura 6.8), sendo dois grupos utilizados: GoS-3 e GoS-5. Sendo assim, o CDA e demais CODECs implementados, utilizam duas opções de agrupamento, GoS-3 e GoS-5.

## 6.4 CODEC Wyner-Ziv no domínio da amostra

O codificador Wyner-Ziv no domínio da amostra (CDA), representado na Figura 6.9, processa diretamente amostras de ECG após separação em GoS. A arquitetura utilizada é similar à proposta por Aaron, Zangh e Girod [7] e ambos utilizam um quantizador uniforme, um CODEC (*i.e* enCOder/DECOder) turbo baseado na teoria de Slepian-Wolf, interpolação de KSs e um módulo de reconstrução.

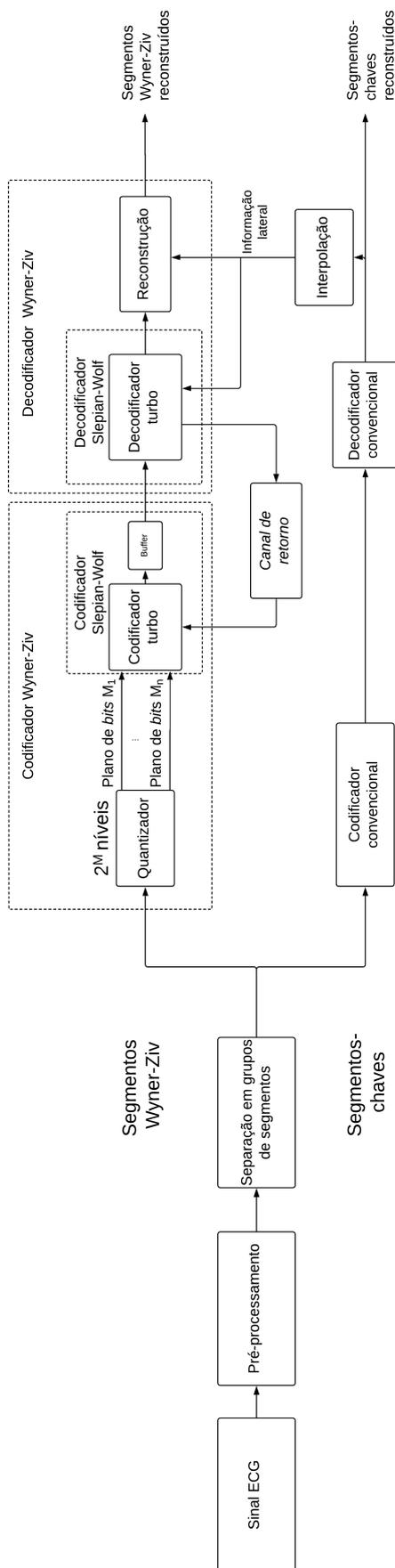


Figura 6.9: Esquema geral do CODEC Wyner-Ziv no domínio da amostra.

Os WZSs são agrupados, formando assim um único vetor e depois quantizados por um quantizador uniforme do tipo *midtread*, com  $2^b$  níveis de quantização, sendo  $b$  o número de *bits*, podendo assumir valores inteiros entre 1 e 8, o que corresponde respectivamente a 2 e 256 níveis. O intervalo de quantização é dado por

$$Q = \frac{R}{2^b}, \quad (6.3)$$

onde  $R$  representa a faixa dinâmica do sinal, ou seja, a diferença entre o maior e menor valor da amostra de WZSs. Portanto, o índice  $Q_f$  de uma amostra  $f$  de WZSs é dado por

$$Q_f(f) = \text{round}\left(\frac{f}{Q}\right). \quad (6.4)$$

Alterando  $b$  na equação 6.3, diferentes pontos de distorção são alcançados, sendo assim, quatro pontos foram utilizados no CDA, os quais são: 2, 3, 4 e 5. Após esta etapa, os símbolos quantizados de WZSs são convertidos em números binários e arranjados em planos de *bits*, formando vetores independentes, onde cada vetor contém *bits* que possuem a mesma posição, ou seja,  $n$  vetores são criados onde o primeiro contém *bits* mais significativos e o último contém *bits* menos significativos. A codificação é executada, inicialmente, a partir do conjunto de *bits* mais significativos, de forma independente dos outros planos de *bits*. Cada plano de *bits* é enviado a próxima etapa que é o codificador de Slepian-Wolf, o qual será detalhado a seguir, assim como os demais módulos do CDA.

### 6.4.1 O codificador Slepian-Wolf

Nesta etapa cada plano de *bit* é inserido no codificador Slepian-Wolf, começando com o vetor de *bits* mais significativos, cada vetor é codificado e decodificado separadamente. O codificador Slepian-Wolf inclui um codificador turbo (Figura 6.10) e um *buffer* e procede da seguinte forma:

- O codificador turbo produz uma sequência de *bits* de paridade (*bits* redundantes) para cada vetor de planos de *bits*; a quantidade de *bits* de paridade produzidos para cada plano depende da taxa do codificador turbo, isto é, da razão de *bits* que saem desse codificador por *bit* que entra. Essa razão é controlada pela perfuração de *bits*, que é a eliminação de alguns *bits* conforme um padrão de perfuração que será apresentada mais a frente;

- Os *bits* sistemáticos, são normalmente utilizados em codificadores turbo sem utilização de técnicas de DSC, porém, na arquitetura DSC os *bits* sistemáticos utilizados são justamente

os fornecidos por SIs já presentes no decodificador. Portanto, tais *bits* sistemáticos gerados no codificador são desconsiderados;

- Os *bits* de paridade correspondem a versão compactada da informação, ou seja, representam o sinal comprimido, estes são produzidos pelo codificador turbo e são armazenados no *buffer*, perfurados e transmitidos mediante solicitação do decodificador através do canal de retorno. A Figura 6.10 ilustra a estrutura do codificador turbo utilizada, o qual é constituído de dois codificadores recursivos sistemáticos convolucionais (RSC) idênticos, como proposto por Berrou, Glavieux e Thitimajshima [54]. Entre os RSCs, dois entrelaçadores aleatórios são empregados para descorrelacionar a entrada do codificador 1 e 2, diminuindo rajadas de erros seguidas provenientes de SIs [26]. O funcionamento do entrelaçador é descrito a seguir.

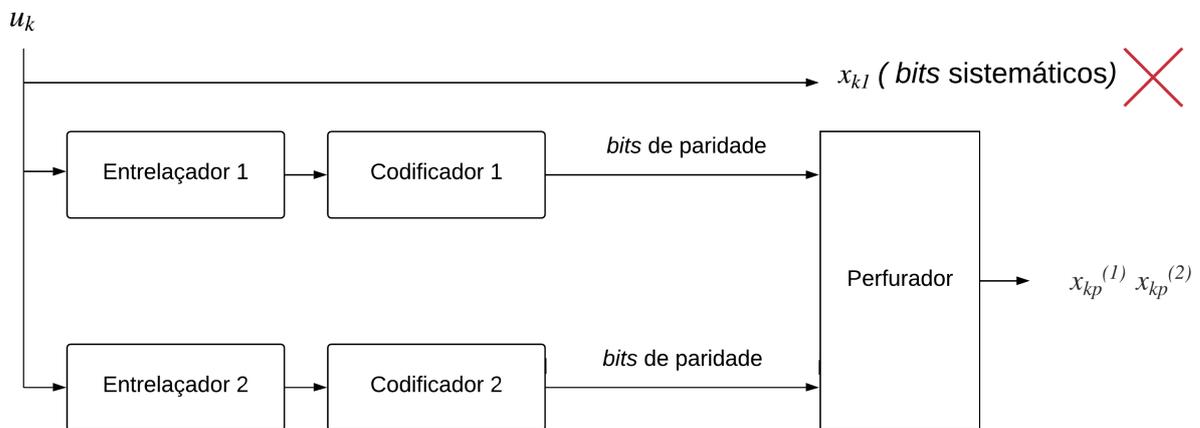


Figura 6.10: Visão geral do codificador Wyner-Ziv.

### 6.4.2 O entrelaçador

Um dos módulos do codificador turbo é o entrelaçador. Genericamente, a sequência de saída do entrelaçador é a sequência de entrada reorganizada em ordem diferente de acordo com um determinado padrão. Por exemplo, seja um padrão de entrelaçamento  $P = [14725936810]$  de uma sequência de entrada de comprimento de 10 bits, e  $m = [24318596010]$  um vetor antes de ser entrelaçado. Portanto, o vetor  $m$  entrelaçado conforme o padrão de entrelaçamento  $P$  torna-se  $m^P = [21948035610]$ .

Existem diversos tipos de entrelaçadores para os códigos turbo. Nesse trabalho o entrelaçador utilizado foi do tipo aleatório, que mostra ser bastante eficiente para médias e grandes larguras [75], sendo que o decodificador tem conhecimento do padrão de entrelaçamento, per-

mitindo assim a realização da etapa de decodificação. A literatura comprova que a largura  $L$  dos *bits* inseridos no codificador altera o desempenho do codificador turbo, de maneira que grandes larguras aproximam o resultado perto do canal de Shannon [59].

Visto todo o funcionamento do entrelaçador, depois desta etapa, *bits* entrelaçados são enviados ao codificador que é detalhado a seguir.

### 6.4.3 A codificação

A arquitetura do codificador turbo foi mostrada na Figura 6.10, o qual inclui, ao lado de cada entrelaçador, codificadores recursivos sistemáticos convolucionais (RSC). O codificador RSC utilizado está ilustrado na Figura 6.11 é tipicamente caracterizado por uma matriz geradora  $G$  que permite obter a saída do codificador RSC para uma dada entrada, a qual é dada por

$$G = \left[ 1 \quad \frac{1+D+D^3+D^4}{1+D^3+D^4} \right] \quad (6.5)$$

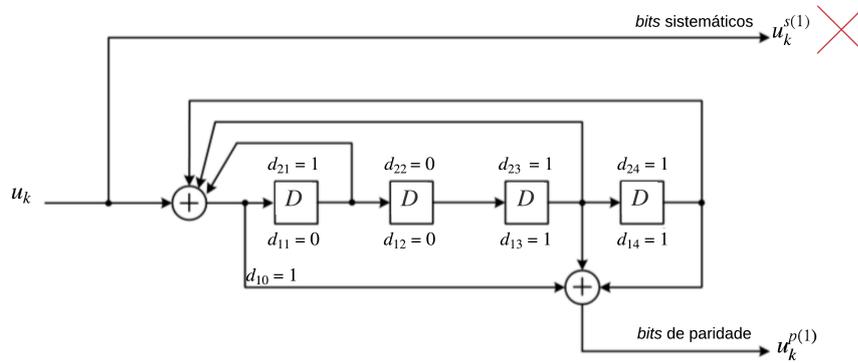


Figura 6.11: Codificador recursivo sistemático convolucional com 4 memórias, 16 estados e a sua matriz geradora representada pela equação (6.5).

Os *bits* sistemáticos  $u_k^{s(1)}$  são iguais aos *bits* de entrada  $u_k$ , contudo, não são enviados, pois os *bits* de SIs que serão utilizados. Os *bits* de paridade  $u_k^{p(1)}$  podem ser dados por

$$u_k^{p(1)} = d_{10} + (s_1^{k-1} \oplus d_{11}) + (s_2^{k-1} \oplus d_{12}^{k-1}) + (s_3^{k-1} \oplus d_{13}^{k-1}) + (s_4^{k-1} \oplus d_{14}^{k-1}), \quad (6.6)$$

onde  $d_{10} = u_k + (s_1^{k-1} \oplus d_{21}) + (s_2^{k-1} \oplus d_{22}^{k-1}) + (s_3^{k-1} \oplus d_{23}^{k-1}) + (s_4^{k-1} \oplus d_{24}^{k-1})$ , o operador  $\oplus$  representa a soma módulo-2 (operação *or-exclusivo*). O símbolo  $s_n^{k-1}$  ( $n=1, 2, 3, 4$ ) representa os registros do deslocamento de  $D$  do estado anterior ( $k-1$ ).

É possível construir uma representação que mostre, para cada estado do codificador RSC, as possíveis transições de estado, bem como a sequência de saída RSC dada uma sequência de entrada de *bits* no codificador RSC. Essa representação é chamada de treliça [76]. A treliça da função geradora representada pela equação (6.5) está ilustrada na Figura 6.12.

Conforme comentado, o codificador turbo realiza apenas operações lógicas somas módulo-2, diferente de técnicas de codificação empregadas por codificadores tradicionais, como por exemplo o JPEG explanado na Seção 4.5, que utiliza técnica de transformada DCT, além de etapa de codificação de entropia, sendo esta última responsável por realizar explorações estatísticas no codificador.

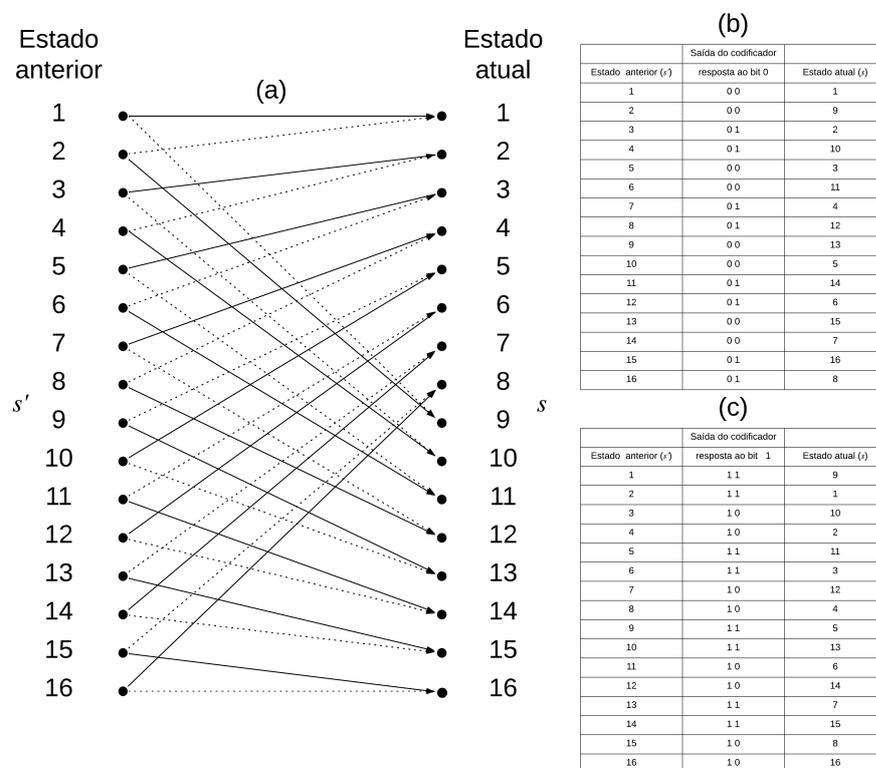


Figura 6.12: (a) Representação da transição de estados da função geradora determinada pela equação (6.5). A Figura (b) mostra a saída do codificador, estado atual e transição para o estado futuro em resposta ao *bit* 0. A Figura (c) mostra a saída do codificador, estado atual e transição para o estado futuro em resposta ao *bit* 1.

Como visto na treliça representada na Figura 6.12, os instantes de tempo  $s'$  e  $s$  representam os estados passados e futuros, sendo que ambas possuem a mesma quantidade de probabilidades, que são os 16 estados, esta treliça é constituída a partir da função geradora representada pela equação (6.5), de maneira que, a quantidade de memórias determina a quantidade de estados possíveis, ou seja, o codificador convolucional representado na Figura 6.11 apresenta 4

memórias que podem representar até 16 estados. Baseado nisso, estruturas de codificadores convolucionais com 3 e 5 memórias foram testadas, sendo que a utilização de 3 memórias apresentaram menor desempenho, ou seja, menor quantidade de erros corrigidos pelo decodificador, o que exige o envio de mais *bits* de paridade, em contrapartida, a utilização de 5 memórias apresentaram desempenho levemente superior, comparada à 4 memórias, porém, o tempo de decodificação aumenta substancialmente, pois a quantidade de estados aumenta consideravelmente, e como consequência a quantidade de cálculos realizados. Portanto, a utilização de 4 memórias é ideal, pois, possui ótima relação de tempo de processamento e desempenho.

Ao término da codificação de determinado plano de *bits*, todos *bits* de paridade gerados são enviados a uma etapa que controla a quantidade de *bits* enviados ao decodificador, a qual é chamada de perfuração, sendo detalhada a seguir.

#### 6.4.4 Etapa de perfuração

Para garantir uma quantidade mínima de *bits* de paridade enviados ao decodificador, o CODEC de Slepain-Wolf possui uma etapa chamada perfurador, que consiste na eliminação de alguns *bits* conforme um padrão de perfuração decrescente [77]. O padrão de perfuração adotado é similar ao descrito por Brites [67], a qual implementa um método que apresenta bons resultados.

O esquema de perfuração é exemplificado na Figura 6.13 onde as saídas dos codificadores convolucionais passam pelo processo de perfuração. As saídas dos RSCs são representadas por  $u_k^{p(1)}$  e  $u_k^{p(2)}$ , ambas passam pelo processo de perfuração que procede assim: vetores de saída  $u_k^{p(1)}$  e  $u_k^{p(2)}$  possuem a mesma quantidade de 8 *bits*, neste exemplo cada vetor é subdividido em blocos com tamanho  $L = 4$  que é a largura de perfuração. Inicialmente, nenhum *bit* de paridade é enviado, se o decodificador solicitar mais *bits*, então o primeiro *bit* de cada bloco representado por  $(b_1, b_5)$  ( Figura 6.10) de  $u_k^{p(1)}$  e  $(p_1, p_5)$  de  $u_k^{p(2)}$  são enviados, caso o decodificador solicite mais *bits* então o segundo de cada bloco são enviados, os quais são os *bits*  $(b_2, b_6)$  e  $(p_2, p_6)$ . Esse procedimento continua até que o critério de parada seja atingido que é a uma taxa de erro na decodificação menor que 0,001. Valores de larguras como 8, 16, 32 e 64 foram experimentalmente utilizadas, sendo que a largura de 32 *bits* por bloco, apresenta melhores resultados e por isso foi adotada nesse trabalho, ou seja, o vetor de saída de cada codificador convolucional é dividido em blocos menores com largura  $L = 32$  e são enviados ao

decodificador, descrito nas seções seguintes, conforme solicitação pelo canal de retorno.

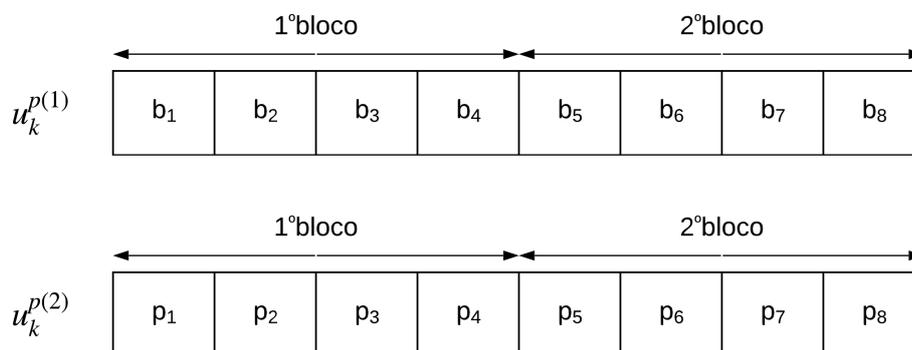


Figura 6.13: Esquema de perfuração utilizando blocos com largura de 4 *bits*.

### 6.4.5 Interpolação de KSs

Como visto anteriormente, os WZSs são enviados ao codificador Wyner-Ziv e KSs são enviados ao codificador convencional. KSs auxiliam a decodificação de WZSs, logo, quanto mais similaridade houver entre ambos, melhor o desempenho de codificação de WZSs, sendo assim, estimativas de WZSs são realizadas, utilizando, para isso, interpolações de KSs.

A Seção 6.3 mostra como é formada a estimativa de WZSs. Sendo que SIs gerados por interpolações são aproximações de WZSs. Como visto na Seção 6.4.1 cada codificador gera *bits* de paridade além de *bits* sistemáticos, contudo, estes últimos não são enviados ao decodificador, pois serão utilizados *bits* sistemáticos gerados por SIs. Entretanto, cada SI é uma estimativa de WZS, logo, pode conter *bits* errados, o que é esperado, pois é uma predição, portanto, *bits* de paridade auxiliam na correção de *bits* sistemáticos errados na etapa de decodificação. KSs fornecidos por sinais de ECG possuem boa similaridade com WZSs, e isso possibilita envio de menos *bits* de paridade, pois menos correções são necessárias.

Resumidamente os KSs interpolados representam formas ruidosas de WZSs, ou seja, predições que buscam aproximações com valores de WZSs, e por isso contém erros, os quais são corrigidos com auxílio de *bits* de paridade na etapa de decodificação apresentada a seguir.

### 6.4.6 A decodificação

O procedimento de decodificação de códigos turbo é realizado utilizando o algoritmo MAP (visto na Seção 5.4.2). Resumidamente, o algoritmo MAP calcula estimativas de quais



$$f(k) = \frac{\alpha}{2} \exp^{-\alpha |WZS_n(k) - SI_n(k)|}, \quad (6.7)$$

$\alpha$  é calculado no codificador conforme a equação 6.8 e transmitido ao decodificador como informação auxiliar, sendo utilizado no algoritmo MAP, precisamente na distribuição que faz estimativa de *bits* sistemáticos, a qual será detalhada mais adiante.

$$\alpha = \sqrt{\frac{2}{\sigma^2}}. \quad (6.8)$$

o parâmetro  $\sigma^2$  é a variância do resíduo entre WZSs e SIs [78].

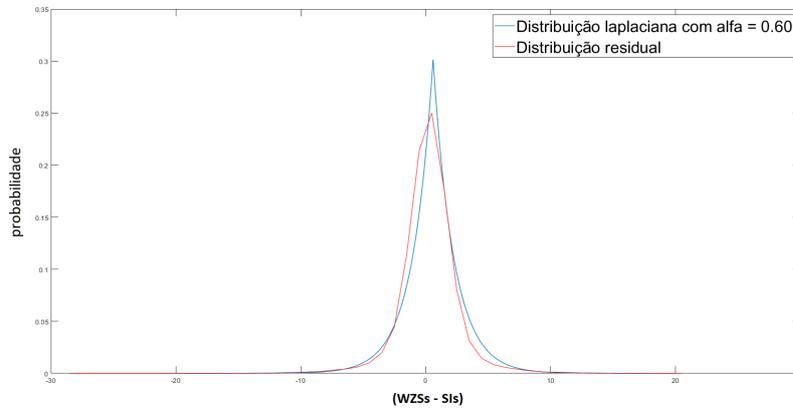


Figura 6.15: Distribuição residual do registro 100 da base de dados do MIT-BIH [23].

Como visto na Seção 5.4.2 as estimativas de *bits*, que chegam na etapa de decodificação, estão relacionadas a cálculos de LLRs. Portanto, a estimativa  $u_k$  é calculada pela LLR na equação (6.9):

$$L(u_k|y) = \ln \frac{P(u_k = 1|y)}{P(u_k = -1|y)} = \ln \frac{\sum_{R_1=0}^{15} P(s', s, y)}{\sum_{R_0=0}^{15} P(s', s, y)} = \ln \frac{\sum_{R_1=0}^{15} \alpha_{k-1}(s') \gamma(s', s) \beta_k(s)}{\sum_{R_0=0}^{15} \alpha_{k-1}(s') \gamma(s', s) \beta_k(s)}, \quad (6.9)$$

onde o numerador representa as probabilidades em resposta ao *bit* "1" e o denominador a resposta ao *bit* 0, sendo este último mapeado para valores iguais a "-1", e o somatório é realizado considerando todos 16 estados, em resposta a "0" ou "1", identificados por  $R_1$  e  $R_0$  no somatório. A parcela  $P(s', s, y)$  representa a probabilidade de um estado  $s'$  transicionar para o estado  $s$  dado que recebeu determinado *bit* sistemático representado por  $y$ .

A probabilidade  $\gamma(s', s)$  é a primeira a ser calculada e representa a probabilidade condicional de, no instante  $k$ , o símbolo recebido ser  $y_k$  e o estado atual ser  $S_k = s$ , sabendo-se que o

estado de onde veio foi  $S_{k-1} = s'$ , ou seja, basicamente representa a probabilidade de transição entre estado passado e futuro. Sendo assim é calculada pelo produto de probabilidades abaixo

$$\gamma_k(s', s) = P(y_k|x_k)P(u_k), \quad (6.10)$$

onde a probabilidade *a priori*  $P(u_k)$  é definida pela equação

$$P(u_k = \pm 1) = e^{\frac{u_k}{2} L^{apriori}}, \quad (6.11)$$

onde ( $u_k = 1$ ) é utilizado no numerador e ( $u_k = -1$ ) no denominador.  $L^{apriori}$  corresponde ao conhecimento prévio da porcentagem dos *bits* serem 1 ou "-1", esse valor inicialmente vale zero, pois probabilidades iguais de "-1" e "1" são consideradas, e conforme iterações no decodificador ocorrem este valor é atualizado, possuindo assim mais confiança a cada iteração. Decodificadores trocam informações chamadas de extrínsecas que é o cálculo de LLR definido pela equação (6.9)), desta forma, a informação extrínseca calculada no primeiro decodificador é enviada ao segundo como informação *a priori*, já sendo estimativas aprimoradas. A cada troca de informações entre decodificadores, mais *bits* são corrigidos. Sendo assim,  $L^{apriori}$  é dado por

$$L^{apriori} = \frac{P(u_k^s = +1)}{P(u_k^s = -1)}, \quad (6.12)$$

onde  $u_k^s$  corresponde a *bits* sistemáticos fornecidos pela informação lateral.

A parcela  $p(y_k|x_k)$  na equação (6.10) é dividida em duas partes, onde a primeira é correspondente ao *bit* sistemático e a segunda à paridade [67]. Considerando que os *bits* de paridade são livres de erros, então, o decodificador sabe a probabilidade exata do *bit* de paridade que pode ser  $P_{u_k^p} = \pm 1$ , dado que foi enviado a paridade  $u_k^p = \pm 1$ , portanto, essa probabilidade pode ser descrita como um impulso através do delta de *Dirac*, contudo, o delta de *Dirac* não é uma função estritamente matemática [79], logo, pode ser aproximado por uma distribuição gaussiana com uma pequena variância, conforme Brites [67].

A parcela correspondente aos *bits* sistemáticos possui em sua composição ruído laplaciano proveniente de SI, conforme comentado anteriormente, logo, matematicamente são modelados pela distribuição laplaciana [67]. Portanto, o termo  $\gamma$  é calculado conforme abaixo

$$\gamma_k(s', s) = e^{\frac{u_k}{2} L^{apriori}} e^{-\alpha|x_{k1}-u_k^s|} e^{\frac{x_{kp} \cdot u_k^p}{\sigma^2}}, \quad (6.13)$$

onde na segunda exponencial  $u_k^s$  é *bit* sistemático gerado por SI. Na terceira exponencial o símbolo  $u_k^p$  refere-se ao *bit* de paridade, além da utilização de uma pequena variância  $\sigma^2$  escolhida aleatoriamente [67]. Os termos  $x_{kl}$  e  $x_{kp}$  referem-se ao fator condicional dado que  $u_k$  foi transmitido, em outras palavras representam os dígitos de saída do codificador para cada transição de estado. O decodificador tem conhecimento da treliça que representa todos estados assim como todas transições possíveis além de conhecimento prévio de valores de saída para cada transição de estado possível, sendo assim este conhecimento prévio é utilizado por meio de  $x_{kl}$  e  $x_{kp}$ , e por isso assumem valores da treliça utilizada e ilustrada na Figura 6.12, a qual encontra-se na Seção 6.4.3.

Inserindo  $\gamma$  (equação (6.13)) na equação (6.9) obtêm-se

$$L(u_k|y) = L^{apriori} + \ln \frac{\sum_{R_1=0}^{15} \alpha_{k-1} e^{-\alpha|x_{k1}-u_k^s|} e^{\frac{x_{kp} u_k^p}{\sigma^2}} \beta_k(s)}{\sum_{R_0=0}^{15} \alpha_{k-1} e^{-\alpha|x_{k1}-u_k^s|} e^{\frac{x_{kp} u_k^p}{\sigma^2}} \beta_k(s)}, \quad (6.14)$$

o termo  $L^{apriori}$  é a informação *a priori* dos *bits* que chegam ao decodificador e o termo logarítmico representa a informação extrínseca que é enviada ao segundo decodificador que a utilizada desta vez como informação *a priori*, e assim calcula uma nova informação extrínseca, agora utilizando o segundo *bit* de paridade enviado pelo segundo codificador. Ao término do cálculo da segunda informação extrínseca no segundo decodificador, a primeira iteração é completada. Um total de 18 iterações são realizadas, ao término da última, a informação  $L(u_k|y)$  é enviada a uma etapa de decisão, neste caso se  $L(u_k|y) > 0$  estima-se o *bit* sendo "1", caso contrário "0". Após decisão, a verificação de *bits* errados é realizada através do cálculo de  $P_e$ , onde são somados todos *bits* diferentes entre *bit* estimado  $\hat{u}_k$  e o *bit* da informação  $u_k$ , depois, essa soma é dividida pela largura do plano de *bits*, esse método é chamado de detecção de erro ideal e foi implementado conforme o trabalho Brites [67]. Sendo assim, se o erro  $P_e$  é maior que  $10^{-3}$  então todos *bits* são considerados decodificados, caso contrário, mais *bits* de paridade são solicitados pelo canal de retorno.

Assim são estimados todos planos de *bits* e enviados para etapa de reconstrução detalhada na próxima seção. O procedimento de trocas de informações extrínsecas, entre ambos decodificadores, está simplificado no algoritmo 1.

**Algorithm 1** MAP

---

```

para  $i = 1$  até 18 (número de iterações) faça
  entrada: decodificador 1 recebe bits sistemáticos (informação lateral) e de paridade
    enviados pelo codificador 1, e  $L_1^{apriori}$  do decodificador 2
   $L_1^{apriori}(1) = 0$ 
  início
     $L_1(u_k|y) = L_1^{apriori} + L_{e1}$ 
     $L_{e1} = L_1(u_k|y) - L_1^{apriori}$ 
  fim
  saída: informação extrínseca  $L_{e1}$  enviado ao decodificador 2 como  $L_2^{apriori}$ 
  entrada: decodificador 2 recebe bits sistemáticos (informação lateral) e de paridade
    enviados pelo codificador 2
  início
     $L_2^{apriori} = L_{e1}$ 
     $L_2(u_k|y) = L_2^{apriori} + L_{e2}$ 
     $L_{e2} = L_2(u_k|y) - L_2^{apriori}$ 
  fim
  saída: informação extrínseca  $L_{e2}$  enviado ao codificador 1 como  $L_1^{apriori}$ 
fim
 $sign(L_2(u_k|y));$ 
se  $erro < 10^{-3}$  então
  Todos bits são considerados decodificados
  senão
    Mais bits de paridade são solicitados pelo canal de retorno e o processo
    iterativo se repete
  fim
fim

```

---

**6.4.7 Reconstrução**

Planos de *bits* decodificados são agrupados formando amostras quantizadas  $Q$ . A seguir,  $Q$  pode ser reconstruído, que é, então, realizado amostra por amostra. Existem três situações possíveis na reconstrução. A primeira é quando o valor da amostra de SI fica abaixo do menor valor do intervalo de quantização da amostra quantizada, a segunda situação é quando a amostra de SI fica no meio do intervalo de quantização, e a terceira situação é quando a amostra de SI fica acima do maior valor do intervalo de quantização da amostra quantizada. No primeiro caso o menor valor do intervalo de quantização é escolhido, no segundo caso o valor correspondente de  $SI_k$  é escolhido e na última situação é escolhido o maior valor do intervalo de quantização [80]. Contudo, SI foi gerado com um preditor polarizado [26], então os melhores resultados

não são os extremos do intervalo de quantização e nem o centro. Então, utiliza-se o valor da reconstrução como

$$\hat{X}_n(k) = \beta SI_n(k) + (1 - \beta)R_n(k), \quad (6.15)$$

onde  $\hat{X}_n(k)$  é a amostra reconstruída do segmento  $n$ ,  $SI_n(k)$  é a informação lateral do segmento  $n$ ,  $R_n(k)$  é o centro do intervalo de quantização da amostra quantizada  $k$  do segmento  $n$  e  $\beta$  é a constante de polarização. A constante  $\beta$  foi definida empiricamente como  $7/8$ .

A seguir acontece o processo inverso do pré-processamento. Onde são recuperados: as ordens originais dos períodos, níveis DC e recuperação das larguras originais dos períodos, conforme a Figura 6.16. Após estes passos o sinal está completamente recuperado (Figura 6.17) e pronto para análise médica.

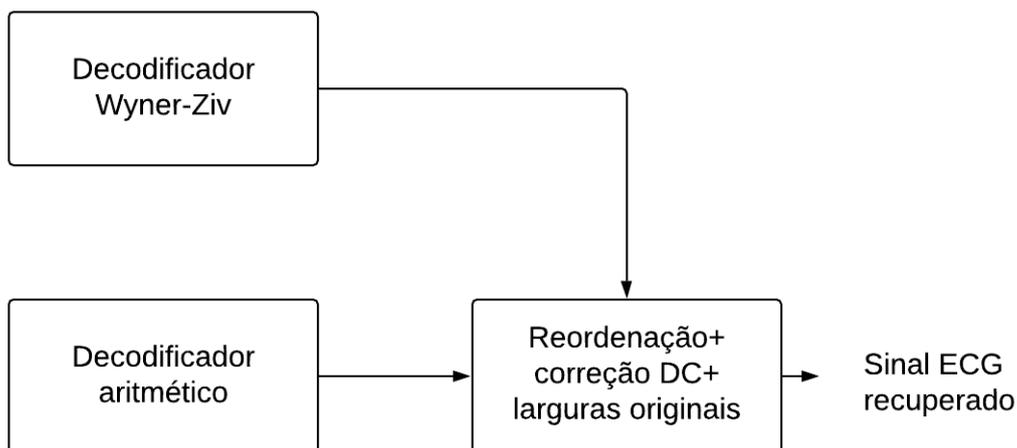


Figura 6.16: Reversão da etapa de pré-processamento.

## 6.5 CODEC Wyner-Ziv no domínio da transformada

Nesta seção será apresentado o CODEC Wyner-Ziv no domínio da transformada (CDT), que é uma extensão do CDA com adicional da transformada DCT, o qual está ilustrado em sua forma completa na Figura 6.18, sendo semelhante a estrutura do CDA, onde a etapa de transformação é realizada antes da quantização, e no decodificador após interpolação. Além disso, após etapa de reconstrução, a operação inversa da DCT chamada de IDCT é realizada.

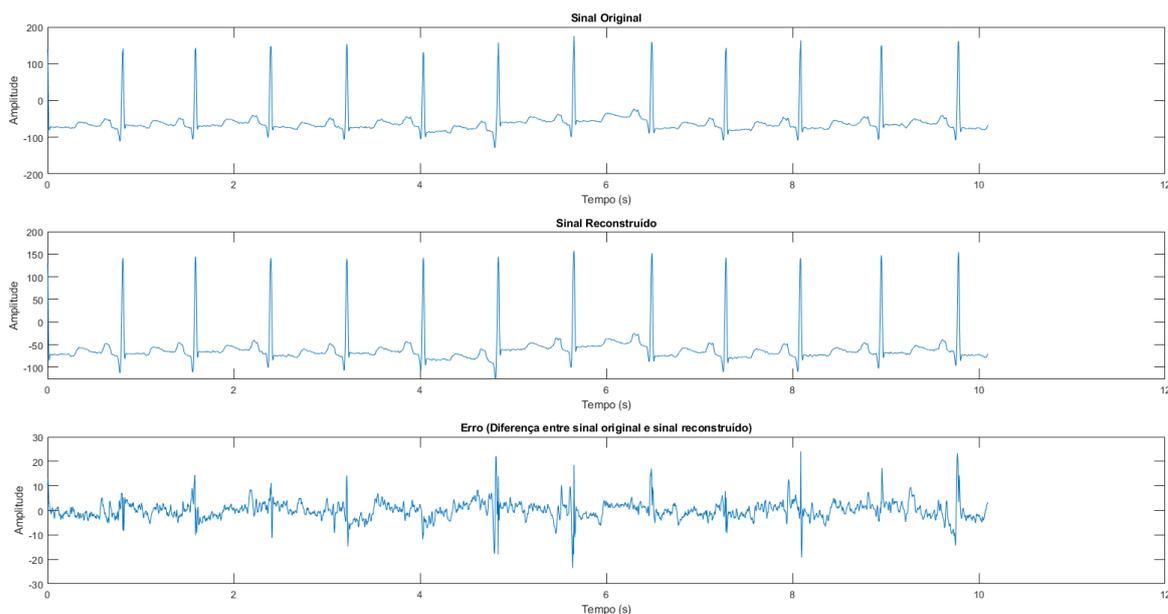


Figura 6.17: Sinal de ECG reconstruído com CDA: (a) sinal original; (b) sinal reconstruído; (c) erro de reconstrução.

Como visto no Capítulo 2, técnicas de transformações são muito utilizadas para fins de compressão de imagens e vídeos, pois exploram correlação espacial entre amostras de um quadro, e neste CODEC a transformação é aplicada utilizando blocos de matrizes  $n \times n$ , descorrelacionando amostras adjacentes de cada bloco, e assim convertendo amostras em coeficientes independentes. Sinais de ECG possuem amostras adjacentes altamente correlacionadas, além de ter alta concentração de componentes em baixa frequência, logo é mais difícil explorar redundâncias no domínio da amostra. Sendo assim, a utilização da transformada discreta do cosseno resolve essa questão, melhorando assim o desempenho de compressão.

A etapa de pré-processamento é a mesma utilizada no CDA, detalhada na Seção 6.3. Depois de segmentação em GoS, WZSs são colocados em uma linha cada, formando uma matriz bidimensional, onde cada linha representa um WZS, a seguir, transformação DCT em blocos é realizada, para isso, o bloco de transformação direta e inverso de tamanho  $4 \times 4$ , definido pelo padrão de compressão H.264 [53] é utilizado, estes blocos possuem as seguintes formas:

$$DCT = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad (6.16)$$

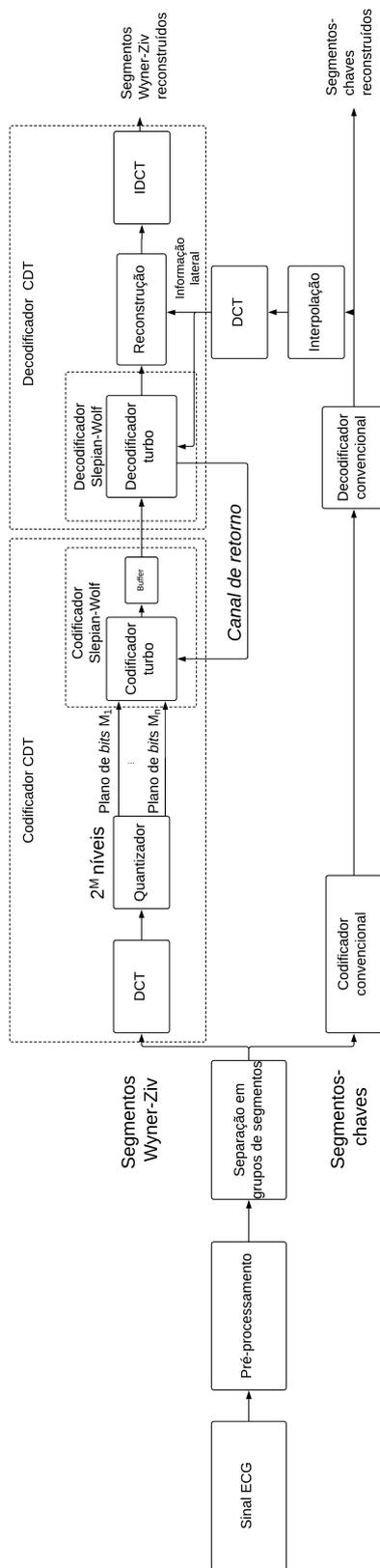


Figura 6.18: Esquema geral do CODEC Wyner-Ziv no domínio da transformada.

e

$$IDCT = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}. \quad (6.17)$$

Contudo, a aplicação da IDCT não restaura a matriz original  $X$  completamente, sendo necessário a aplicação da matriz  $H$  que pode ser incorporada à quantização [53], de tal forma que

$$IDCT H DCT = I \quad (6.18)$$

onde  $I$  representa a matriz identidade. A matriz  $H$  é dada por

$$H = \begin{bmatrix} 1/4 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/5 \end{bmatrix}. \quad (6.19)$$

Uma característica interessante da transformada DCT utilizada que é do tipo inteira, a qual é composta apenas por números inteiros nas matrizes de transformação direta e inversa mostradas nas equações (6.16) e (6.17). Isso permite que sejam realizadas apenas somas e deslocamento de *bits*, este último possibilitado pela multiplicação de amostras por  $\pm 2$ , sem multiplicações mais elaboradas [81]. Após processo de transformação, cada bloco  $4 \times 4$  é quantizado utilizando uma das matrizes de quantização na Figura 6.19.

A Figura 6.21 ilustra o resultado da transformada DCT de um registro de ECG da base de dados utilizada nessa dissertação, onde estão representados valores de amostras antes e depois da transformada DCT. Sinais de ECG possuem componentes de frequência concentrados em baixa frequência, o que fica evidenciado após a transformada DCT ilustrado na Figura 6.21, sendo estes representados por amplitudes altas na primeira posição de cada bloco  $4 \times 4$ , por essa razão, esta parcela é chamada de componente DC, o qual mostra justamente a média de cada sub-bloco e por isso apresenta valores positivos e amplitude alta.

Após transformação e quantização, as 16 bandas de coeficientes  $BC_n$ , ilustradas na Figura 6.20, são agrupadas de acordo com sua posição, de tal forma que o primeiro coeficiente

$$\begin{bmatrix} 32 & 16 & 8 & 8 \\ 16 & 8 & 8 & 4 \\ 8 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 128 & 64 & 32 & 16 \\ 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \end{bmatrix} \quad \begin{bmatrix} 256 & 64 & 32 & 32 \\ 64 & 32 & 32 & 16 \\ 32 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \end{bmatrix}$$

Figura 6.19: Matrizes de quantização.

$BC_0$  contem os maiores valores da transformação sendo chamada de banda de coeficientes DC e as demais de AC.

$BC_0$	$BC_1$	$BC_2$	$BC_3$	
$BC_4$	$BC_5$	$BC_6$	$BC_7$	
$BC_8$	$BC_9$	$BC_{10}$	$BC_{11}$	...
$BC_{12}$	$BC_{13}$	$BC_{14}$	$BC_{15}$	
				⋮

Figura 6.20: Bandas de coeficientes geradas após etapa da transformada.

O procedimento de quantização para as bandas de coeficientes DC são diferentes da AC. Bandas DC possuem valores com amplitude positiva que expressam a energia média de cada bloco  $4 \times 4$ . Dessa forma podem ser então quantizados usando o mesmo quantizador utilizado no CDA (Seção 6.4), que usa a quantização uniforme com  $2^{M_k}$  níveis, sendo  $M_k = 2, 3, 4, 5$  e  $6$ . O intervalo de quantização é dado por

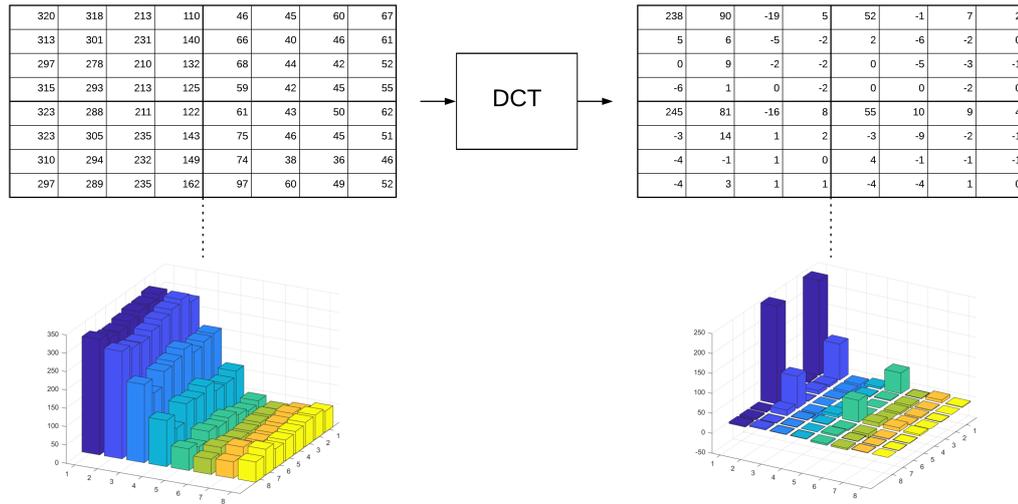


Figura 6.21: Resultados da transformada DCT em 4 sub-blocos de uma matriz  $X_n$  de um registro de ECG utilizado nesta dissertação.

$$W_{DC} = \frac{DC_{max}}{2^{M_k}}, \quad (6.20)$$

onde  $DC_{max}$  é o valor do nível DC máximo da banda DC. As bandas AC, diferente das DC, utilizam a quantização em torno de zero do tipo *midtread* pois possuem valores distribuídos ao redor de zero. O intervalo de quantização desses coeficientes é dado por

$$W_{AC} = \frac{2 AC_{max}}{2^{M_k}}. \quad (6.21)$$

Os valores máximos são enviados como informação auxiliar ao decodificador para que o processo de quantização inverso seja realizado. As matrizes de transformação são conhecidas tanto no codificador quanto no decodificador. Logo, os processos de codificação, decodificação, assim como a reconstrução, são realizados de forma igual ao CDA.

## 6.6 Resultados

Nesta seção os resultados experimentais obtidos com os dois CODECs abordados neste capítulo serão apresentados. Tanto CDA quanto o CDT utilizam técnicas de pré-processamento, assim como o mesmo codificador turbo, sendo diferenciados apenas pela utilização da transformada DCT no segundo. Dois tamanhos de GoS são utilizados: o primeiro de tamanho 3 (GoS-3) e o segundo possui grupos de 5 segmentos (GoS-5). Antes de apresentar tais resultados é im-

portante saber a origem dos dados codificados, ou seja, a base de dados, e também as métricas de desempenho utilizadas, os quais são delineados a seguir.

### 6.6.1 Base de Dados

A base dados utilizada neste trabalho é a base de arritmia do MIT-BIH, que foi o primeiro conjunto de dados disponível para avaliação e detecção de arritmias e, portanto, extremamente utilizado para tal fim, sendo também utilizado em diversas pesquisas envolvendo processamento de sinais, ao redor do mundo [82]. O *Hospital Beth Israel of Boston* (BIH), desde o início, planejava disponibilizar essas gravações para a comunidade científica, a fim de estimular o trabalho neste campo e encorajar avaliações estritamente reprodutíveis e objetivamente comparáveis, de diferentes pesquisas. Quarenta e sete indivíduos foram então selecionados ao acaso e um total de quatro mil registros com 24 horas de duração cada foram criados. Dessas 24 horas selecionadas, algumas tiveram duração diminuída para 10 minutos e outras para 30, e cada registro possui dois canais, ou seja, duas derivações. Como consequência, cada indivíduo apresenta dois registros e, dessa forma, um total de 48 sinais estão disponibilizados.

As pessoas que participaram das gravações de ECG dividem-se em vinte e cinco homens, com idades entre 32 a 89 anos, e vinte e duas mulheres, com idades entre 23 e 89 anos, sendo que aproximadamente 60% dos sujeitos eram pacientes internados [82].

Essas gravações foram digitalizadas com 360 amostras por segundo e por canal, apresentando resolução de 11 *bits* em uma faixa de 10 mV [82]. Dois ou mais cardiologistas anotaram, independentemente, cada registro [82]. Na maioria deles, a derivação DII modificada, conhecida como MLII, foi utilizada em um dos canais, a qual foi obtida pela colocação de eletrodos no tórax, como é a prática padrão para registro ambulatorial de ECG, ao passo que, no outro canal, utilizou-se uma dentre as seguintes derivações:  $V_1$ ,  $V_2$ ,  $V_4$  ou  $V_5$  [82].

### 6.6.2 Métricas de desempenho

A avaliação de desempenho de algoritmos de compressão de ECG é geralmente realizada através de avaliação de eficácia e complexidade computacional. Sendo o primeiro medido pela razão de compressão (*compression ratio* -  $CR$ ) que representa a taxa de compressão de um conjunto de dados, e o erro de reconstrução associado, geralmente calculado através do *percent root mean square difference* ( $PRD$ ), que representa a distorção do sinal. Em resumo,

curvas de *PRD versus CR* são normalmente plotadas e comparadas ao que se obtém com outros esquemas. A complexidade computacional, por sua vez, é a parte prática da implementação de uma metodologia, que geralmente se traduz em tempo de unidade de processamento (*computer processing unit* - CPU), ou também refere-se à quantidade de operações relevantes que um algoritmo executa para finalizar um processo. A *CR* é geralmente calculada através de

$$CR = \frac{b_{original}}{b_{comprimido}}, \quad (6.22)$$

onde  $b_{original}$  é o número de *bits* utilizados na representação original e  $b_{comprimido}$  é o número de *bits* utilizados na representação comprimida. O *PRD*, por sua vez, é dado por

$$PRD = 100 \frac{\sqrt{\sum_{n=n-1}^N (x(n) - \hat{x}(n))^2}}{\sum_{n=n-1}^N (x(n) - \mu)^2}, \quad (6.23)$$

onde  $x(n)$  representa a  $n$ ésima amostra original,  $\hat{x}(n)$  denota a  $n$ ésima amostra recuperada e  $\mu$  é a linha base do conversor analógico para digital.

Alguns autores utilizam como métrica de desempenho a equação (6.23) [25, 73, 83–85]. Entretanto, outros autores [38, 69–71] adotam o *PRD* sem a subtração da linha base, contudo, os mesmos comparam seus resultados com algoritmos que retiram tal linha, portanto, subtende-se que esta linha base é retirada nestes trabalhos antes do cálculo de *PRD*.

### 6.6.3 Resultados experimentais

No CDA o sinal de ECG é codificado unidimensionalmente, onde todos WZSs formam um único vetor, e são quantizados utilizando níveis com valores iguais a 64, 32, 16 e 8, gerando até quatro pontos de distorção, a seguir, amostras quantizadas são separadas em planos de *bits*, sendo que cada plano é codificado separadamente, estes são enviados ao codificador turbo que é composto por dois codificadores convolucionais idênticos e antecidos de entrelaçadores que permitem a separação de rajadas de *bits* errados. O tipo de entrelaçador utilizado é do tipo aleatório. Os *bits* de paridade produzidos por cada codificador são enviados ao perfurador que controla a quantidade enviada ao decodificador. O perfurador cria blocos de *bits* de mesmo tamanho, sendo 32 o tamanho utilizado para todos blocos. A etapa de decodificação é composta por dois decodificadores idênticos em série, sendo que o primeiro recebe *bits* de paridade do primeiro codificador, e o seguinte recebe bits do codificador 2, os quais realizam trocas de informações extrínsecas entre decodificadores. Sendo que ao final da decodificação do segundo

decodificador, considera-se que uma iteração foi realizada, de maneira que um total de 18 iterações são realizadas, onde a cada iteração estimativas de *bits* tornam-se mais confiáveis. Ao término da última iteração é verificado se o erro de *bits* decodificados é menor que 0,001, caso negativo, mais *bits* de paridade são solicitados ao perfurador, pelo canal de retorno. Sendo o erro menor que 0,001, o plano de *bits* é considerado decodificado. O parâmetro  $\alpha$  da exponencial que representa o modelo laplaciano é calculado no codificador e enviado ao decodificador como informação auxiliar, sendo o mesmo utilizado para toda sequência. Na etapa de reconstrução o valor de  $\beta$  foi definido experimentalmente igual  $7/8$  para todos segmentos reconstruídos.

O CDT diferente do CDA codifica o sinal de ECG na forma bidimensional, ou seja, todos WZSs são colocados em uma matriz  $m \times n$ , onde cada linha desta matriz representa um WZS, de forma que  $n$  é múltiplo de 4, pois, a seguir são realizadas transformações DCT em blocos  $4 \times 4$ , A seguir estes blocos são quantizados utilizando até quatro níveis de distorção, possibilitadas por quatro matrizes de quantização (ver Seção 6.5). Depois, bandas de coeficientes  $BC_k$  são codificadas separadamente pelo codificador turbo. O tamanho do bloco de perfuração é o mesmo utilizado no CDA. O parâmetro  $\alpha$  é calculado no codificador para cada  $BC_k$ , e são enviados ao decodificador junto com valores máximos de cada banda utilizadas na quantização. O valor de  $\beta$  na reconstrução foi o mesmo do CDA.

A base de dados utilizada foi a disponibilizada pelo MIT-BIH, como explanado na Seção 6.6.1. Um total de 6 registros dessa base foram utilizados nos experimentos e os mais explorados na literatura foram escolhidos, os quais são: 100, 102, 107, 115, 117 e 119. Apenas os 10 primeiros minutos de cada registros foram codificados. Os resultados estão resumidos na Tabela 6.1, sendo que os resultados desta tabela referem-se apenas a codificação de WZSs. SIs são gerados através de interpolação de KSs reconstruídos perfeitamente, ou seja, são codificados sem distorção, tendo como objetivo a avaliação de compressão somente dos WZSs, conforme trabalhos de DSC [26, 60, 67].

Analisando primeiramente somente os resultados do CDA na Tabela 6.1, comprova-se que a utilização de GoS-3 resulta em resultados superiores ao CDA GoS-5, conforme análise da média de *PRD* e *CR* de ambos na tabela e também nas curvas de desempenho na Figura 6.24, o que já era esperado, pois, conforme comentado na Seção 6.3, tamanhos menores de GoS implicam em predição superior de WZSs, a qual é realizada através de interpolação de KSs, logo resultados de GoS-3 sempre serão superiores a GoS-5, conforme mostrado na Tabela 6.1. Ainda analisando somente resultados do CDA, especificamente o CDA GoS-3, os registros

117 (Figura 6.22) e 119 apresentaram bons resultados, o que é consequência da alta similaridade entre segmentos adjacentes, o que contribui para que o sinal seja bem regular, diferente do registro 107 (Figura 6.22), que não obteve desempenho tão bom, em virtude de menor regularidade entre segmentos adjacentes. Analisando desta vez o CDA GoS-5, os registros 117 e 107 apresentaram respectivamente o melhor e pior desempenho, pelo mesmo motivo da regularidade entre segmentos adjacentes, comentados anteriormente. A baixa regularidade entre segmentos adjacentes do registro 107 é consequência de uma anomalia presente no coração, a qual refere-se a um paciente que possui bloqueio de ramo em um dos ventrículos, o que é evidenciado por longos complexos *QRSs*, conforme será mostrado mais a frente nas figuras de sinais originais e reconstruídos.

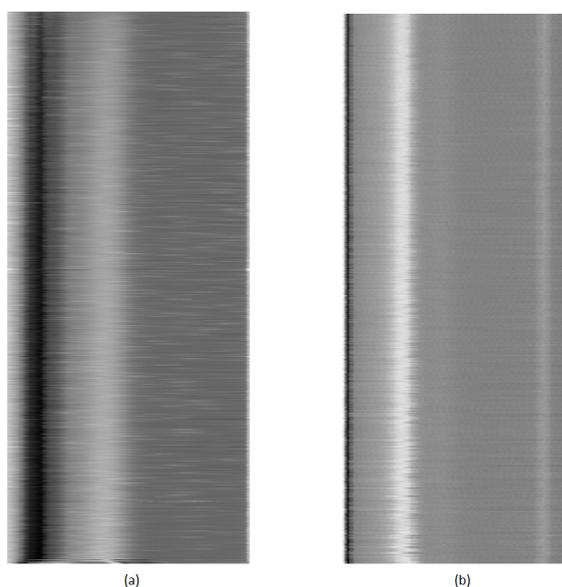


Figura 6.22: Registros após etapa de pré-processamento. (a) registro 107; (b) registro 117.

Comparando desta vez o CDA com algoritmos da literatura, os quais utilizam estrutura DSC, que é o caso do SDE [17], o qual possui estrutura similar ao CDA, ou seja, ambos realizam a codificação no domínio da amostra, porém possuem algumas diferenças, como por exemplo, a forma como *bits* de paridade são utilizados no decodificador, o valor de  $\beta$  na etapa de reconstrução e também há diferença no esquema de perfuração. Comparando-se então os resultados do CDA GoS-3 e SDE GoS-3, constata-se que possuem certa proximidade nos resultados, sendo que o CDA GoS-3 apresenta resultados superiores nos registros 100,102 e 115. Analisando desta vez CDA GoS-5 e SDE-GoS-5, verifica-se que o primeiro apresenta resultados superiores quando codificados os registros 100, 115 e 117, onde mais uma vez os resultados

apresentam-se próximos e em alguns casos superiores à literatura. Por último, comparando com algoritmos que não possuem arquitetura DSC [25, 38, 69, 71], os quais apresentam resultados superiores aos CODECs que utilizam DSC, sendo o algoritmo de Jha e Kolekar [38] o melhor, este que possui estrutura baseada na utilização de transformada wavelet, a qual permite exploração de redundâncias intra-batimentos de sinais de ECG, de forma que componentes de frequência que pouco contribuem para o sinal são descartados.

Portanto, o CDA não alcança taxas superiores à codificadores tradicionais, porém os resultados são satisfatórios. O motivo disso está na falta de ferramentas de exploração intra-batimentos. Apesar disso, o CDA é o CODEC com menor complexidade pois processa diretamente o vetor unidimensional das amostras de sinais de ECG, realizando no codificador apenas somas módulo-2, o que é muito simples. Portanto, dependendo do dispositivo é uma interessante solução, entretanto, caso o dispositivo alvo exija resultados superiores ao apresentado pelo CDA, pode-se utilizar a solução apresentada pelo CDT, o qual tem seus resultados comentados a seguir.

O CDT apresenta resultados superiores ao CDA, conforme mostrado na Tabela 6.1, em razão de melhor exploração de redundâncias intra-batimentos, permitido pela transformada DCT, a qual é ideal para sinais de ECG, pois estes apresentam grande quantidade de energia concentrada em baixas frequências conforme ilustrado na Figura 6.23, e essa característica é ideal para utilização da transformada DCT. Ressaltando que a transformada utilizada é do tipo inteira, que realiza apenas somas e deslocamento de *bits*, o que é proporcionado pela matriz de transformação em blocos do CODEC H.264 (ver Seção 6.5). Primeiramente, analisando somente resultados do CDT GoS-3, o qual mostra resultados bons para todos registros, inclusive até superiores quando comparado ao seu equivalente TDE GoS-3, nos casos dos registros 115 e 117, e nos demais, os resultados são bem próximos. Por sua vez, quando utilizado GoS-5 os resultados são inferiores ao CDT-GoS-3, o que já era esperado, conforme comentado anteriormente, GoS-3 sempre apresenta resultados superiores ao GoS-5, contudo, o GoS-5 tem bons resultados. Apesar de não apresentar desempenho similar à algoritmos que usam métodos tradicionais como Jha e Kolekar [38], o CDT representa uma ótima solução quando uma boa relação taxa-distorção é importante, de maneira que o dispositivo alvo suporte o pequeno acréscimo que é a etapa de transformada DCT.

Outra importante análise são as curvas de desempenho ( $CR \times PRD$ ) dos CODECs implementados, sendo ilustradas nas Figuras 6.24, 6.25 e 6.26, que mostram o desempenho dos

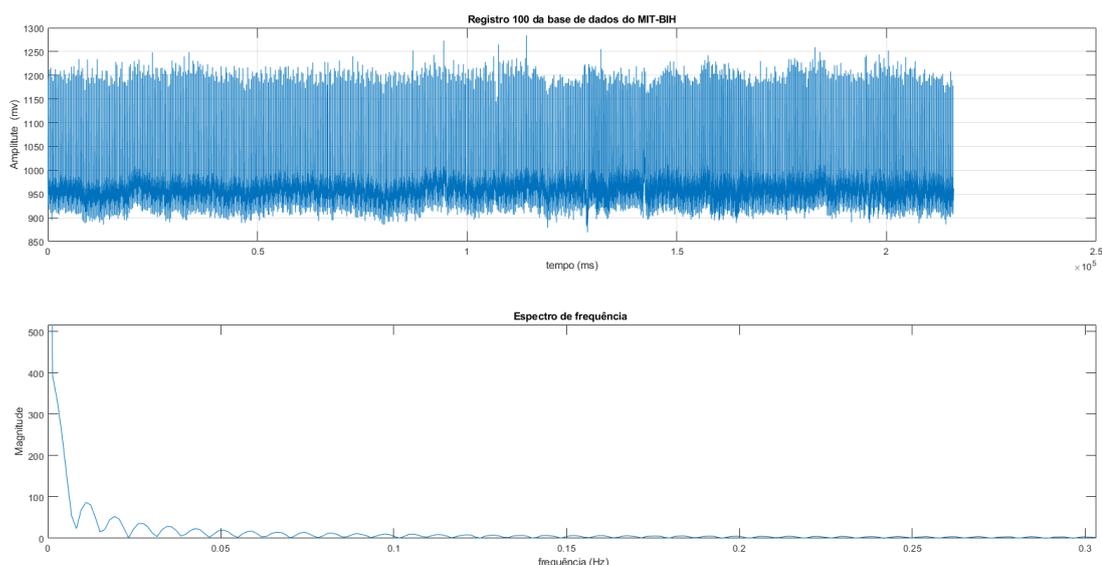


Figura 6.23: Espectro de frequência do registro 100 da base de dados do MIT-BIH.

Tabela 6.1: Comparação de desempenho ( $PRD \times CR$ ) entre os algoritmos CDA, CDT, e algoritmos da literatura [17, 38, 69–71].

Algoritmo	100		102		107		115		117		119		Média CR	Média PRD
	CR	PRD	CR	PRD	CR	PRD	CR	PRD	CR	PRD	CR	PRD		
Jha [38]	39,90	0,23	27,90	0,50	39,60	0,66	46,80	0,28	47,00	0,31	38,50	0,47	39,95	0,41
Tun [69]	10,79	0,42	10,04	0,35	9,82	0,19	-	-	-	-	-	-	-	-
Lee [70]	22,94	1,95	25,91	1,39	18,55	3,93	19,88	2,98	24,43	1,17	19,31	2,05	21,84	2,24
Fathi [71]	31,40	3,20	27,50	2,20	-	-	-	-	29,30	3,00	-	-	-	-
MMP EqDC [17]	24,00	3,30	-	-	10,70	1,70	24,00	1,26	13,00	0,91	-	-	-	-
MMP EqDC/SS [17]	-	-	29,49	2,49	-	-	30,60	2,92	-	-	20,90	1,93	-	-
SDE GoS-3 [17]	26,33	11,27	26,76	10,18	13,64	6,66	26,87	14,48	21,57	2,17	21,23	3,66	22,73	8,07
SDE GoS-5 [17]	25,01	14,21	24,82	13,16	13,53	12,68	27,98	18,14	16,82	2,65	20,32	4,55	21,41	10,90
TDE GoS-3 [17]	29,26	3,81	28,45	5,28	9,57	2,77	36,41	4,19	29,58	1,62	20,35	1,81	25,60	3,25
TDE GoS-5 [17]	23,23	4,06	27,35	6,47	9,66	4,44	39,27	4,64	27,22	2,06	29,84	2,66	26,10	4,06
CDA GoS-3	24,00	6,95	20,76	6,50	15,42	7,14	24,02	6,00	21,50	3,20	22,43	4,00	21,36	5,63
CDA GoS-5	24,00	9,01	23,57	14,99	13,72	13,22	22,00	8,17	19,00	2,20	21,56	5,01	20,64	8,77
CDT GoS-3	22,00	3,35	28,93	5,50	7,83	3,95	39,60	3,50	30,85	1,49	19,00	2,84	24,70	3,44
CDT GoS-5	26,52	4,50	27,35	6,00	11,18	5,20	34,72	4,20	25,05	1,75	21,00	2,94	24,30	4,10

registros 100, 102, 107, 115 e 119, as quais mostram a variação da distorção ( $PRD$ ) em razão do aumento da taxa de compressão ( $CR$ ). Nestes gráficos observa-se o desempenho superior do CDT comparado ao CDA mostrado na Figura 6.26, sendo que o melhor foi justamente o CDT GoS-3, pois utiliza a transformada DCT que possibilita a exploração de redundâncias intratamentos. Além disso, o desempenho superior do GoS-3 é nítido comparado ao GoS-5, isto ocorre devido a distorção adicionada pela interpolação dos segmentos de grupos maiores, pois, quanto maior o GoS maior a distorção. Um fato interessante nessas figuras é que o CODEC CDA apresenta distorção alta quando submetida a taxas elevadas de compressão, mais uma vez, isso ocorre devido a falta de ferramentas que permitam a exploração de redundâncias intratamentos, o que pode ser proporcionado pela transformada DCT e evidenciado pelas curvas

do CDT. Os registros 117 e 119 (Figuras 6.24 e 6.25) apresentam pouca variação de *PRD* para taxas elevadas de *CR*, o motivo está na boa regularidade de segmentos adjacentes apresentadas por esses registros.

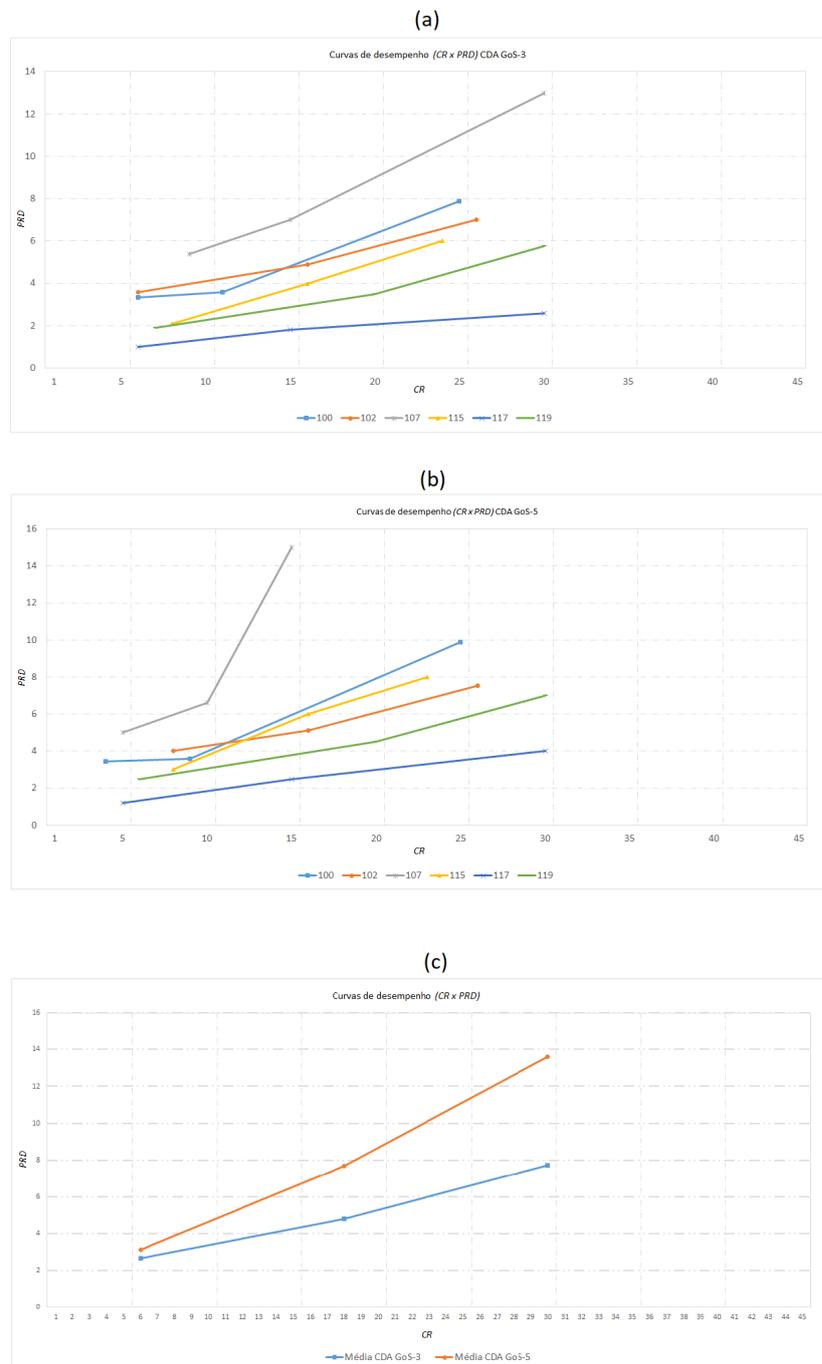
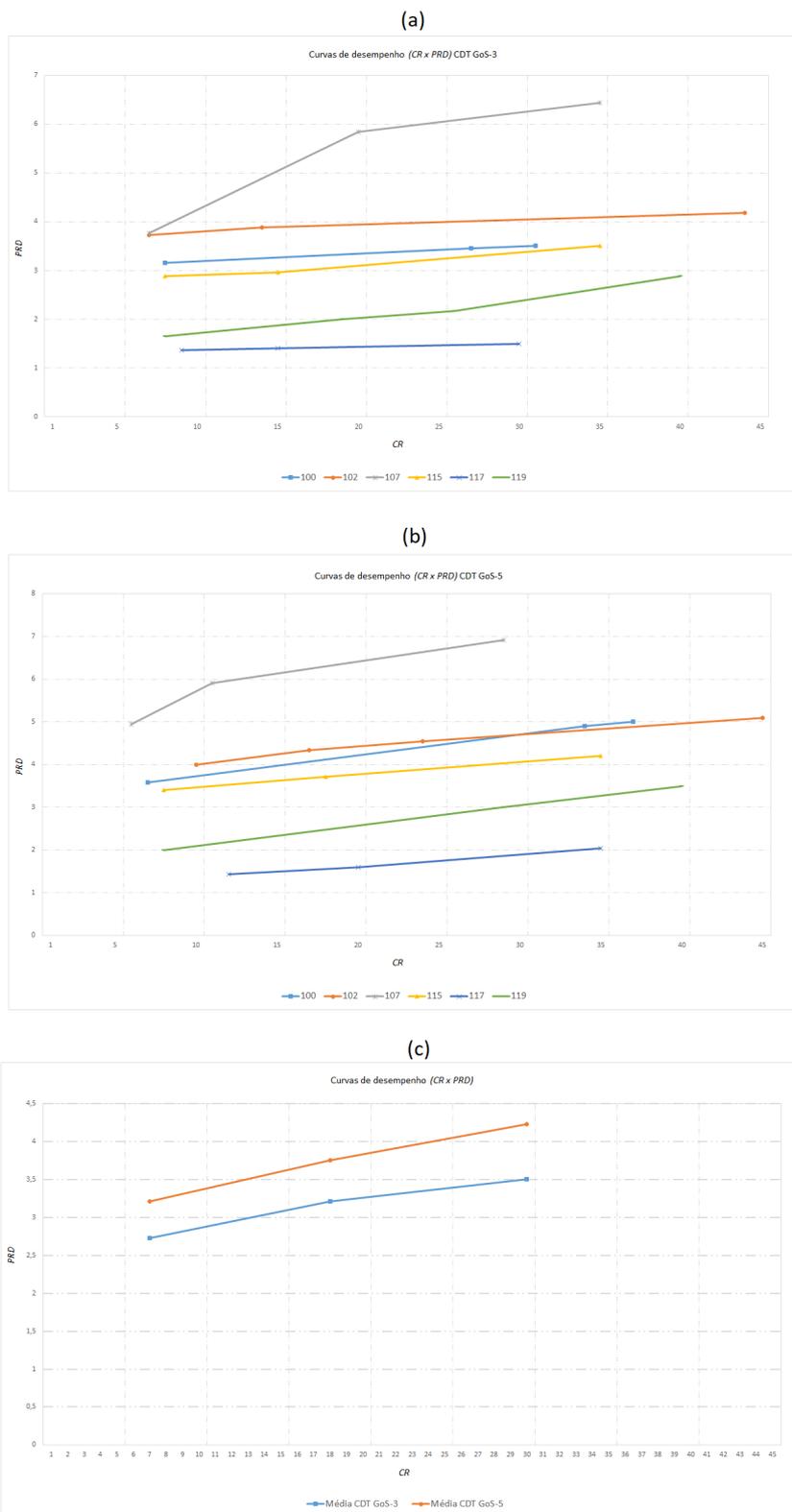


Figura 6.24: Curvas de desempenho; (a) CDA GoS-3; (b) CDA GoS-5; (c) Média das curvas de CDA GoS-3 e GoS-5.

As Figuras 6.27 até 6.34 mostram sinais originais de ECG (Registros 100 e 107) durante



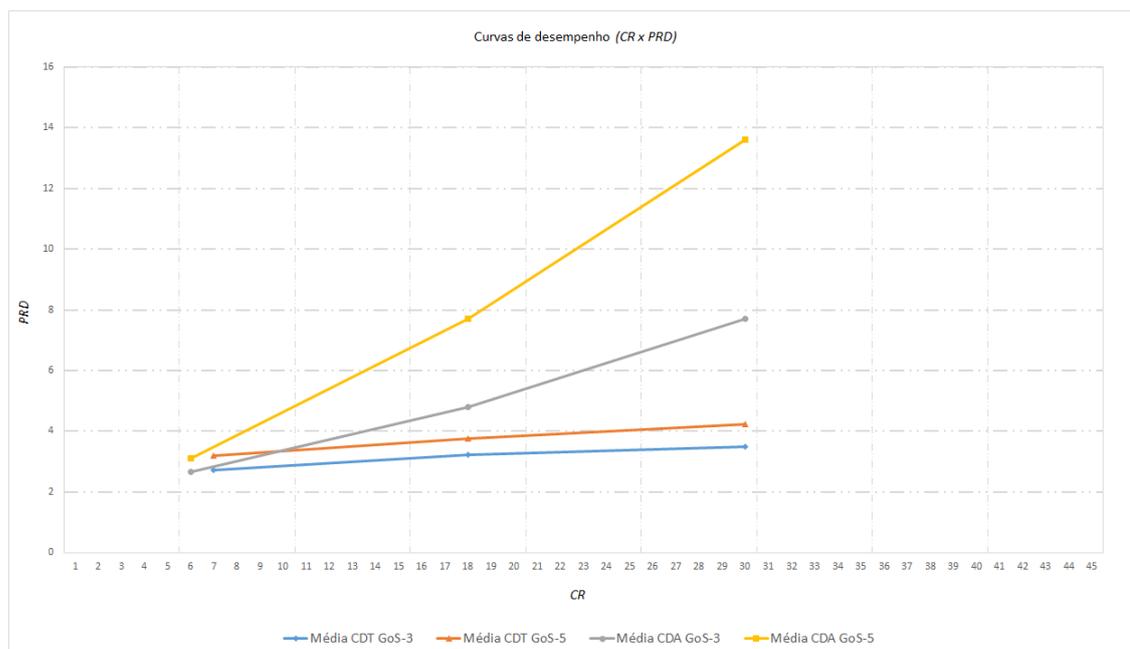


Figura 6.26: Desempenho de compressão de CDA e CDT.

os primeiros 10 segundos, onde também são mostrados estes registros comprimidos utilizando os CODECs CDA e CDT com GoS-3 e GoS-5. Como mostrado na Tabela 6.1 ambos CODECs apresentam bons resultados de *PRD*. Contudo, as distorções de sinais de ECG reconstruídos precisam ser verificados utilizando conceitos apresentados no Capítulo 3. A primeira Figura 6.27 apresenta distorção moderada nas ondas P e T, logo são adequadas para diagnóstico. A Figura 6.28 apresenta  $PRD = 4.74$ , sendo este um valor levemente alto, contudo, aceitável. Isso acontece devido ao registro 107 não apresentar boa regularidade, o que afeta o desempenho de compressão, elevando assim a distorção.

O registro 107 representado nas Figuras 6.28, 6.30, 6.32 e 6.34, é de um paciente que apresenta bloqueio de ramo em um dos ventrículos, o que é apresentado por sequências longas de complexos *QRSs*, sendo este alongamento mantido no sinal comprimido. O registro 100 apresenta um bom resultado, como mostrado pela Figura 6.31.

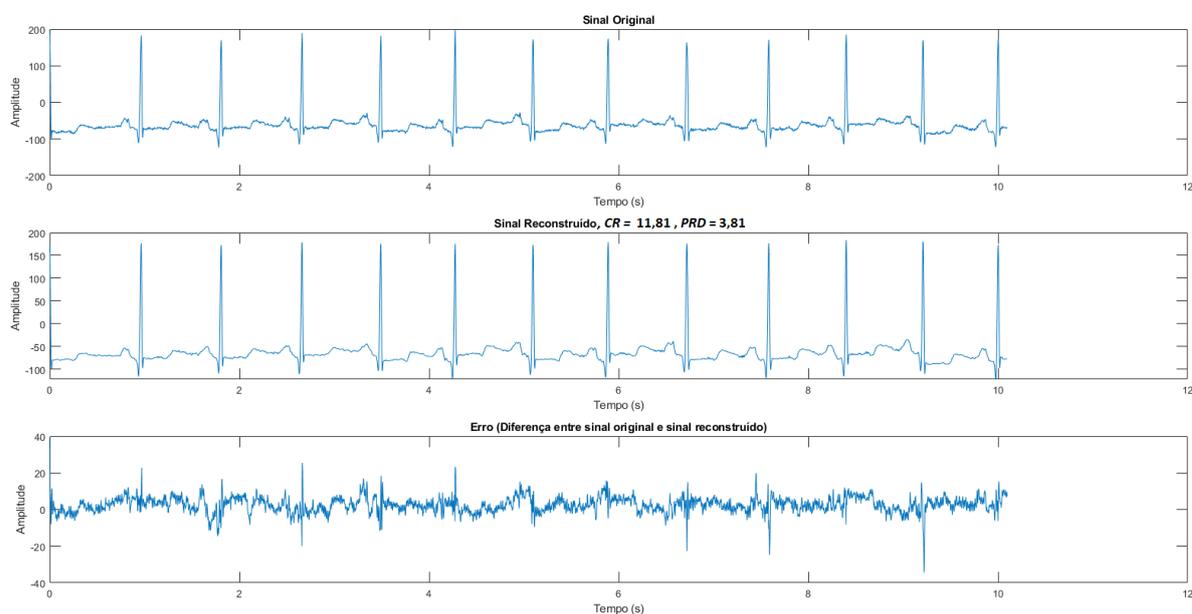


Figura 6.27: Registro 100 reconstruído com CDA GoS-3 com  $CR = 11,81$  e  $PRD = 3,81$  .

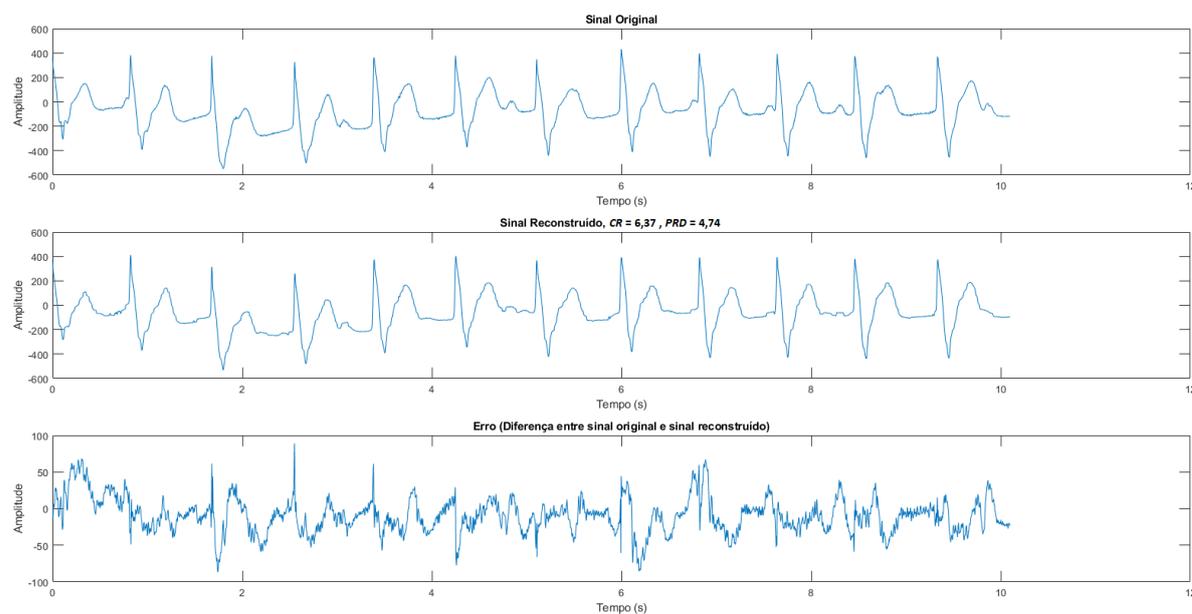


Figura 6.28: Registro 107 reconstruído com CDA GoS-3 com  $CR = 6,37$  e  $PRD = 4,74$ .

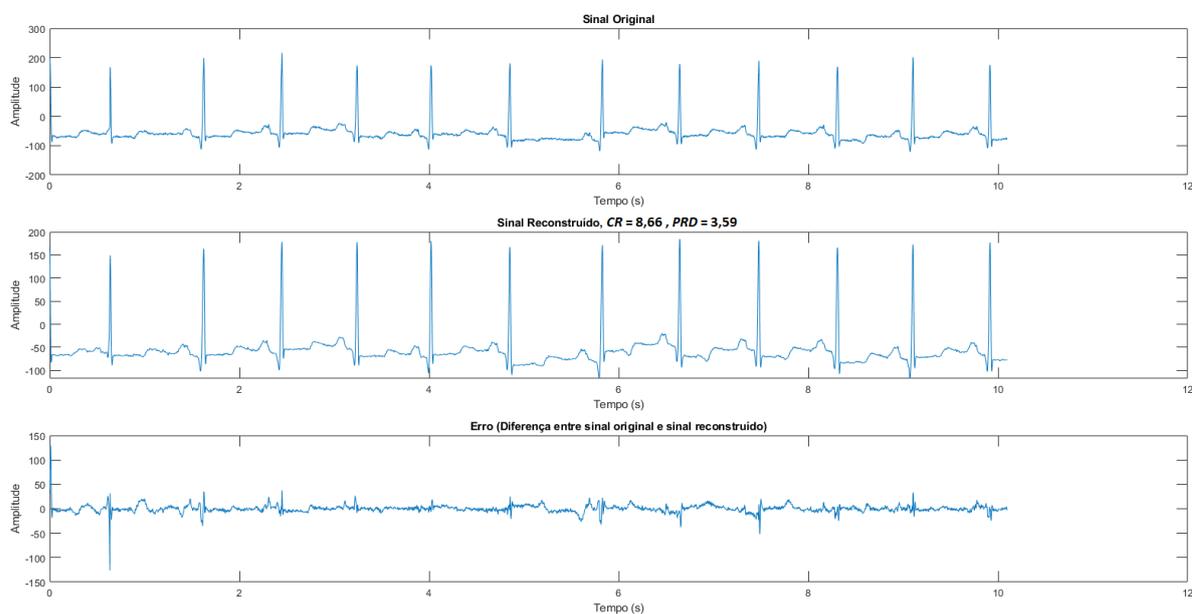


Figura 6.29: Registro 100 reconstruído com CDA GoS-5 com  $CR = 8,66$  e  $PRD = 3,59$ .

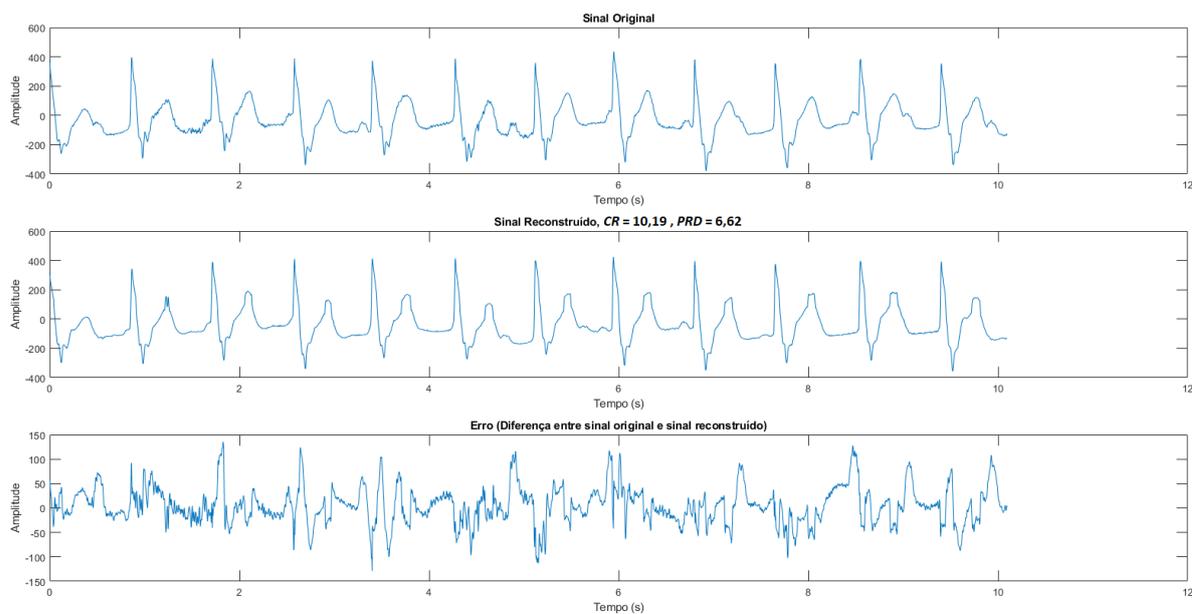


Figura 6.30: Registro 107 reconstruído com CDA GoS-5 com  $CR = 10,19$  e  $PRD = 6,62$ .

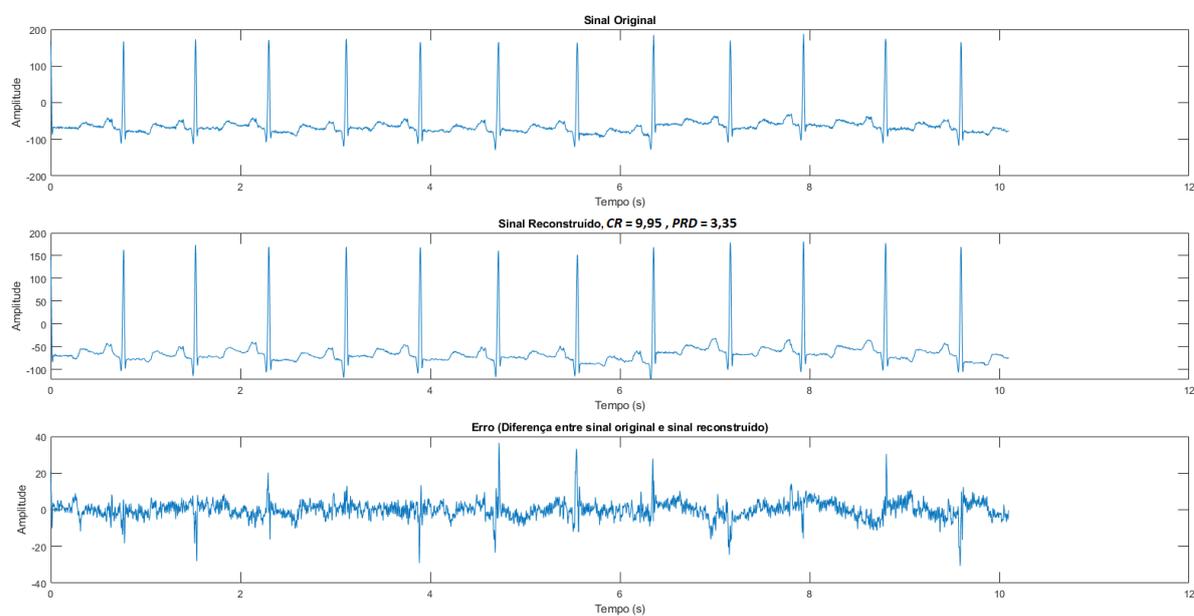


Figura 6.31: Registro 100 reconstruído com CDT GoS-3 com  $CR = 9,95$  e  $PRD = 3,35$ .

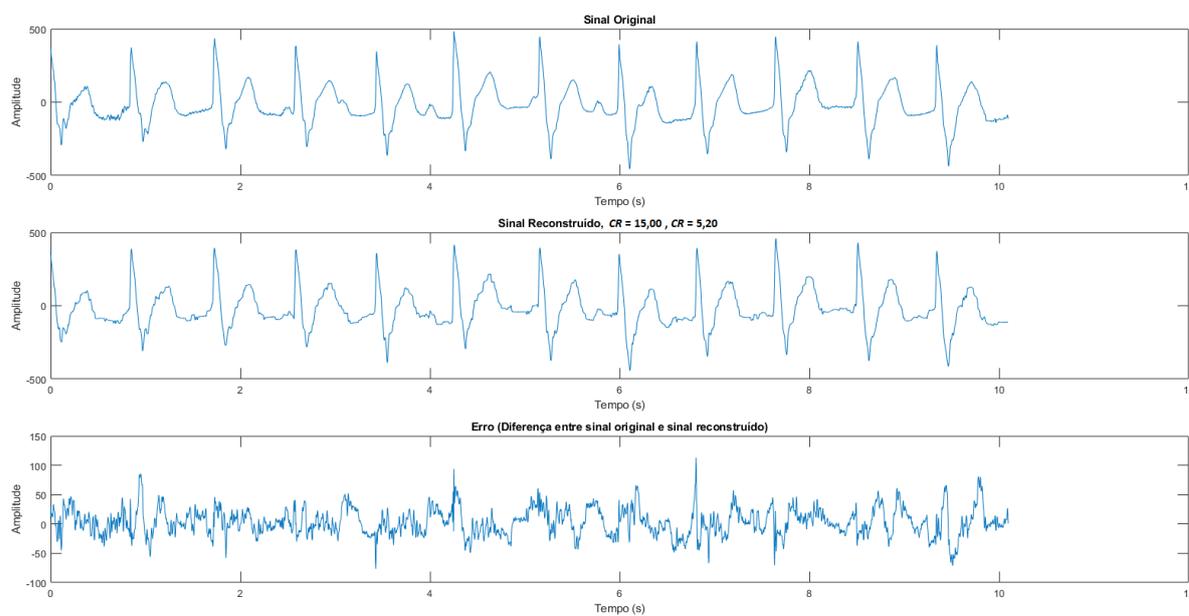


Figura 6.32: Registro 107 reconstruído com CDT GoS-3 com  $CR = 15,00$  e  $PRD = 5,20$ .

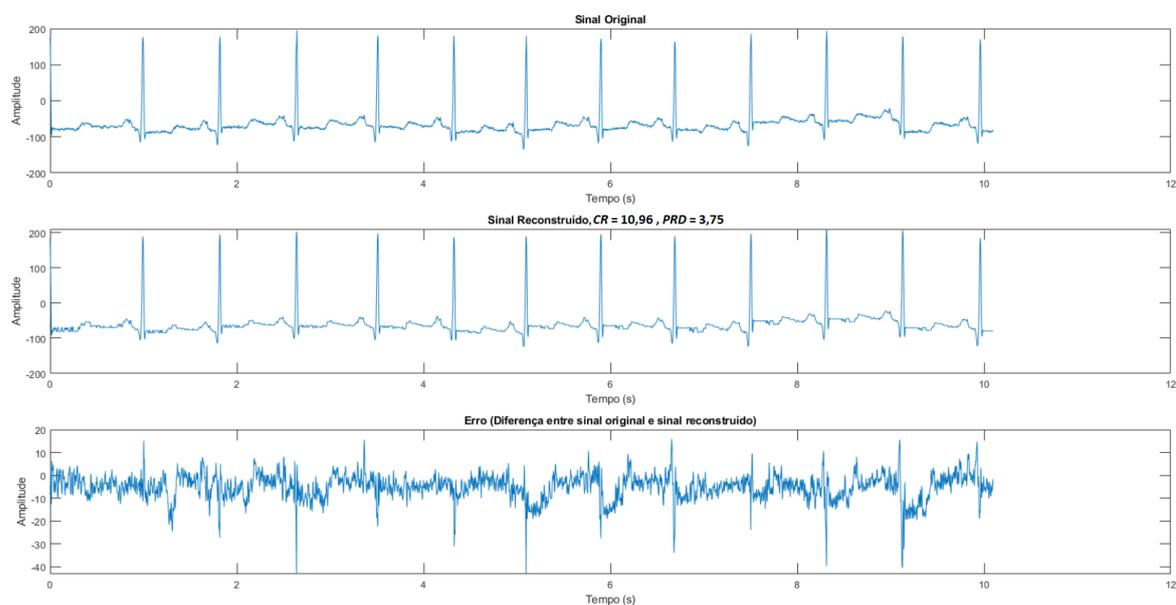


Figura 6.33: Registro 100 reconstruído com CDT GoS-5 com  $CR = 10,96$  e  $PRD = 3,75$ .

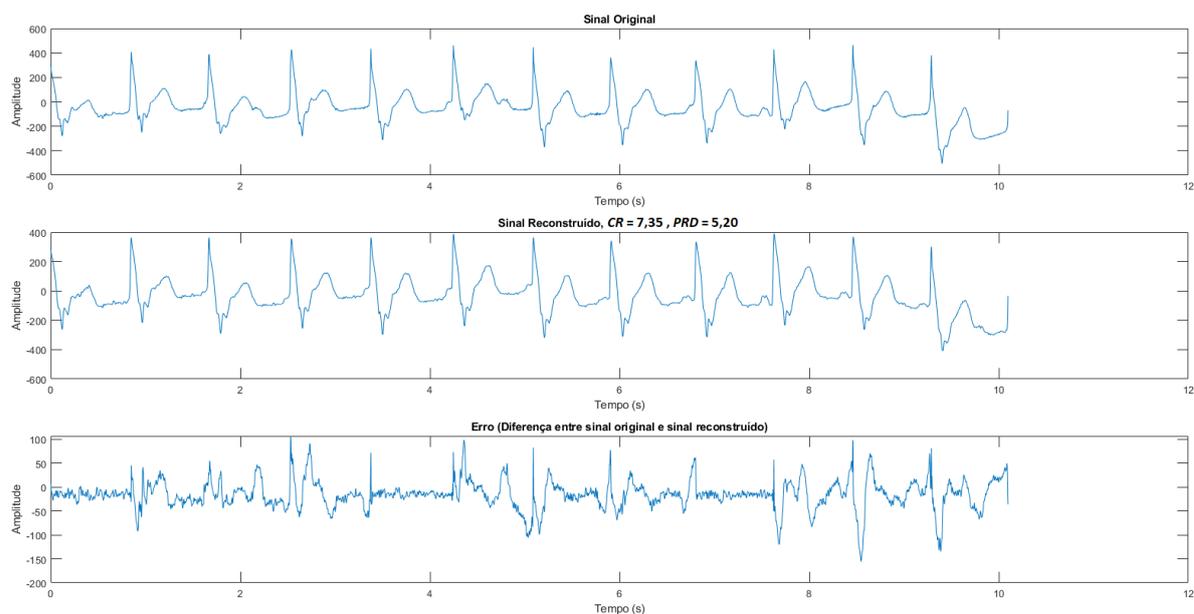


Figura 6.34: Registro 107 reconstruído com CDT GoS-5 com  $CR = 7,35$  e  $PRD = 5,20$ .

### 6.6.4 Discussão e análise de resultados

Na seção anterior, os resultados do CDA e CDT foram vistos através de métricas de *PRD* e *CR*, onde o primeiro mostra a distorção do sinal e o segundo refere-se à taxa de compressão, os quais são amplamente utilizados pela literatura para fins de análise de desempenho. Além disso, uma terceira métrica é em relação a complexidade do algoritmo, que pode ser expressa pela quantidade de operações relevantes realizadas pelo algoritmo, a qual pode ser mostrada como uma função de  $n$  que refere-se ao tempo de execução de operações relevantes em função do tamanho da entrada [86], onde  $n$  representa a quantidade de elementos processados, o que pode ser desde uma função simples como por exemplo uma soma simples, que tem um tempo de processamento contante representado por  $O(1)$ , até mesmo operações mais elaboradas que podem ser expressas em notação de  $O(f)$ , onde  $f$  pode representar operações quadráticas, cúbicas, logarítmicas, etc. Mais detalhes sobre a notação  $O(f)$  podem ser vistas no artigo de Danziger [86].

Em relação à complexidade, algumas comparações são válidas como no caso do CDA que realiza operações lógicas do tipo soma módulo-2 no codificador, que possui a complexidade da ordem  $O(n)$ , o que significa que esse processo tem uma resposta linear no tempo para uma quantidade de  $n$  bits inseridos no codificador, da mesma forma o quantizador utilizado no CDA e CDT possuem complexidade igual a  $O(n)$ . Isso significa que o CDA tem complexidade simples e menor que, por exemplo, a complexidade do algoritmo de Lee Kim e Lee [38], o qual possui diferentes etapas de codificação, e uma delas é a utilização da transformada DCT real similar a utilizada no JPEG [87], a qual tem complexidade de ordem  $(k8)^4$ , o que representa uma função polinomial, que de acordo com a quantidade de elementos processados necessita de uma quantidade de tempo crescente de forma polinomial, conforme mostrado na Tabela (6.2), onde  $k$  representa a quantidade de blocos  $8 \times 8$ . Além disso, Lee Kim e Lee [70] utilizam um codificador de entropia do tipo *Huffman* que precisa montar e fazer busca em dicionário e possui complexidade da ordem de  $O(n \log(n))$  (Tabela (6.2)). Diferente de Lee Kim e Lee, a transformada DCT utilizada nesta dissertação é do tipo inteira que possui menor complexidade, a qual realiza apenas 96 somas e 32 deslocamento de bits para cada bloco de tamanho  $4 \times 4$ , o que em notação de  $O(f)$  pode ser expressa como uma função linear  $O(n)$ , pois são operações simples como deslocamento de bits e somas, conforme a Tabela (6.3). Os algoritmos de Jha e Kolekar [38], Tun, Moe e Nang [69] e também Fathi e Faraji-Kheirabadi [71] possuem etapa de trans-

Tabela 6.2: Etapas relevantes de algoritmos da literatura [38, 69–71] e seus valores de complexidade em notação  $O(f)$ .

DWT (matriz $m \times n$ )	DCT (blocos $8 \times 8$ )	Quantização	Codificação <i>Huffman</i>	Codificação aritmética
$O(4mn)$	$(k8)^4$	$O(n)$	$O(n \log(n))$	$O(n \log(n+a))Sn$

Tabela 6.3: Etapas do codificador do CDA e CDT e seus valores de complexidades em notação  $O(f)$ .

	DCT inteira (blocos $4 \times 4$ )		Quantização	Codificador convolucional
CDA - Complexidade	-		$O(n)$	Somas módulo-2
				$O(n)$
CDT - Complexidade	Deslocamento de <i>bits</i>	Somas	$O(n)$	Somas módulo-2
	$O(n)$			$O(n)$

formada discreta wavellet ( *Discrete Wavellet Transform* - DWT), a qual possui complexidade  $O(4mn)$  [88], que é inferior a complexidade da DCT real, porém tem complexidade maior que a DCT utilizada nesta dissertação que é do tipo inteira.

Resumidamente, conforme mostrado nas Tabelas (6.2) e (6.3) o codificador turbo possui menor complexidade em notação  $O(f)$  comparado à algoritmos da literatura, o que pode ser visto nas operações relevantes realizadas como a DCT inteira, quantização e somas módulo-2. Sendo tais operações, realizadas pelo CDA e CDT, menos complexas que operações como a DWT, DCT, codificação *Huffman* e codificação aritmética. Portanto, menor complexidade significa a exigência de menor capacidade de poder computacional do hardware presente no codificador, o que é ideal para dispositivos embarcados, como, por exemplo, o *Holter* de ECG.

Em relação a complexidade do decodificador, esta possui alta complexidade, pois, realiza cálculos com pontos flutuantes de forma iterativa, ou seja, inúmeros cálculos de probabilidades de forma repetitiva, o que exige alto poder computacional.

Como mostrado nas análises na seção anterior, os CODECs CDA e CDT mantêm a maior parte da informação utilizada em diagnósticos. Contudo, taxas de compressão menores que 10 : 1 apresentam maior precisão, o que não impede que sejam utilizadas taxas maiores, desde que o valor de *PRD* esteja dentro de valores aceitáveis, que normalmente é abaixo de 7, pois, *PRDs* abaixo de 7 permitem a permanência de informações importantes necessárias à análise médica, conforme a tese de Lima Filho [17].

Concluindo este capítulo, os CODECs implementados mostraram que a compressão de ECG, utilizando técnicas de DSC, é possível, o que foi mostrado nos resultados apresentados nesta seção. Contudo, o CODEC CDT, especificamente, pode apresentar resultados ainda me-

---

lhores com uma codificação diferente do que foi apresentado, o que foi possível através de codificação residual de ECG. Sendo assim, a metodologia que emprega a codificação residual utilizando o CDT é apresentada no próximo capítulo.

# Capítulo 7

## Utilizando resíduo de ECG

No capítulo anterior, o CDA e CDT foram apresentados, tendo estes a finalidade de realizar a codificação de sinais de ECG em sua forma completa. Contudo, existem metodologias de codificação distribuída que realizam a codificação residual de vídeo, as quais alcançam resultados superiores à codificação de vídeo completo, sendo o trabalho de Aaron, Varodayan e Girod [89] um interessante exemplo de codificação residual que mostra como o sinal residual de vídeo pode ser empregado na arquitetura DSC, onde a arquitetura DSC no domínio da amostra foi utilizada. Portanto, o esquema de DSC residual não se restringe somente a vídeo, pois pode ser também utilizado em sinais de ECG, o que será mostrado nesse capítulo através de uma técnica de pré-processamento residual, utilizando o CDT como base. A metodologia de DSC residual implementada é baseada no CDT, o que não impede que seja implementada no CDA em trabalhos futuros.

O processamento residual permite apenas que informações relevantes de WZSs (segmentos Wyner-Ziv) sejam codificadas, para que isso aconteça resíduos de WZSs e SIs (informação lateral) são gerados tanto no codificador quanto no decodificador. A informação residual gerada é inserida no CODEC base utilizado que é o CDT, sendo este CODEC o mesmo apresentado no capítulo anterior, apenas diferenciado pela informação a ser codificada que é o resíduo de ECG. A geração do resíduo acontece antes da aplicação da transformada DCT em WZSs e SIs, conforme ilustrado na Figura 7.1, onde  $\bar{s}_i$  designa a média de KSs (segmentos-chaves) interpolados.

Conforme ilustrado na Figura 7.1, o  $WZSs_{res}$  é gerado no lado do codificador, e para isso, a média de SIs é utilizada, ou seja, o codificador tem conhecimento de SIs, o que é re-

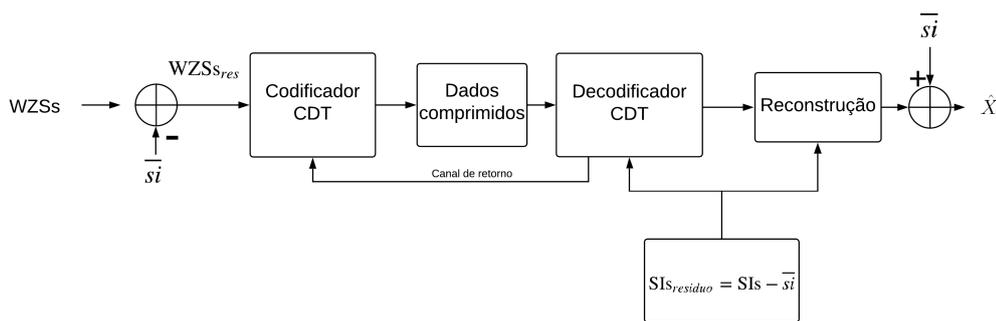


Figura 7.1: Codificação de resíduo de ECG.

sultado de interpolações de KSs previamente codificados e decodificados. Isso implica em pequeno acréscimo computacional ao codificador, por outro lado, oferece desempenho superior a codificação do sinal de ECG completo, o que será mostrado nos resultados. A metodologia de pré-processamento residual, assim como toda sua estrutura, é detalhada na seção seguinte.

## 7.1 Metodologia

Como visto na Seção 6.5, cada WZS é colocado em uma linha de uma matriz  $WZS_{m \times n}$ , a partir daí o resíduo é gerado, de forma que uma primeira técnica foi experimentada, a qual forma o resíduo a partir da subtração da média de WZSs de cada WZS, depois envia-se a média de WZSs como informação auxiliar ao decodificador, de forma que seja somada ao resíduo reconstruído. Contudo, esta técnica não obteve resultados satisfatórios, pois informações auxiliares são enviadas ao decodificador, o que compromete o desempenho de compressão, logo, uma segunda técnica foi utilizada, ilustrada na Figura 7.2, que é a utilização do resíduo  $WZS_{res}$  gerado a partir da subtração de cada linha  $WZS_{m \times n}$  da média de SIs representada por  $\bar{s}_i$ , e da mesma forma no decodificador SIs residuais são gerados. Dessa forma, esta última técnica, quando aplicada, oferece desempenho melhor que a primeira apresentada. Além disso, a necessidade de enviar informação auxiliar ao decodificador foi eliminada nesta segunda técnica.

A geração residual de WZSs é realizada após o pré-processamento, que é constituído das etapas de: detecção de períodos; normalização; equalização DC e ordenação por similaridade. A seguir, a separação em GoS é realizada, sendo utilizado GoS-3 e GoS-5. Depois, WZSs são colocados em uma linha de uma matriz, conforme a equação (7.2), formando uma matriz  $WZS_{m \times n}$ , onde cada linha dessa matriz corresponde a um WZS, os quais são subtraídos de  $\bar{s}_i$

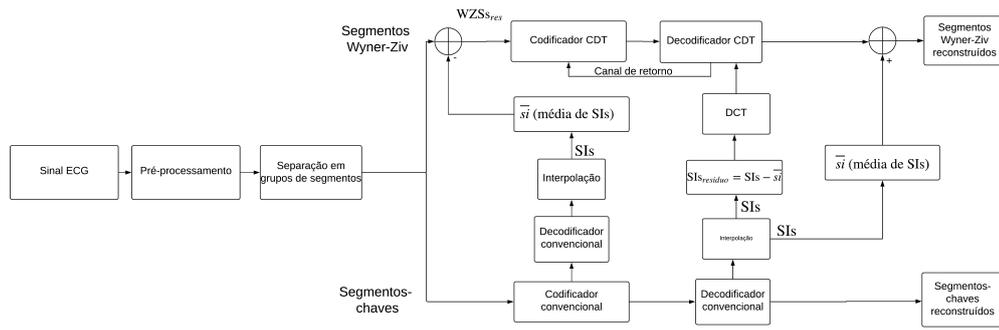


Figura 7.2: Esquema do CODEC DCT com utilização do resíduo de ECG.

que corresponde a média de SIs, gerando assim o  $WZSs_{res}$ , conforme delineado a seguir

$$WZSs = [x_1 \ x_2 \ x_3 \ \dots \ x_n]^T, \quad (7.1)$$

onde  $x_n$  representa um período de ECG e pode ser expandido conforme a matriz a seguir

$$WZSs_{m \times n} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}, \quad (7.2)$$

onde cada linha de  $WZSs$  representa um WZS. O cálculo da média de SIs é dado a seguir

$$\bar{s}i = \frac{1}{n} \sum_1^n s_n, \quad (7.3)$$

sendo  $s_n$  vetores que representam linhas da matriz de SIs,  $\bar{s}i$  representa o vetor com a média de SIs. Agora pode-se calcular o resíduo através da equação

$$WZSs_{res}(n) = x_n - \bar{s}i, \quad (7.4)$$

onde  $x_n$  representa o  $n$ ésimo WZS e o segundo termo é a média de SIs. A forma do sinal residual do registro 117 é mostrada na Figura 7.3, a qual mostra que o resíduo possui amplitude máxima em torno de 150, enquanto, o sinal completo possui amplitude em torno de 400, o que é muito bom, pois uma quantidade menor de informação é codificada.

A geração residual de SIs ocorre de forma similar, sendo esta muito importante, pois, conforme explanado na Seção 6.4.5, refere-se à predição de WZSs, que auxilia tanto na decodificação quanto na reconstrução de WZSs, portanto, quanto maior a similaridade entre WZSs e

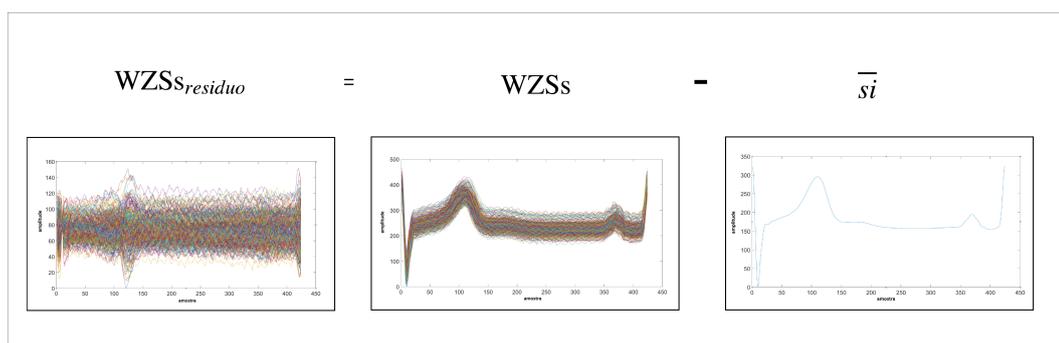


Figura 7.3: Sinal residual do registro 117 gerado a partir da equação (7.4).

sua predição, melhor é o desempenho de compressão, conforme teoria de codificação distribuída de Slepian-Wolf, desta forma, é necessário realizar a geração de SI residual.

Resíduos de WZSs são, então, enviados para etapa de transformada DCT em blocos de tamanhos  $4 \times 4$ , onde são geradas 16 bandas de coeficientes após transformação e quantização, conforme explanado no Capítulo 6.5, essas bandas estão ilustradas na Figura 7.4. Uma importante observação, em relação a bandas de sinais de ECG residuais, é o histograma de cada banda que apresenta todas as bandas com valores em torno de zero conforme a Figura 7.5, diferente do que acontece no CDT base que tem na banda  $BC_0$  concentração da média de todas as bandas e por isso apresenta amplitudes sempre positivas (ver Figura 7.6). Isso é consequência da diminuição da amplitude do sinal de ECG, afetando assim o nível DC, portanto, a quantização empregada para este e demais coeficientes é em torno de zero do tipo (*midtread*).

O procedimento de transformada DCT, separação em bandas de coeficientes, quantização e codificação após geração de resíduos, assim como a reconstrução feita no decodificador Wyner-Ziv seguem a mesma metodologia empregada no CDT (Seções 6.4.6 e 6.4.7). A diferença está apenas na utilização do resíduo do ECG em vez do sinal de ECG completo. Sendo assim, resultados desta técnica de pré-processamento residual são mostradas a seguir.

## 7.2 Resultados

Esta seção apresenta resultados da codificação residual de ECG utilizando o CDT, o qual é chamado de CDTR. Para melhor análise de resultados a sequência de testes é separada em três etapas. Primeiramente, a comparação de resultados com algoritmos que utilizam técnicas tradicionais da literatura [17, 38, 69–71] é apresentada, e também com algoritmos que possuem

$BC_0$	$BC_1$	$BC_2$	$BC_3$	
$BC_4$	$BC_5$	$BC_6$	$BC_7$	
$BC_8$	$BC_9$	$BC_{10}$	$BC_{11}$	...
$BC_{12}$	$BC_{13}$	$BC_{14}$	$BC_{15}$	

⋮

Figura 7.4: Blocos de bandas de coeficientes geradas pela transformada DCT.

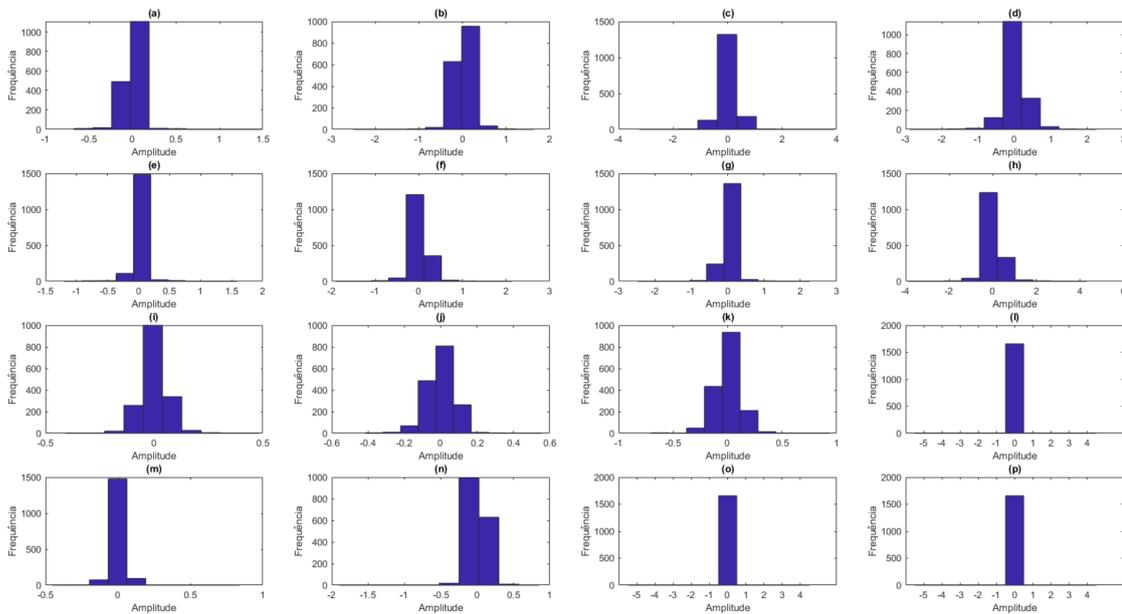


Figura 7.5: Resíduo - Histograma de bandas de coeficientes BC: (a)  $BC_0$ , (b)  $BC_1$ , (c)  $BC_2$ , (d)  $BC_3$ , ..., (p)  $BC_{15}$ .

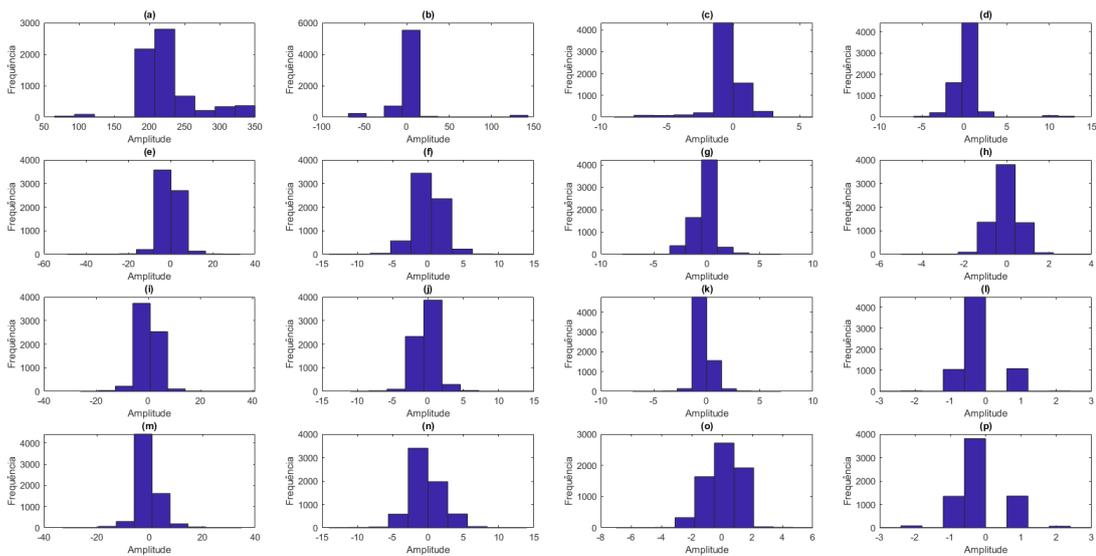


Figura 7.6: Sinal ECG completo - Histograma de bandas de coeficientes BC: (a)  $BC_0$ , (b)  $BC_1$ , (c)  $BC_2$ , (d)  $BC_3$ , ..., (p)  $BC_{15}$ .

esquema DSC [17]. A seguir, a curva que representa a taxa-distorção de registros utilizados é mostrada, por fim, figuras reconstruídas de sinais de ECG comparadas ao sinal original, assim como o erro de reconstrução são mostradas.

A base de dados utilizada é do MIT-BIH, da qual os registros 100, 102, 107, 115, 117 e 119 foram codificados. A medida objetiva de distorção utilizada é o *PRD*, e a medida de compressão é o *CR*.

Inicialmente, a etapa de pré-processamento é realizada, onde os 10 primeiros minutos de cada registro são codificados. Nesta etapa inicial os períodos de sinais de ECG são detectados, normalizados, equalizados e organizados por ordem de similaridade, conforme Seção 6.3, depois, segmentos pré-processados são separados em GoS, sendo utilizado GoS-3 e GoS-5, assim então, WZSs estão prontos para passar pelo processo de geração residual, que acontece conforme a equação (7.4), e da mesma forma resíduos de SIs são gerados.

Os resultados da codificação residual estão na Tabela 7.1. Primeiramente, analisando resultados do CDTR GoS-3, o qual apresenta resultados superiores para todos registros quando comparado ao CDT GoS-3, sendo o registro 117 o que apresenta melhor desempenho, em virtude da amplitude máxima do resíduo deste registro diminuir de aproximadamente 400 para em torno de 150, além do resíduo manter a regularidade de segmentos adjacentes conforme foi mostrado na Figura 7.8, o que permite a codificação de menor quantidade de informação, diferente do registro 107 que apresenta redução da amplitude máxima de aproximadamente 1200

Tabela 7.1: Comparação de desempenho ( $PRD \times CR$ ) entre os algoritmos CDA, CDT e CDTR e algoritmos da literatura [17, 38, 69–71, 71]

Algoritmo	100		102		107		115		117		119		Média CR	Média PRD
	CR	PRD	CR	PRD	CR	PRD	CR	PRD	CR	PRD	CR	PRD		
Jha [38]	39,90	0,23	27,90	0,50	39,60	0,66	46,80	0,28	47,00	0,31	38,50	0,47	39,95	0,41
Tun [69]	10,79	0,42	10,04	0,35	9,82	0,19	-	-	-	-	-	-	-	-
Lee [70]	22,94	1,95	25,91	1,39	18,55	3,93	19,88	2,98	24,43	1,17	19,31	2,05	21,84	2,24
Fathi [71]	31,40	3,20	27,50	2,20	-	-	-	-	29,30	3,00	-	-	-	-
MMP EqDC [17]	24,00	3,30	-	-	10,70	1,70	24,00	1,26	13,00	0,91	-	-	-	-
MMP EqDC/SS [17]	-	-	29,49	2,49	-	-	30,60	2,92	-	-	20,90	1,93	27,00	2,45
SDE GoS-3 [17]	26,33	11,27	26,76	10,18	13,64	6,66	26,87	14,48	21,57	2,17	21,23	3,66	22,73	8,07
SDE GoS-5 [17]	25,01	14,21	24,82	13,16	13,53	12,68	27,98	18,14	16,82	2,65	20,32	4,55	21,41	10,90
TDE GoS-3 [17]	29,26	3,81	28,45	5,28	9,57	2,77	36,41	4,19	29,58	1,62	20,35	1,81	25,60	3,25
TDE GoS-5 [17]	23,23	4,06	27,35	6,47	9,66	4,44	39,27	4,64	27,22	2,06	29,84	2,66	26,10	4,06
CDA GoS-3	24,00	6,95	20,76	6,50	15,42	7,14	24,02	6,00	21,50	3,20	22,43	4,00	21,36	5,63
CDA GoS-5	24,00	9,01	23,57	14,99	13,72	13,22	22,00	8,17	19,00	2,20	21,56	5,01	20,64	8,77
CDT GoS-3	22,00	3,35	28,93	5,52	7,83	3,95	39,60	3,50	30,85	1,49	19,00	2,84	24,70	3,44
CDT GoS-5	26,52	4,50	27,35	6,00	11,18	5,20	34,72	4,21	25,05	1,75	21,00	2,94	24,30	4,10
CDTR GoS-3	25,00	2,50	25,00	3,62	15,00	3,60	30,00	2,80	30,00	1,42	20,00	1,61	24,17	2,58
CDTR GoS-5	20,00	3,40	25,00	4,60	15,00	6,05	30,00	3,41	30,00	1,61	20,00	2,11	23,33	3,52

para 1000, sendo isso em virtude da regularidade baixa deste registro (ver Figura 7.7), o que compromete tanto o sinal completo quanto o resíduo. Por sua vez, o CDTR GoS-5 apresenta resultados inferiores ao CDTR GoS-3, o que era esperado, pois conforme comentado na Seção 6.3, GoS menores possuem capacidade de predição superior a grupos maiores, sendo os registros que apresentaram melhor e pior resultados, respectivamente, 117 e 107, pelos mesmos motivos de regularidade destes sinais comentado anteriormente.

Comparando os resultados do CDTR na Tabela 7.1 com o CDT e CDA, observa-se que CDTR apresenta desempenho melhor que o CDT e por sua vez superior ao CDA, comparando o mesmo tamanho de GoS, o que já era esperado, pois, resíduo de ECG contém amplitude inferior comparado ao sinal de ECG completo, além disso, as redundâncias são mantidas, o que garante bom desempenho. Comparando o CDTR com algoritmos da literatura que fazem uso de DSC [17], verifica-se também resultados superiores para todos registros, neste caso considerando CODECs com mesmo tamanho de GoS. Uma terceira comparação, desta vez com algoritmos tradicionais [17, 38, 69–71], verifica-se que CDTR aproxima-se de tais algoritmos, contudo, não os supera, o que é normal em DSC, pois conforme comentado na Seção 6.6.3, uma perda de taxa acontece no DSC. Resumidamente, o CDTR apresenta resultados superiores ao CDT.

A Figura 7.9 apresenta as curvas desempenho  $PRD \times CR$  dos registros 100, 102, 107, 115, 117 e 119 do CDTR. Onde resultados de diferentes taxas de compressão são mostrados. No caso do CDTR GoS-3 mostrado na Figura 7.9 (a), os registros 117 e 119 aparecem com  $PRD$  mais próximos de zero, em razão de regularidade alta entre segmentos adjacentes, além de apresentarem resíduo com menor amplitude, o que contribui grandemente para o bom desempenho

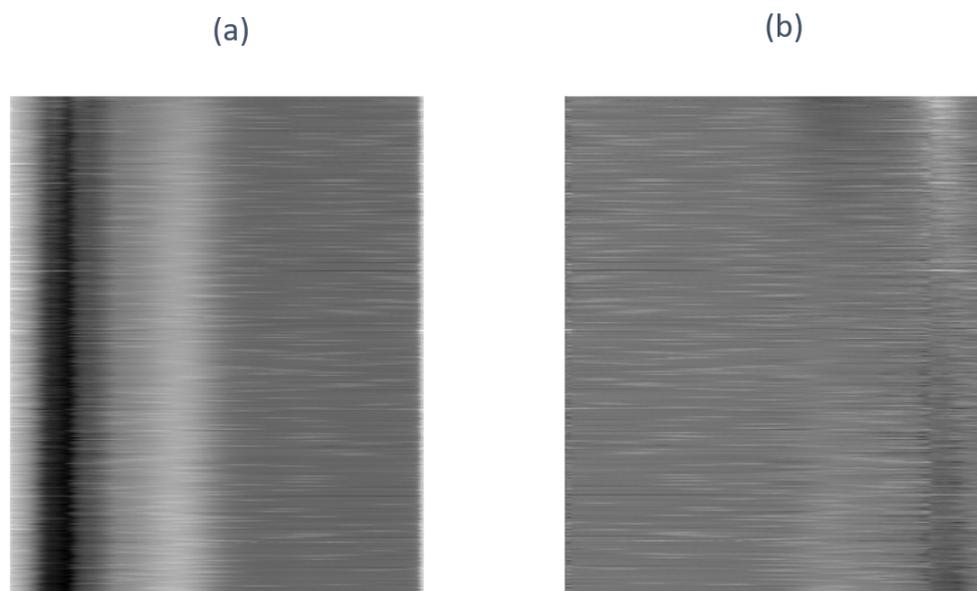


Figura 7.7: WZSs do registro 107. (a) Sinal completo; (b) Resíduo

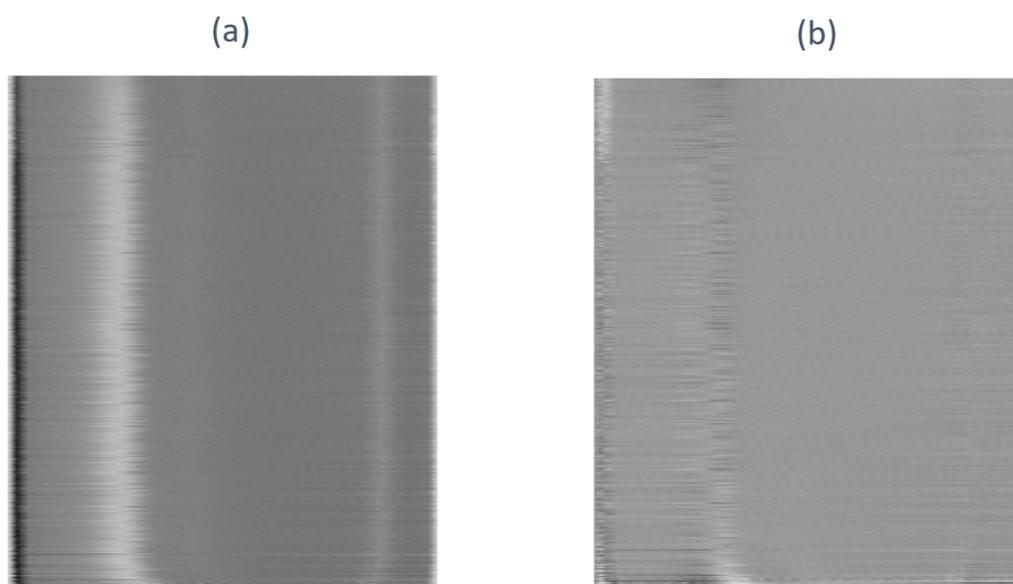


Figura 7.8: WZSs do registro 117. (a) Sinal completo; (b) Resíduo

destes registros, diferente do registro 107 que possui amplitude residual baixa e não possui boa regularidade entre segmentos adjacentes, conforme comentado anteriormente. Esse comportamento repete-se quando utilizado GoS-5 mostrado na Figura 7.9 (b). Contudo, o CDTR GoS-3 supera os resultados do CDTR GoS-5, em razão da predição de WZSs ser superior quando o GoS de tamanho 3 é utilizado.

Como visto na Seção 6.1, o CDT GoS-3 apresenta resultados melhores em comparação ao CDA. Entretanto, quando o resíduo de ECG é utilizado, alcança-se resultados superiores ao CDT, considerando o mesmo tamanho de GoS, conforme mostrado na Figura 7.10, a qual mostra a média de desempenho das cruvas dos registros codificados. A razão destes resultados está na codificação residual de ECG, que possui menor quantidade de informação aliada ao CDT que possui etapa de transformação DCT que permite exploração de redundâncias intrabatimentos. Logo, a utilização de resíduo de sinais de ECG mostra um ganho significativo no desempenho de compressão, pois existe maior descarte de informações desnecessárias à codificação, permitindo assim o envio de menos *bits* de paridade ao decodificador.

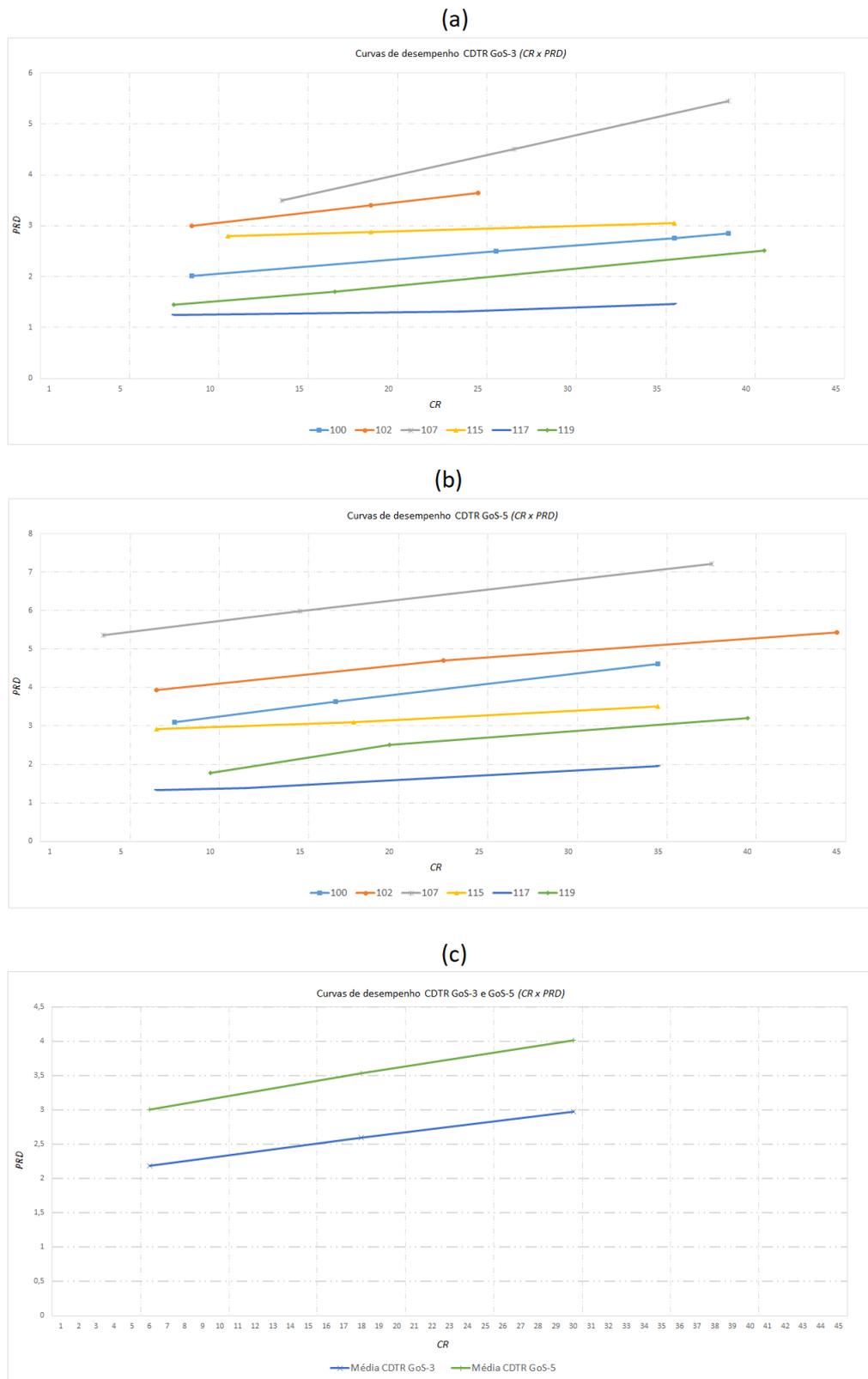


Figura 7.9: Desempenho de compressão do CDTR GoS-3 e GoS-5 dos registros 100, 102, 107, 115, 117 e 119. (a) CDTR GoS-3; (b) CDTR GoS-5; (c) Média de CDTR GoS-3 e GoS-5

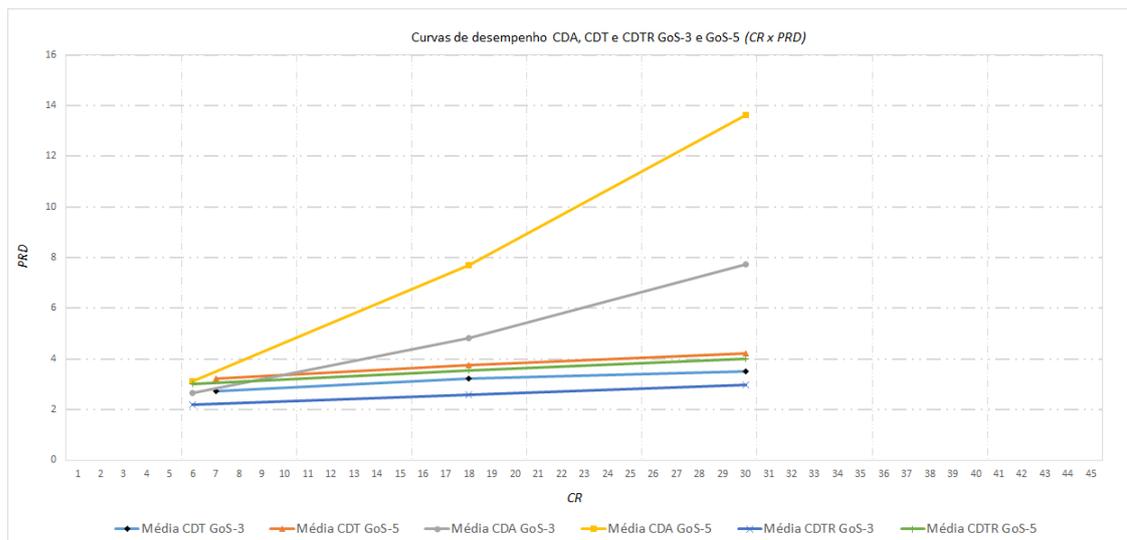


Figura 7.10: Média do desempenho de compressão do CDA, CDT e CDTR GoS-3 e GoS-5

As Figuras 7.11 a 7.14 mostram sinais de ECG originais, reconstruídos e o erro de reconstrução. Onde são mostrados sinais dos registros 100, 107 e 109. Os 10 primeiros segundos de cada registro são mostrados. Os registros 100 e 107 apresentaram distorção leve na onda R. O registro 107 (Figura 7.13 e 7.14) refere-se a um paciente que apresenta bloqueio de ramo, o que é evidenciado por longos complexos QRSs, sendo essa característica mantida no sinal reconstruído. Outro detalhe da Figura 7.13 que mostra resultado de codificação do registro 107, o qual não possui regularidade em sua estrutura original, logo apresenta erro de reconstrução moderado.

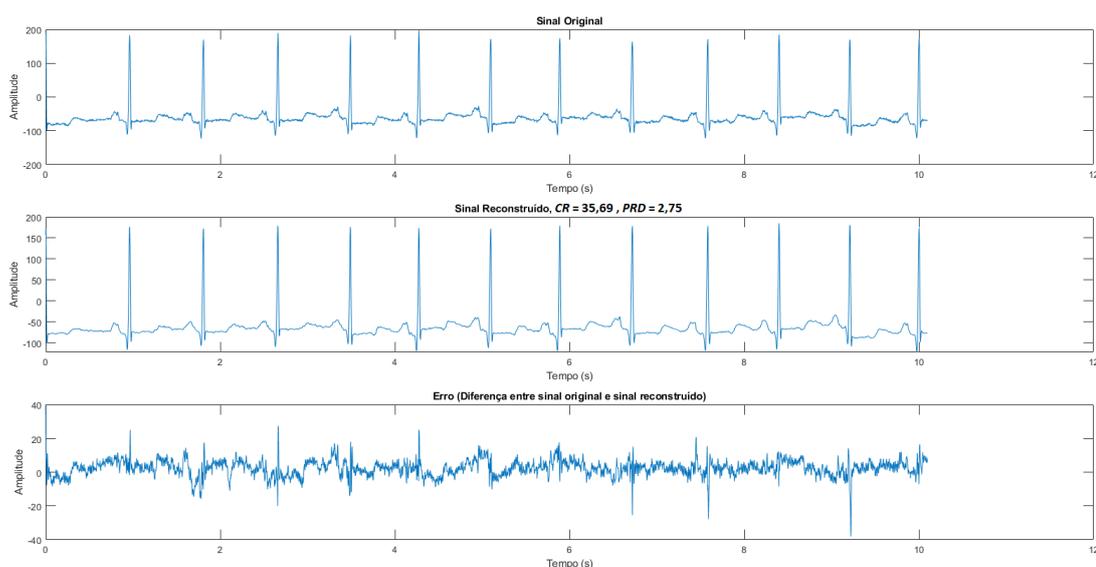


Figura 7.11: Registro 100 reconstruído com CDTR GoS-3 com  $CR = 35,69$  e  $PRD = 2,75$ .

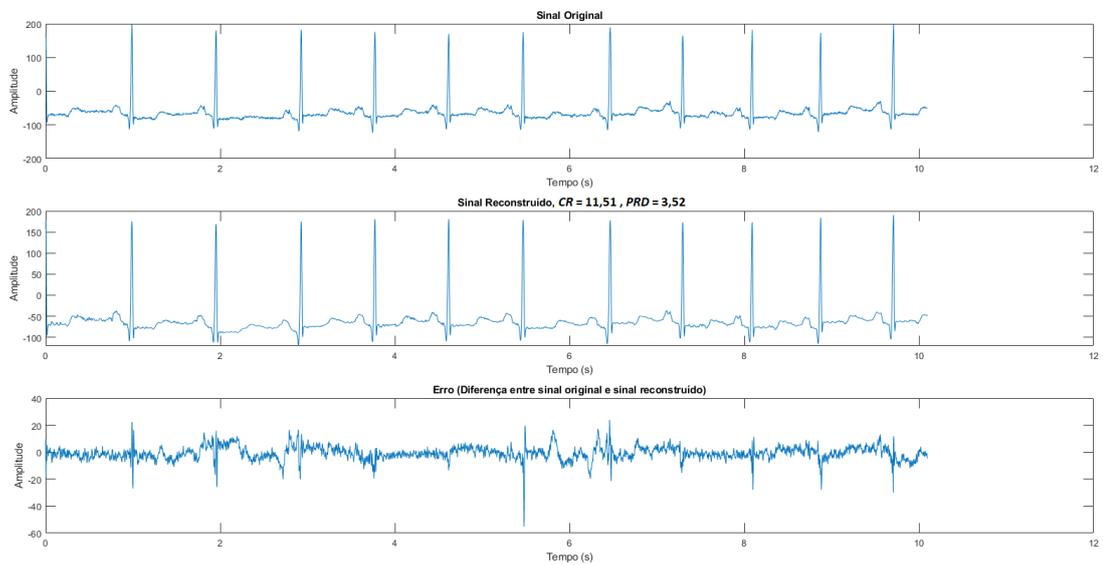


Figura 7.12: Registro 100 reconstruído com CDTR GoS-5 com  $CR = 11,51$  e  $PRD = 3,52$ .

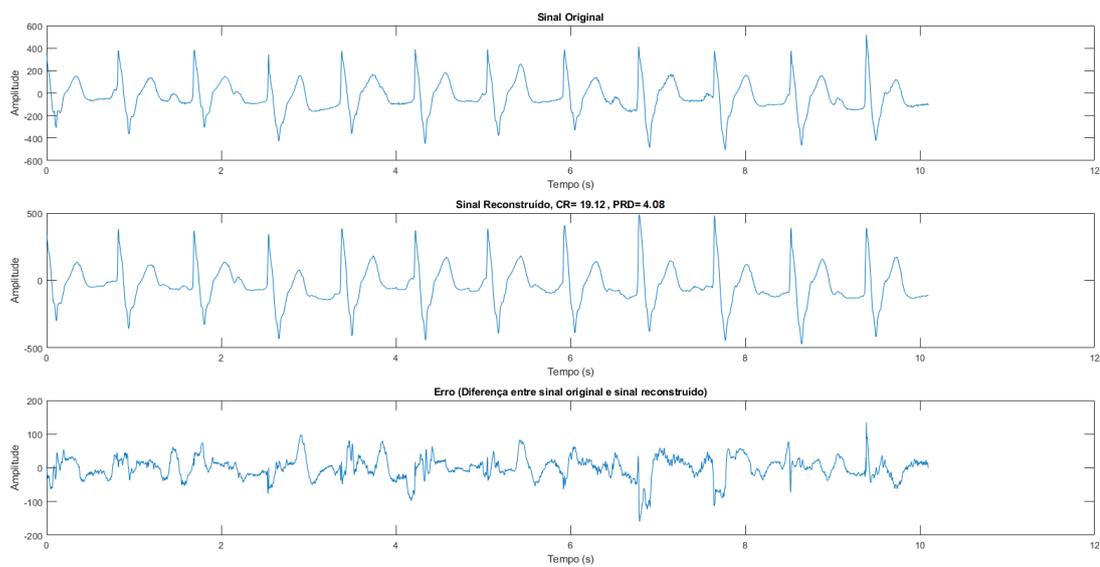


Figura 7.13: Registro 107 reconstruído com CDTR GoS-3 com  $CR = 19,12$  e  $PRD = 4,08$ .

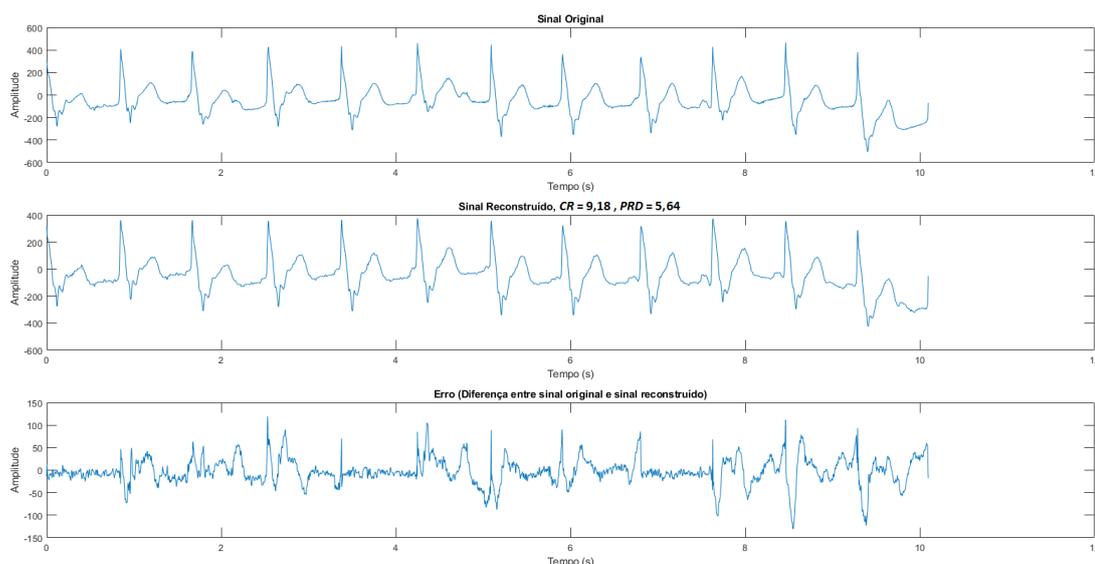


Figura 7.14: Registro 107 reconstruído com CDTR GoS-5 com  $CR = 9,18$  e  $PRD = 5,64$ .

### 7.2.1 Discussão e análise de resultados

A codificação residual mostrou-se superior a codificação de sinais de ECG completo, o que era esperado, entretanto, o mesmo possui acréscimo de complexidade, que é o cálculo do resíduo no codificador, contudo, são realizadas operações aritméticas básicas no cálculo do resíduo como: soma, subtração, multiplicação e divisão, não havendo, portanto, complexas explorações estatísticas no lado do codificador e nem transformações que trabalham com pontos flutuantes como nos trabalhos de Jha e Kolekar [38] e Tun, Moe e Naing [69]. Sendo assim, o CDTR torna-se uma solução viável quando busca-se relações taxas-distorção superiores ao CDT base desde que o dispositivo utilizado suporte um pequeno acréscimo computacional.

Uma característica comum nos CODECs implementados e apresentados até agora é que todos utilizam o canal de retorno, o qual pode não ser fisicamente implementável, pois a teoria DSC busca a realização de codificadores simplificados. Sendo assim, o canal de retorno, o qual tem a função de solicitar *bits* de paridade ao perfurador, garantindo assim uma quantidade mínima de *bits* enviados, pode ser eliminado, permitindo o envio único de *bits* de paridade sem a necessidade de solicitações de *bits*. Essa proposta é detalhada no capítulo seguinte.

# Capítulo 8

## Retirando o canal de retorno

Neste capítulo, a metodologia do CDT sem utilização do canal de retorno será apresentada, este que é um fator limitante para diversas aplicações práticas, pois, sua implementação pode ser inviável. Pensando nisso, uma metodologia de eliminação desse canal é proposta, a qual é implementada sem aumento substancial de complexidade, através da realização de estimativas da taxa de perfuração de *bits* de paridade, o que permite que estes sejam enviados uma única vez. A Figura 8.1 representa a estrutura do CDT sem canal de retorno, onde mostra que o codificador CDT tem conhecimento de SIs no lado do codificador, dessa forma, uma função foi elaborada com esse conhecimento, na qual são utilizadas variáveis que possuem relação com a taxa de *bits* na etapa de perfuração, sendo estas variáveis escolhidas após observação de resultados do CDT com canal de retorno.

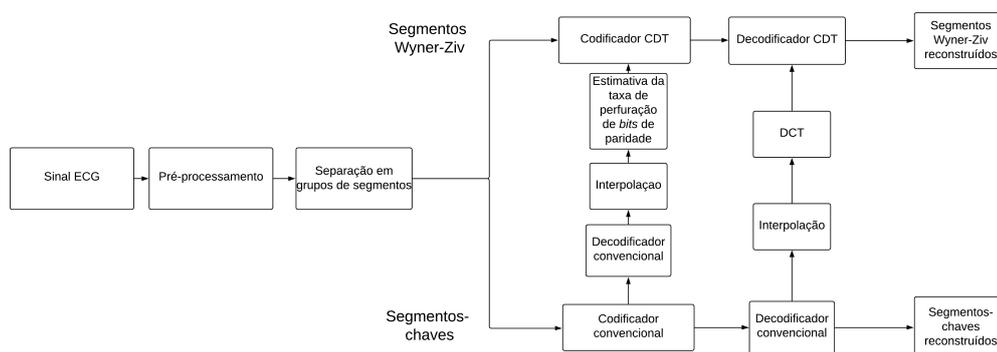


Figura 8.1: Estrutura do CDT com estimador da taxa de perfuração.

## 8.1 Metodologia

Como visto na Seção 6.4.4, o perfurador envia *bits* de paridade ao decodificador alterando padrões de perfuração, onde inicialmente nenhum *bit* é enviado, e conforme decodificador solicita mais *bits*, o padrão é alterado, provocando o envio de mais *bits*. A largura de perfuração utilizada no CDT é de 32 *bits*, ou seja, os *bits* de paridade que saem do codificador RSC são divididos em grupos de 32 *bits*. Essa mesma largura é mantida nessa metodologia e o estimador de taxas atua diretamente estabelecendo o padrão de perfuração que garante uma certa quantidade de *bits* enviados ao decodificador, de forma que ao final da decodificação a taxa de erro, conseqüentemente, seja menor que  $10^{-3}$ , e dessa forma permite a decodificação sem necessidade de solicitar mais *bits*. Sendo assim, na idealização da função que fornece o valor de perfuração foi necessário utilizar fundamentos de entropia de Shannon, delineada na equação (8.1), a qual calcula a máxima taxa de *bits* na ocorrência de uma certa probabilidade de erro  $p$ , conforme trabalho de Lee, Tillo e Jeon [24]. Sendo assim, a entropia de  $p$  é dada por

$$H(p) = -p \log_2(p) - (1 - p) \log_2(1 - p), \quad (8.1)$$

onde  $H(p)$  é a entropia da mensagem com erro e  $p$  refere-se à probabilidade de *bits* errados. O cálculo de  $p$  é realizado através da soma de todas as ocorrências de bits diferentes entre um plano de bits de WZSs originais e suas predições SIs, que então é dividida pela largura do plano. Sendo assim,  $p$  representa, no cálculo da entropia, o ruído adicionado.

De fato, experimentalmente observa-se que quanto mais semelhança existir entre WZSs e SIs, menor é o valor de  $p$  e menos *bits* de paridade são enviados, garantindo assim, melhor desempenho de compressão. Logo, essa interessante relação entre taxa de perfuração e taxa de erro  $p$  foi observada, o que foi utilizado no cálculo de estimativas juntamente a entropia, conforme é delineado a seguir.

Após cálculo de entropia  $H(p)$ , a capacidade do canal é calculada através da fórmula

$$C = 1 - H(p), \quad (8.2)$$

onde  $C$  denota a capacidade do canal binário e  $H(p)$  refere-se à entropia da taxa de erro para determinado plano de *bits*. Entretanto,  $C$  representa a capacidade máxima de um canal binário, o que é possível apenas teoricamente. Portanto, aplicações práticas sempre terão valores menores

ou próximos a  $C$ . Sendo assim, conforme Lee, Tillo e Jeon [24], a taxa pode ser obtida através da inequação

$$R_c = \frac{k}{m+k} \leq C, \quad (8.3)$$

sendo  $m$  a quantidade de *bits* de paridade mínima que precisam ser enviados ao decodificador, e a parcela  $k$  representa a largura do plano de *bits*. Logo, para um dado plano de *bits* de largura  $k$ , uma quantidade  $m$  de *bits* de paridade é necessário. Dessa forma, quanto menos *bits* de paridade, menor é a taxa. O termo  $m$  pode ser isolado e definido assim

$$m = \frac{1-C}{C} k. \quad (8.4)$$

Apesar de saber a quantidade mínima de *bits* de paridade necessários, isso ainda não é suficiente, pois a taxa precisa ser dividida entre blocos de perfuração. Sendo assim, uma solução aproximada foi encontrada empiricamente, depois de inúmeros experimentos, a qual é expressa a seguir

$$T_b = \frac{k}{b} \log\left(1 + \frac{m}{k}\right) - \log(p), \quad (8.5)$$

onde  $k$  representa a largura do plano de *bits*,  $b$  foi definido com valor igual a 64,  $m$  corresponde ao amontoado de *bits* de paridade e  $p$  representa a probabilidade de *bits* errados entre WZSs (segmentos Wyner-Ziv) e SIs (informação lateral) do plano de *bits* a ser codificado. O segundo logaritmo é um fator de ajuste que permite estimativas sempre acima do valor esperado, evitando que haja estimativas abaixo, o que pode gerar erros catastróficos na reconstrução. Sendo assim  $T_b$  fornece valores entre 0 e 32, que correspondem a quantidade de *bits* de paridade enviados por um dos dois codificadores RSCs em cada bloco, onde o mesmo valor de  $T_b$  é considerado para ambos codificadores, por exemplo, determinado plano de *bits* de WZSs e SIs resultou em um  $T_b = 5$ , isso significa que 5 de 32 *bits* são enviados em cada bloco de tamanho 32 na saída do primeiro codificador, assim como no segundo codificador.

A estrutura desta metodologia de estimação da taxa de perfuração é exemplificada na Figura 8.2, onde mostra a função  $f(H(p), p)$ , a qual utiliza as variáveis  $p$  e  $H(p)$  para fornecer a taxa de perfuração adequada.

Importante ressaltar que a equação (8.5) foi definida através de inúmeros experimentos, utilizando para isso dados previamente codificados com o CDT, o que permitiu justamente a

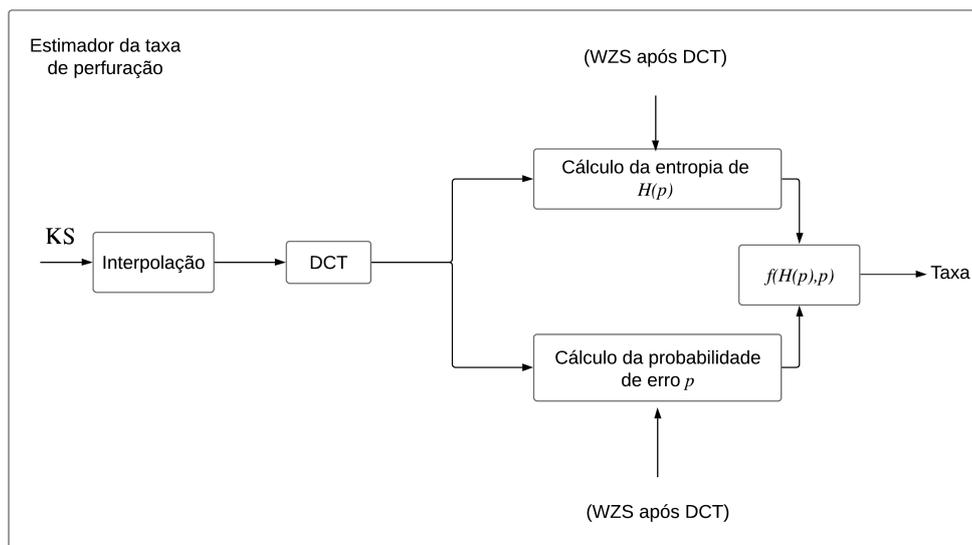


Figura 8.2: Estrutura de controle da taxa de perfuração.

análise de variáveis fornecidas por tal CODEC, sendo a entropia e probabilidade de *bits* errados as variáveis que mostraram relação com a taxa de perfuração, por essa razão, são justamente as variáveis utilizadas na equação (8.5). Como resultado, o conteúdo apresentado fornece toda fundamentação necessária para compreensão dos resultados obtidos experimentalmente, os quais são descritos na próxima seção.

## 8.2 Resultados

Nessa seção, os resultados do CDT sem canal de retorno são apresentados, onde, em primeiro lugar a comparação de resultados do CDT com e sem canal de retorno é mostrada. A seguir, resultados de codificação de registros de ECG no CDT sem retorno são comparados a algoritmos da literatura que possuem estrutura DSC, e também com arquiteturas que utilizam técnicas tradicionais de codificação. Depois, as curvas que representam  $PRD \times CR$  de CDT com e sem retorno são mostradas. Por último, o desempenho de compressão de WZSs utilizando diferentes taxas de compressão de KSs (segmentos-chaves) são mostrados em uma tabela .

Inicialmente, o desempenho do CDT com e sem canal de retorno são comparados, onde os 10 primeiros minutos dos registros 100, 102, 107, 115, 117 e 119 da base de dados do MIT-BIH [82] foram codificados utilizando GoS-3 e GoS-5. Sendo assim, a Figura 8.3 mostra a porcentagem de *bits* excedidos do valor apresentado com canal de retorno. Por exemplo, o

registro 117 quando utilizado GoS-3 sem retorno envia em torno 30% de *bits* a mais que o valor de referência, a qual é a quantidade de *bits* totais enviados pelo CDT com retorno, por outro lado, o registro 107 quando codificado com GoS-5 sem canal de retorno necessita enviar quase 50% a mais, quando utilizado GoS-5. Os resultados do registro 107 são reflexos de uma característica importante de sinais de ECG, que é a regularidade do sinal, o que, por sua vez, afeta as previsões de WZSs, o que influencia na estimativa da taxa de *bits*. Outra importante análise de desempenho do estimador é mostrada nas Figuras de comparação da codificação com e sem canal retorno dos registros 107 e 117, as quais são ilustradas nas Figuras 8.4 a 8.7, as quais mostram o valor de perfuração de cada bloco de tamanho 32 para as 16 bandas de coeficientes de 5 planos de *bits*, de forma que nessas figuras observa-se a proximidade do valor estimado com o valor esperado no caso do registro 117 (Figuras 8.5 e 8.7), diferente do registro 107 que apresenta uma distância maior entre valor esperado e estimado (Figuras 8.4 e 8.6).

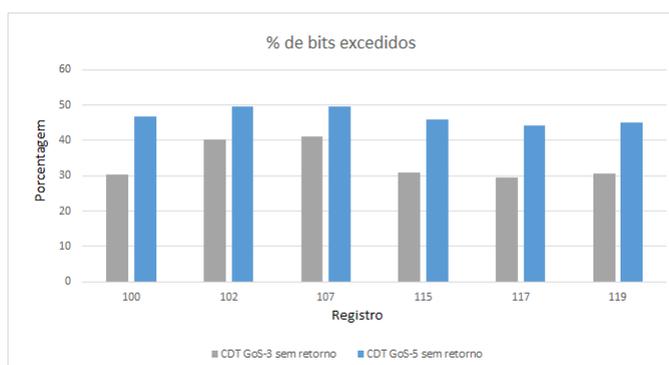


Figura 8.3: Porcentagem de *bits* excedidos dos registros 100, 102, 107, 115, 117 e 119 do CDT sem canal de retorno, utilizando GoS-3 e GoS-5

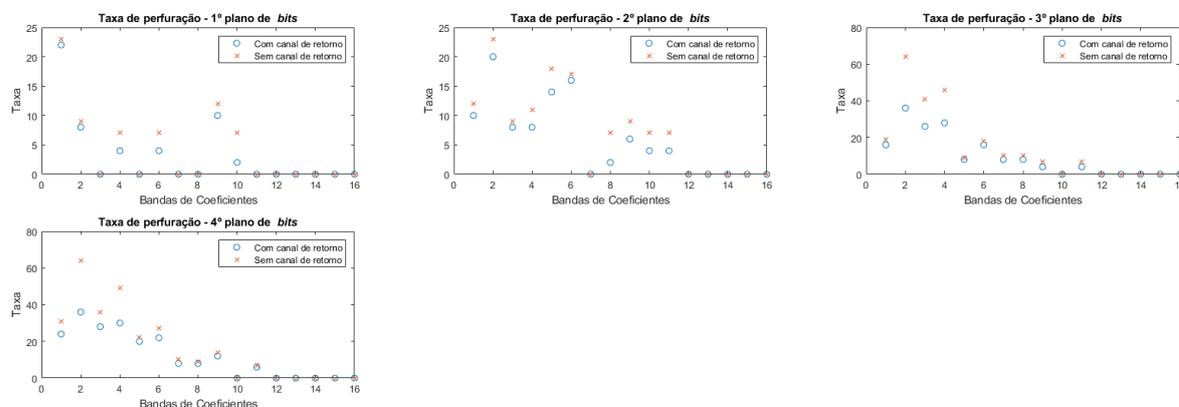


Figura 8.4: Desempenho de taxa de perfuração do registro 107 do CDT GoS-3 com e sem canal de retorno.

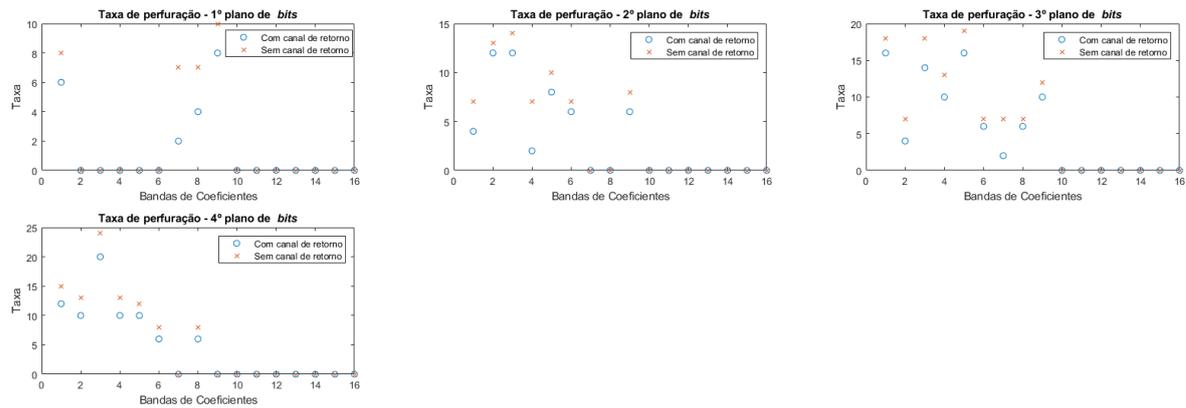


Figura 8.5: Desempenho de taxa de perfuração do registro 117 do CDT GoS-3 com e sem canal de retorno.

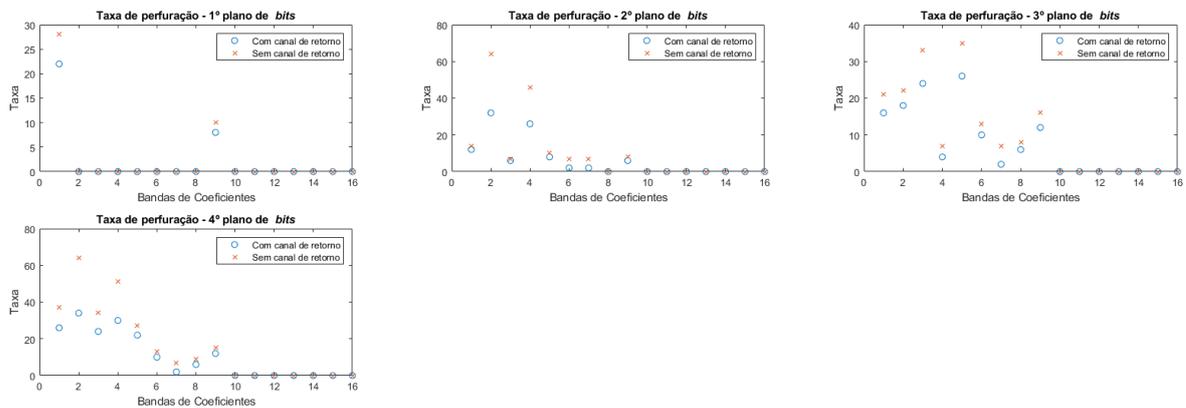


Figura 8.6: Desempenho de taxa de perfuração do registro 107 do CDT GoS-5 com e sem canal de retorno.

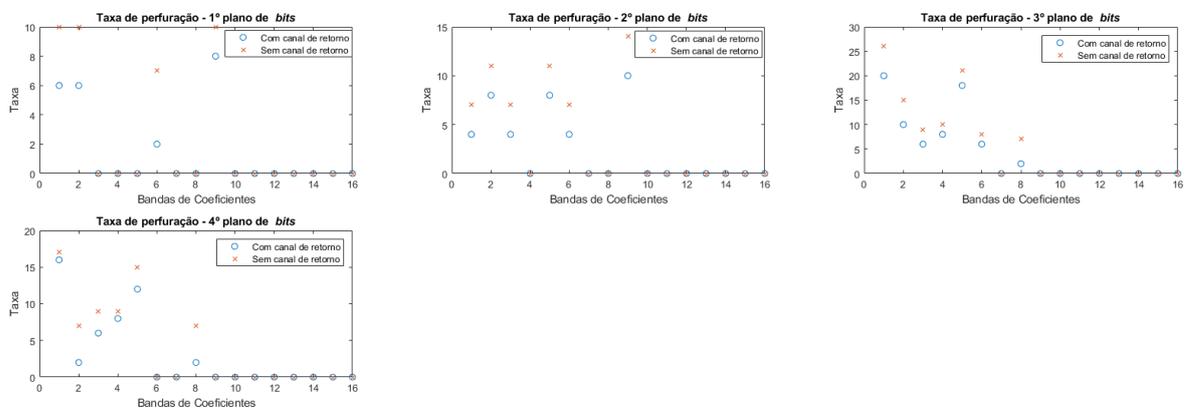


Figura 8.7: Desempenho de taxa de perfuração do registro 117 do CDT GoS-5 sem canal de retorno.

A Tabela 8.1 mostra os resultados de todas metodologias implementadas nessa dissertação, as quais são: CDA, CDT, CDTR e CDT sem retorno, utilizando GoS-3 e GoS-5. Um total de 100 WZSs dos registros 101, 102, 109, 112, 118, 122, 214 e 220 foram codificados, seguindo o que foi realizado no trabalho de Wu, Tsai e Chang [14], o qual apresenta dois algoritmos que possuem arquitetura DSC, denominados DSC1 e DSC2, onde ambos utilizam quantização vetorial, diferente de todos CODECs implementados nesse trabalho que utilizam quantização uniforme. Outras metodologias são comparadas, por exemplo, os algoritmos que utilizam a transformada wavellet [38, 69, 71, 84] e algoritmos que utilizam transformada DCT [70].

Analisando os dados da Tabela 8.1, começando pelos resultados do CDT sem retorno, a diferença para o CDT com retorno pode ser verificada pela média de  $CR$  e  $PRD$  apresentada nas duas últimas colunas, onde verifica-se que o CDT sem retorno apresenta um resultado inferior ao CDT base, o que era esperado, pois, conforme mostrado anteriormente no início dessa seção o CDT sem retorno realiza estimativas sempre acima do que é esperado, o que é evidenciado nessa tabela. Por outro lado, o CDTR GoS-3 apresenta resultados superiores quando comparado a todos algoritmos apresentados nessa dissertação, assim como, quando comparado ao DSC1 e DSC2, sendo inclusive superior em boa parte dos registros. Além disso, mais uma vez a diferença de resultados entre GoS-3 e GoS-5 é evidenciada nesses resultados. Uma última comparação, desta vez com algoritmos que utilizam codificadores tradicionais [25, 38, 69, 71], percebe-se que a codificação DSC não consegue superioridade a tais codificadores tradicionais, o que é natural, pois existe uma perda de taxa intrínseca à codificação Wyner-Ziv, conforme mostrado na Seção 5.3.

Dentre os registros apresentados na Tabela 8.1, os que apresentaram melhor e pior desempenho, para todos CODECs implementados, foram respectivamente, 122 e 214, o motivo está na regularidade desses registros, o que afeta o desempenho da predição de WZSs, dessa forma, o desempenho de compressão de WZSs é influenciado.

A segunda parte desta seção de resultados são as curvas de desempenho de compressão ( $CR \times PRD$ ), as quais são mostradas na Figura 8.8, onde a Figura 8.8 (a) mostra o desempenho do CDT sem retorno GoS-3 para os registros 100, 102, 17, 115 e 117. No caso do GoS-3, o registro 117 apresenta desempenho superior aos demais, ao contrário do registro 107, contudo, em geral os resultados do GoS-3 são satisfatórios, principalmente para taxas de  $CR$  menores que 1 : 10, pois acima disso observa-se uma elevação da distorção. Os resultados do CDT

Tabela 8.1: Comparação de desempenho (*PRD* x *CR*) entre os algoritmos CDA, CDT, CDTR, CDT sem retorno e algoritmos da literatura [14, 38, 69–71, 84, 90],

Algoritmo	101		102		109		112		118		122		214		220		Média <i>CR</i>	Média <i>PRD</i>
	<i>CR</i>	<i>PRD</i>																
Jha [38]	48,70	0,23	27,90	0,50	38,60	0,38	53,90	0,22	42,60	0,67	20,20	0,29	34,70	0,43	46,20	0,32	39,10	0,38
Tun [69]	10,06	0,30	10,04	0,35	10,72	0,32	-	-	-	-	-	-	-	-	-	-	-	-
Lee [70]	23,86	2,09	25,91	1,39	19,89	0,76	23,82	1,00	19,83	2,04	21,30	1,05	19,97	1,15	20,06	2,81	21,83	1,54
Fathi [71]	28,40	3,40	27,50	2,20	-	-	30,00	2,90	-	-	-	-	-	-	-	-	-	-
JPEG2000 [84]	25,86	4,95	21,45	5,65	18,46	3,21	18,64	1,39	21,62	3,27	19,52	1,96	20,82	5,96	22,72	3,00	21,14	3,67
DSC1 [14]	25,86	5,79	21,45	9,40	18,46	4,08	18,64	4,29	21,62	3,28	19,52	5,90	20,82	7,04	22,72	3,84	21,14	5,45
DSC2 [14]	25,86	3,53	21,45	3,87	18,46	2,46	18,64	1,39	21,62	2,07	19,52	1,31	20,82	4,22	22,72	2,32	21,14	2,65
CDA GoS-3	20,55	4,56	28,50	8,99	13,61	4,10	19,69	1,86	20,05	3,24	19,99	2,46	19,75	5,88	20,60	3,45	20,34	4,32
CDA GoS-5	19,40	4,62	20,95	6,10	12,50	4,55	17,01	4,50	19,88	3,50	19,05	3,01	18,01	6,20	19,50	3,62	18,29	4,51
CDT GoS-3	28,36	4,34	23,00	3,85	21,91	4,95	20,00	1,53	23,83	2,85	19,61	1,77	20,77	5,63	25,03	2,43	22,81	3,42
CDT GoS-5	26,35	4,55	23,02	3,97	14,95	4,13	18,00	3,58	17,19	2,50	22,64	2,72	20,53	5,86	23,85	3,23	20,82	3,82
CDTR GoS-3	33,21	2,00	22,34	3,70	20,52	4,43	29,95	1,37	17,62	1,86	19,15	1,68	17,35	3,95	22,10	2,00	22,78	2,62
CDTR GoS-5	13,77	2,14	20,93	4,71	18,35	4,70	18,70	1,40	16,50	1,95	20,92	1,92	17,78	5,19	24,49	2,35	19,23	3,05
CDT GoS-3 sem retorno	14,34	2,09	22,66	4,19	21,07	5,87	21,97	3,54	19,72	2,09	18,66	2,10	15,56	4,30	20,78	2,55	19,35	3,34
CDT GoS-5 sem retorno	11,87	2,53	14,16	9,37	15,18	8,56	20,04	3,82	11,61	1,87	16,67	2,36	15,44	5,44	17,00	2,99	15,25	4,62

sem retorno GoS-5 (Figura 8.8) (b), da mesma maneira apresenta o mesmo comportamento do GoS-3, contudo, GoS-3 tem resultados superiores conforme a Figura 8.8 (c).

Um comportamento observado no CDT, sem canal de retorno, quando comparado ao CDT com retorno, é que conforme a taxa de *CR* aumenta, o desempenho do estimador diminui, ou seja, a curva do CDT sem retorno se distancia da curva do CDT com retorno, o que é mostrado na Figura 8.9 que mostra a média dos registros 100,102,107,115,117 e 119. Isso acontece em razão do estimador ser otimizado para taxas de *CR* abaixo de 1 : 10, a qual apresenta os melhores resultados de *PRD*. O que não inviabiliza a utilização de taxa maiores.

Conforme a metodologia de DSC aplicada a compressão de sinais de ECG, os WZSs são decodificados com auxílio de SIs, sendo WZSs codificados no codificador Wyner-Ziv e SIs gerados a partir de KSs codificados e decodificados por um codificador convencional. Entretanto, as análises de resultados das implementações dessa dissertação, visam somente resultados de WZSs, pois busca-se a avaliação de desempenho nos algoritmos que agem apenas nos WZSs. Portanto, os KSs são considerados sem perdas, ou seja, uma codificação sem perdas é assumida, o que permite a utilização de KSs perfeitamente reconstruídos no decodificador. Contudo, a qualidade da reconstrução de WZSs está relacionada com o nível de compressão de segmentos KSs. O que foi experimentalmente comprovado e exibido na Tabela 8.2, que mostra a taxa de compressão de WZSs dos CODECs implementados nesta dissertação, utilizando três pontos de compressão de KSs, para isso, o JPEG2000 foi utilizado como codificador convencional. Portanto, primeiramente, KSs são considerados sem distorção, a seguir uma taxa de 1 : 2 no JPEG2000 foi considerada, e em terceiro o mesmo valor de *CR* é considerado entre WZSs e KSs. Analisando dados dessa Tabela 8.2 constata-se que, a medida que a taxa de compressão de KSs aumenta, o *CR* de WZSs diminui e o *PRD* de WZSs aumenta, o que era esperado, pois

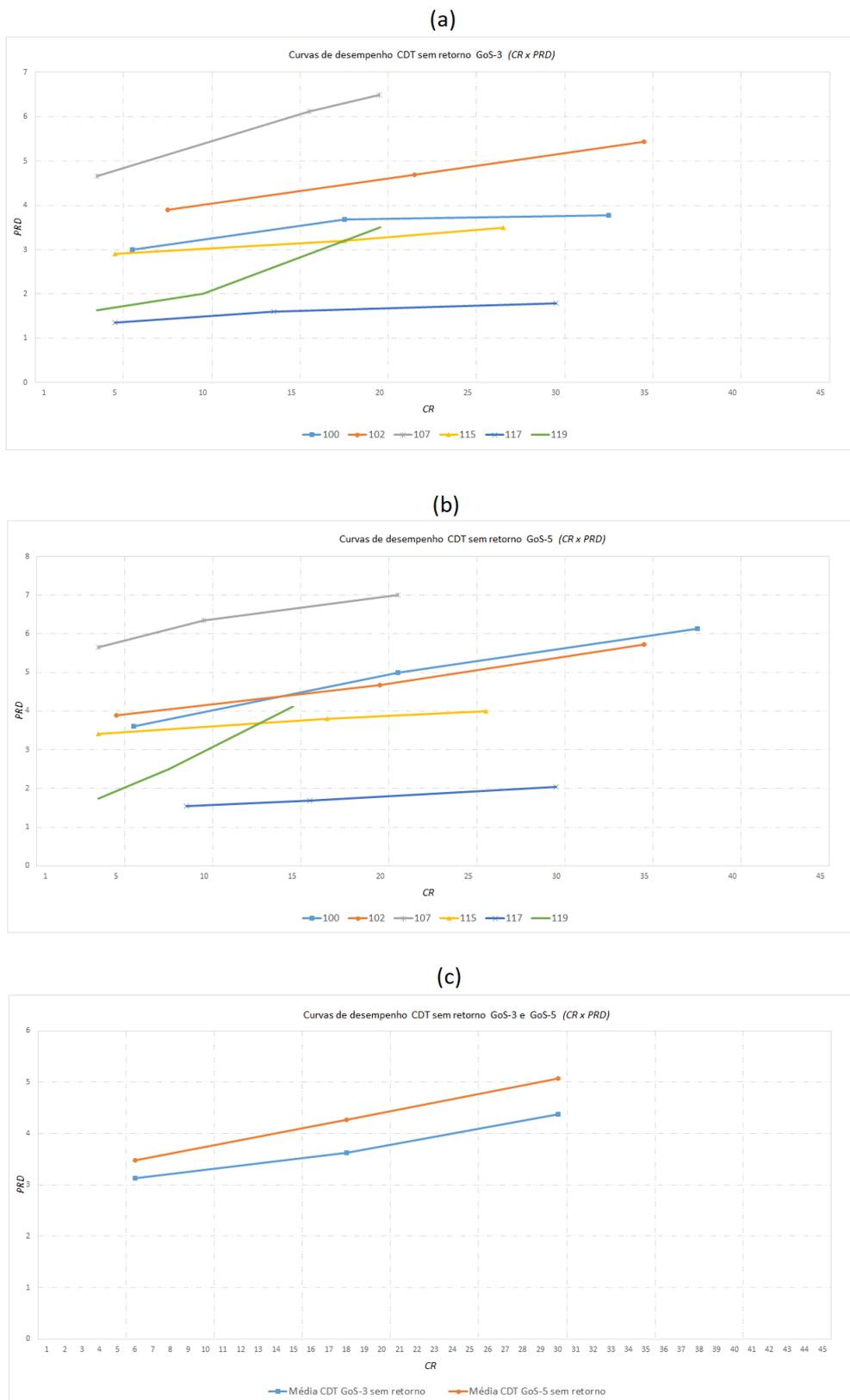


Figura 8.8: Desempenho de compressão do CDT sem canal de retorno, (a) CDT sem retorno GoS-3, (b) CDT sem retorno GoS-5 e (c) a média de resultados de CDT sem retorno, utilizando GoS-3 e GoS-5.

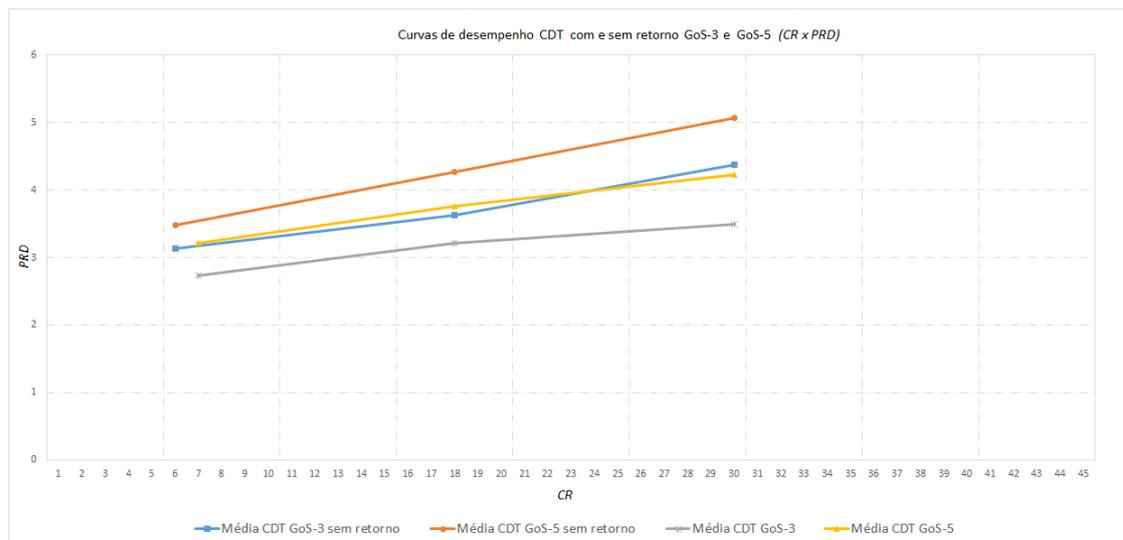


Figura 8.9: Média do desempenho de compressão dos CODEC's CDT com e sem canal de retorno, utilizando GoS-3 e GoS-5.

aumentando o  $CR$  de KSs, conseqüentemente diminui a qualidade da predição realizada através da interpolação de KSs, logo isso implica em diminuição da qualidade de SIs, o que afeta o desempenho de compressão de WZSs. Outra importante observação é o valor de  $CR$  considerando o sinal de ECG totalmente reconstruído, ou seja, agrupando WZSs e KSs, sendo assim, o valor de  $CR$  nesta situação é menor que o valor de  $CR$  de WZSs.

### 8.2.1 Discussão e análise de resultados

Os resultados do CDT sem retorno são satisfatórios, embora seja o codificador que necessite da realização de uma quantidade de cálculos maior dentre as metodologias apresentadas nessa dissertação, pois, há acréscimo de exploração estatística por meio de cálculos de entropia e de probabilidades. Contudo, o ganho desta metodologia de eliminação do canal está na eliminação de *hardware* relacionado ao canal de retorno. De fato, o acréscimo computacional é balanceado pela diminuição de circuitaria relacionado ao canal de retorno, caracterizando assim o CDT sem canal de retorno como uma boa solução, mesmo que o desempenho de compressão seja inferior a solução com canal.

Tabela 8.2: Desempenho de compressão das metodologias apresentadas nesta dissertação utilizando diferentes valores de taxas de compressão de KSs

Algoritmo	Registro	KSs sem distorção			KSs com CR= 1:2			(CR KSs) = ( CR WZSs)		
		CR WZSs	PRD WZSs	PRD WZSs + KSs	CR WZSs	PRD WZSs	PRD WZSs + KSs	CR WZSs	PRD WZSs	PRD WZSs + KSs
CDA GoS-3	100	10,69	3,55	3,49	9,58	3,99	3,58	9,45	4,08	3,65
CDA GoS-5	100	11,75	4,20	3,53	10,05	4,61	3,62	9,50	4,70	3,71
CDT GoS-3	100	12,98	3,45	2,11	11,88	3,61	2,26	11,55	3,75	2,31
CDT GoS-3	100	12,46	3,65	2,63	12,25	3,69	2,64	10,48	3,82	2,80
CDTR GoS-3	100	18,21	2,34	1,71	15,32	2,51	1,85	15,01	2,82	1,99
CDTR GoS-5	100	10,91	3,45	3,06	10,89	3,46	3,08	10,85	3,49	3,15
CDT Gos 3 sem retorno	100	13,18	3,52	2,46	13,15	3,53	2,48	12,49	3,80	2,52
CDT Gos 5 sem retorno	100	13,03	4,10	2,70	13,00	4,11	2,71	12,89	4,22	2,82
CDA GoS-3	102	20,76	7,00	3,51	20,70	7,11	4,89	20,76	7,13	4,90
CDA GoS-5	102	16,79	6,24	5,96	16,79	6,24	5,96	16,75	6,25	5,98
CDT GoS-3	102	20,00	3,45	2,32	19,01	3,98	2,34	18,82	4,11	2,77
CDT GoS-3	102	27,35	4,59	4,13	25,00	4,79	4,65	23,24	4,81	4,69
CDTR GoS-3	102	40,54	3,93	2,90	40,30	3,93	2,90	40,30	3,95	3,38
CDTR GoS-5	102	30,00	4,71	3,52	29,80	4,73	3,54	29,01	4,89	3,92
CDT Gos 3 sem retorno	102	21,86	4,50	3,28	21,80	4,52	3,29	20,90	5,03	4,64
CDT Gos 5 sem retorno	102	19,51	4,64	3,92	19,60	4,65	3,93	16,03	4,71	4,14
CDA GoS-3	107	15,42	7,14	6,41	15,40	7,15	6,41	14,19	7,14	6,41
CDA GoS-5	107	10,00	7,00	6,50	9,91	7,10	6,55	9,85	7,13	6,58
CDT GoS-3	107	11,36	4,56	3,97	11,31	4,58	3,99	11,14	4,56	4,02
CDT GoS-5	107	11,18	5,20	4,75	10,70	5,36	4,95	8,37	5,72	5,43
CDTR GoS-3	107	13,52	3,74	2,57	13,53	3,75	2,58	13,44	3,78	2,67
CDTR GoS-5	107	10,00	5,67	4,91	9,98	5,68	4,92	9,91	5,75	5,06
CDT Gos 3 sem retorno	107	8,46	5,88	3,85	8,45	5,89	3,86	8,31	6,05	4,07
CDT Gos 5 sem retorno	107	9,05	5,91	4,52	9,01	5,92	4,53	8,91	6,02	5,55
CDA GoS-3	115	9,88	2,89	1,93	9,62	2,89	1,96	9,36	2,89	1,99
CDA GoS-5	115	10,00	4,00	3,82	9,88	4,11	3,89	9,76	3,95	3,93
CDT GoS-3	115	14,39	3,06	1,91	14,36	3,08	1,98	14,39	3,15	2,01
CDT GoS-5	115	17,46	3,48	2,41	17,42	3,49	2,48	17,31	3,81	2,86
CDTR GoS-3	115	31,00	3,00	1,72	30,00	3,15	1,75	29,05	3,21	2,00
CDTR GoS-5	115	30,36	3,00	2,62	31,30	3,01	2,63	24,81	3,55	3,44
CDT Gos 3 sem retorno	115	12,69	3,24	2,01	12,65	3,26	2,03	12,55	3,28	2,10
CDT Gos 5 sem retorno	115	9,26	3,65	2,69	9,25	3,66	2,71	9,01	3,79	2,84
CDA GoS-3	117	10,97	1,42	1,21	10,97	1,48	1,23	10,04	1,49	1,25
CDA GoS-5	117	11,02	1,89	1,28	10,95	1,92	1,31	10,91	1,99	1,36
CDT GoS-3	117	11,43	1,42	0,78	11,00	1,44	0,80	11,17	1,49	0,87
CDT GoS-5	117	12,80	1,48	1,23	12,80	1,51	1,25	11,10	1,55	1,28
CDTR GoS-3	117	14,73	1,15	0,93	14,70	1,15	0,94	14,59	1,17	0,99
CDTR GoS-5	117	11,93	1,29	1,12	11,70	1,26	1,12	10,04	1,30	1,16
CDT GoS-3 sem retorno	117	14,61	1,60	1,15	14,59	1,61	1,16	14,50	1,69	1,21
CDT GoS-5 sem retorno	117	8,78	1,62	1,19	8,71	1,63	1,20	8,69	1,71	1,31
CDA GoS-3	119	10,00	2,30	1,48	10,00	2,35	1,59	10,00	2,65	1,61
CDA GoS-5	119	10,00	3,20	3,01	9,50	3,29	3,10	9,39	3,15	3,15
CDT GoS-3	119	10,33	1,65	1,49	9,62	1,65	1,49	9,47	2,04	1,96
CDT GoS-5	119	13,67	2,35	2,25	13,58	2,40	2,30	12,26	2,45	2,38
CDTR GoS-3	119	16,00	1,61	1,37	15,89	1,62	1,38	15,85	1,96	1,50
CDTR GoS-5	119	16,48	2,26	2,00	16,49	2,29	2,06	16,55	2,35	2,15
CDT Gos 3 sem retorno	119	6,78	2,10	0,91	6,64	2,25	0,93	6,62	2,41	0,97
CDT Gos 5 sem retorno	119	6,41	1,55	1,39	6,36	1,56	1,40	6,10	1,61	1,47

# Capítulo 9

## Conclusões

As principais implementações realizadas nesta dissertação podem ser divididas em quatro principais categorias: onde a primeira é a implementação de um codificador que realiza operações simples, de maneira que a maior parte da exploração estatística foi transferida para o decodificador. Isso é viabilizado com a utilização de fundamentos de DSC, o que foi mostrado nas implementações realizadas, as quais são CDA e CDT, possibilitando a utilização em dispositivos embarcados como o *Holter* de ECG; utilização de técnicas de pré-processamento como o tratamento inicial do sinal de ECG assim como a geração de GoS, o que permite tanto melhora de desempenho, quanto a adequação do sinal de ECG a estrutura DSC; codificação de resíduo de ECG, o que permitiu melhora significativa em relação a codificação de ECG completo; por último, a retirada do canal de retorno, possibilitando o envio de uma única vez todos *bits* de paridade.

Primeiramente, a potencialidade da codificação distribuída de ECG foi demonstrada através dos CODECs CDA e CDT, que mostram a viabilidade da aplicação na compressão de sinais de ECG, onde o primeiro, que é a base dos demais, propõe método de codificação direta, e o segundo propõe inclusão de transformada DCT, melhorando assim o desempenho quando comparado ao CDA.

Em segundo lugar, tanto CDA quanto CDT utilizam técnicas de pré-processamento que tem como objetivo realizar tratamento em sinais de ECG possibilitando que segmentos adjacentes possuam mais semelhança, o que é ideal para DSC. Além disso, permite melhor desempenho de compressão, pois realizam a organização do sinal de ECG de maneira que mais redundâncias são formadas. Isso possibilita a adequação do sinal de ECG ao esquema DSC,

pois, permite predições de WZSs com mais qualidade.

Em terceiro, a capacidade de compressão do CDT foi ampliada com aplicação da codificação de resíduo de sinais de ECG, ou seja, somente o resto da diferença entre WZSs e SIs, sendo isso realizado da mesma forma quando utilizado GoS-3 ou GoS-5, o que é feito tanto no codificador na geração de resíduo de WZSs, quanto no decodificador no resíduo de SIs. Os resultados da aplicação desta técnica apresentam melhora significativa de desempenho em relação a codificação do sinal de ECG completo.

Por último, essa dissertação apresenta o potencial da compressão de ECG utilizando técnicas de DSC. Entretanto, essa arquitetura de codificação apresenta uma característica bastante restritiva, que é o canal de retorno, esse que é necessário no controle da taxa de codificação de *bits* de paridade, os quais são ajustados perfeitamente ao canal de correlação por meio de padrões de perfuração. Logo, nessa dissertação foi proposta uma metodologia que permite a retirada desse canal através da estimativa de taxas de perfuração realizadas no perfurador, por meio de análise de variáveis e aproximação de uma função fundamentada em basicamente duas variáveis, a probabilidade de *bits* errados e entropia conjunta de WZSs e SIs, previamente codificados com canal de retorno. Baseado nisso, a eliminação completa desse canal foi fundamentada, entretanto, tal solução tem como característica taxas menores em relação a solução do CDT com canal de retorno, isso reside no fato que a estimativa é uma aproximação da taxa ideal, logo não alcança taxas superiores. Porém, seu ganho está na eliminação de *hardware* referente ao canal de retorno.

Os objetivos intencionados na realização deste trabalho foram alcançados, o que foi mostrado nos CODECs implementados. Contudo, outras melhorias podem ser realizadas, as quais podem apresentar resultados superiores a este trabalho, sendo estas relacionadas abaixo.

- A potencialidade de compressão de sinais de ECG utilizando DSC pode ser melhorada com a utilização de duas ou mais derivações de ECG, considerando que tais derivações possuam similaridade em suas estruturas. De fato, pode-se utilizar uma derivação que forneça informação lateral para uma ou mais derivações correlacionadas.
- A utilização do estimador de *bits* de paridade pode ser estendida para o CDA e CDTR, de maneira que o primeiro realiza a codificação de sinais de ECG diretamente e o segundo realizada codificação residual.

- A técnica de predição de *bits* de paridade, a qual foi implementada no CDT sem canal de retorno, apresenta desempenho satisfatório, porém isso não impede que a predição seja aperfeiçoada por meio de outros métodos.

Como conclusão principal, esse estudo fornece uma metodologia simples e adaptável para diferentes dispositivos alvo, os quais necessitam desde uma solução mais simples (CDA), ou até os que precisam de uma boa relação taxa-distorção (CDT), mesmo com pequeno acréscimo computacional. Além disso, o CDT mostrou resultados expressivos com a codificação residual, portanto, essa solução apresenta-se melhor que o CDT base. Por último, a metodologia sem utilização do canal retorno propõe uma solução satisfatória quando tal canal não é viável.

# Referências Bibliográficas

- [1] MARCHETTI, P. H.; DUARTE, M. Instrumentação em eletromiografia. *Laboratório de Biofísica, Escola de Educação Física e Esporte. São Paulo: Universidade de São Paulo, 2006.*
- [2] ADUR, R. et al. Sistema de processamento de sinais biomédicos: módulo didático de eletroencefalograma. Florianópolis, SC, 2008.
- [3] BERALDO, O. A. *Processamento digital do sinal de eletrocardiograma para aplicação em experimentos de fisiologia cardíaca.* Tese (Doutorado) — Universidade de São Paulo, 2017.
- [4] NICOLAU, J. C.; POLANCZYK, C. A.; PINHO, J. A.; BACELLAR, M. S. d. C.; RIBEIRO, D. G. L.; DARWICH, R. N.; RIBEIRO, A. L. P.; DUNDA, M. M. E.; GERMINI-ANI, H.; FRANÇA, F. F. et al. Diretriz de interpretação de eletrocardiograma de repouso. *Arquivos Brasileiros de Cardiologia, SciELO Brasil, v. 80, p. 1–18, 2003. ISSN 0066782X.*
- [5] SAYOOD, K. *Introduction to data compression.* : Morgan Kaufmann, 2017. 6-10 p. ISBN 9780126208627.
- [6] XIONG, Z.; LIVERIS, A. D.; CHENG, S. Distributed source coding for sensor networks. *IEEE signal processing magazine, IEEE, v. 21, n. 5, p. 80–94, 2004. ISSN 10535888.*
- [7] AARON, A.; ZHANG, R.; GIROD, B. Wyner-ziv coding of motion video. In: *IEEE. Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on.* 2002. v. 1, p. 240–244. ISBN 0780375769.
- [8] LIU, X.; ZHAO, K.; ZHOU, X.; HUANG, H. The study of wyner-ziv image compression algorithm using in wmsns. In: *IEEE. Communications and Mobile Computing (CMC), 2010 International Conference on.* 2010. v. 3, p. 353–357. ISBN 1424463289.

- [9] BRITES, C.; ASCENSO, J.; PEREIRA, F. Studying temporal correlation noise modeling for pixel based wyner-ziv video coding. In: IEEE. *Image Processing, 2006 IEEE International Conference on*. 2006. p. 273–276. ISBN 1424404800.
- [10] VIJAYANAGAR, K. R.; KIM, J.; LEE, Y.; KIM, J. bok. Low complexity distributed video coding. *Journal of Visual Communication and Image Representation*, v. 25, n. 2, p. 361 – 372, 2014. ISSN 1047-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1047320313002204>>.
- [11] MAHMOOD, A.; DOOLEY, L. S.; WONG, P. Transform domain distributed video coding using larger transform blocks. In: *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2017. p. 1–5.
- [12] BRITES, C.; ASCENSO, J.; PEREIRA, F. Improving transform domain wyner-ziv video coding performance. In: IEEE. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. 2006. v. 2, p. II–II. ISBN 142440469X.
- [13] AARON, A.; RANE, S.; GIROD, B. Wyner-ziv video coding with hash-based motion compensation at the receiver. In: *2004 International Conference on Image Processing, 2004. ICIP '04*. 2004. v. 5, p. 3097–3100 Vol. 5. ISSN 1522-4880.
- [14] WU, H.-T.; TSAI, W.-Y.; CHANG, W.-W. Distributed source coding for real-time ecg signal monitoring. *IEICE TRANSACTIONS on Information and Systems*, The Institute of Electronics, Information and Communication Engineers, v. 97, n. 8, p. 2157–2165, 2014. ISSN 17451361.
- [15] WYNER, A.; ZIV, J. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on information Theory*, IEEE, v. 22, n. 1, p. 1–10, 1976.
- [16] FORNEY, G. D. Coset codes. i. introduction and geometrical classification. *IEEE Transactions on Information Theory*, IEEE, v. 34, n. 5, p. 1123–1151, 1988. ISSN 00189448.
- [17] FILHO, E. B. d. L. *Aplicações em Codificação de Sinais: O Casamento Aproximado De Padrões Multiescalas e a Codificação Distribuída de Eletrocardiograma*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2008.

- [18] SLEPIAN, D.; WOLF, J. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, IEEE, v. 19, n. 4, p. 471–480, 1973. ISSN 00189448.
- [19] PRADHAN, S. S.; RAMCHANDRAN, K. Distributed source coding using syndromes (discus): Design and construction. *IEEE transactions on information theory*, IEEE, v. 49, n. 3, p. 626–643, 2003.
- [20] BRONZINO, J. D. *Biomedical engineering handbook*. : CRC press, 1999. ISBN 0849304628.
- [21] MORRIS, F.; BRADY, W. J.; CAMM, A. J. *ABC of clinical electrocardiography*. : John Wiley & Sons, 2009. ISBN 1444312499.
- [22] SUN, C.-C.; TAI, S.-C. Beat-based ecg compression using gain-shape vector quantization. *IEEE transactions on biomedical engineering*, IEEE, v. 52, n. 11, p. 1882–1888, 2005. ISSN 00189294.
- [23] MASSACHUSETTS INSTITUTE TECHNOLOGIE. *What is the electrocardiogram ECG (ECG)??*. 2019. Disponível em: <<https://www.mit.edu>>. Acesso em: 03 jan. 2019.
- [24] Lee, H.; Tillo, T.; Jeon, B. Hybrid wyner-ziv video coding structure for removing feedback channel. In: *2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. 2009. p. 858–862. ISSN 1555-5798.
- [25] Buckley and K. M. Ecg data compression using cut and align beats approach and 2-d transforms. *IEEE Transactions on Biomedical Engineering*, v. 46, n. 5, p. 556–564, May 1999. ISSN 0018-9294.
- [26] FILHO, B. E.; RODRIGUES, N. M.; SILVA, E. A. da; FARIA, S. M. de; SILVA, V. M. da; CARVALHO, M. B. de et al. Ecg signal compression based on dc equalization and complexity sorting. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 55, n. 7, p. 1923–1926, 2008.
- [27] FELDMAN, J.; GOLDWASSER, G. P. Eletrocardiograma: recomendações para a sua interpretação. *Revista da SOCERJ*, v. 14, n. 4, p. 256, 2004.
- [28] RAMOS, Â. P.; SOUSA, B. S. Eletrocardiograma: princípios, conceitos e aplicações. *Centro de Estudos de Fisiologia do Exercício*, 2007.

- [29] KOEPPEN, B. M.; STANTON, B. A. *Berne & Levy Physiology, Updated Edition E-Book*. : Elsevier Health Sciences, 2009.
- [30] BARRETT, K. E.; BARMAN, S. M.; BOITANO, S.; BROOKS, H. L. *Fisiología médica de Ganong*. : AMGH Editora, 2014.
- [31] HILL, J.; TIMMIS, A. Abc of clinical electrocardiography: Exercise tolerance testing. *BMJ: British Medical Journal*, BMJ Publishing Group, v. 324, n. 7345, p. 1084, 2002.
- [32] SCHWARZ, L. Artigo de revisão: eletrocardiograma. *Revista Ilha Digital*, v. 1, p. 3–19, 2009. ISSN 21772649.
- [33] DESPOPOULOS, A.; SILBERNAGL, S. *Color atlas of physiology*. : Thieme, 2003.
- [34] ESCALONA-MORÁN, M.; SORIANO, M. C.; GARCÍA-PRIETO, J.; FISCHER, I.; MIRASSO, C. R. Multivariate nonlinear time-series estimation using delay-based reservoir computing. *The European Physical Journal Special Topics*, Springer, v. 223, n. 13, p. 2903–2912, 2014.
- [35] KOLEKAR, M. H.; KUMAR, V. *Biomedical signal and image processing patient care*. : IGI Global, 2018. 46-65 p. ISBN 9781522528296.
- [36] CAMPITELLI, M. A. Compressão de sinais ecg utilizando dwt com quantização não-linear e por sub-bandas.
- [37] FARIA, L.; FONSECA, L.; COSTA, M. Performance evaluation of data compression systems applied to satellite imagery. *Journal of Electrical and Computer Engineering*, v. 2012, 04 2012.
- [38] Jha, C. K.; Kolekar, M. H. Efficient ecg data compression and transmission algorithm for telemedicine. In: *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*. 2016. p. 1–6. ISSN 2155-2509.
- [39] COX, J.; NOLLE, F.; FOZZARD, H.; OLIVER, G. Aztec, a preprocessing program for real-time ecg rhythm analysis. *IEEE Transactions on Biomedical Engineering*, IEEE, n. 2, p. 128–129, 1968. ISSN 00189294.

- [40] NAVE, G.; COHEN, A. Ecg compression using long-term prediction. *IEEE transactions on Biomedical Engineering*, IEEE, v. 40, n. 9, p. 877–885, 1993.
- [41] JALALEDDINE, S. M.; HUTCHENS, C. G.; STRATTAN, R. D.; COBERLY, W. A. Ecg data compression techniques-a unified approach. *IEEE transactions on Biomedical Engineering*, IEEE, v. 37, n. 4, p. 329–343, 1990. ISSN 00189294.
- [42] MUELLER, W. C. Arrhythmia detection program for an ambulatory ecg monitor. *Biomedical sciences instrumentation*, v. 14, p. 81–85, 1978. ISSN 00678856.
- [43] ABENSTEIN, J. P.; TOMPKINS, W. J. A new data-reduction algorithm for real-time ecg analysis. *IEEE Transactions on Biomedical Engineering*, IEEE, n. 1, p. 43–48, 1982.
- [44] BLANCHARD, S.; BARR, R. Comparison of methods for adaptive sampling of cardiac electrograms and electrocardiograms. *Medical and Biological Engineering and Computing*, Springer, v. 23, n. 4, p. 377–386, 1985. ISSN 01400118.
- [45] SINGH, B.; KAUR, A.; SINGH, J. A review of ecg data compression techniques. *International journal of computer applications*, Foundation of Computer Science, v. 116, n. 11, 2015. ISSN 09758887.
- [46] FIRA, C. M.; GORAS, L. An ecg signals compression method and its validation using nns. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 55, n. 4, p. 1319–1326, 2008. ISSN 00189294.
- [47] MCCAUGHERN, R.; ROSIE, A. M.; MONDS, F. C. Two methods of asynchronous data compression. *Electronics Letters*, v. 4, n. 17, p. 363–364, August 1968. ISSN 0013-5194.
- [48] COHEN, A.; POLUTA, M.; SCOTT-MILLAR, R. Compression of ecg signals using vector quantization. In: *IEEE South African Symposium on Communications and Signal Processing*. 1990. p. 49–54.
- [49] ZHANG, B.; ZHAO, J.; CHEN, X.; WU, J. Ecg data compression using a neural network model based on multi-objective optimization. *PLOS ONE*, v. 12, p. e0182500, 10 2017.
- [50] BENDIFALLAH, A.; BENZID, R.; BOULEMDEN, M. Improved ecg compression method using discrete cosine transform. *Electronics Letters*, v. 47, p. 87 – 89, 02 2011.

- [51] SHARMA, M.; WADHWANI, A. K. Efficient algorithm for ecg coding. *International Journal of Scientific and Engineering Research*, vol. 2, no. 6, p. 67-75, p. 67–75, 06 2011.
- [52] KURNIAWAN, A.; PURBOYO, T. W.; PRASASTI, A. L. Implementation of image compression using discrete cosine transform (dct) and discrete wavelet transform (dwt). *International Journal of Applied Engineering Research*, v. 12, n. 23, p. 13951–13958, 2017.
- [53] Malvar, H. S.; Hallapuro, A.; Karczewicz, M.; Kerofsky, L. Low-complexity transform and quantization in h.264/avc. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, p. 598–603, July 2003. ISSN 1051-8215.
- [54] BERROU, C.; GLAVIEUX, A.; THITIMAJSHIMA, P. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In: *Proceedings of ICC '93 - IEEE International Conference on Communications*. 1993. v. 2, p. 1064–1070 vol.2.
- [55] GOLOMB, S. Run-length encodings (corresp.). *IEEE transactions on information theory*, IEEE, v. 12, n. 3, p. 399–401, 1966. ISSN 00189448.
- [56] HUFFMAN, D. A. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, IEEE, v. 40, n. 9, p. 1098–1101, 1952. ISSN 00968390.
- [57] WITTEN, I. H.; NEAL, R. M.; CLEARY, J. G. Arithmetic coding for data compression. *Communications of the ACM*, ACM, v. 30, n. 6, p. 520–540, 1987. ISSN 00010782.
- [58] ZIV, J.; LEMPEL, A. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, IEEE, v. 23, n. 3, p. 337–343, 1977. ISSN 00189448.
- [59] SHANNON, C. E. A mathematical theory of communication. *Bell system technical journal*, Wiley Online Library, v. 27, n. 3, p. 379–423, 1948. ISSN 00058580.
- [60] GIROD, B.; AARON, A. M.; RANE, S.; REBOLLO-MONEDERO, D. Distributed video coding. *Proceedings of the IEEE*, IEEE, v. 93, n. 1, p. 71–83, 2005.
- [61] PRADHAN, S. S.; RAMCHANDRAN, K. Distributed source coding using syndromes (discus): design and construction. *IEEE Transactions on Information Theory*, v. 49, n. 3, p. 626–643, March 2003. ISSN 0018-9448.

- [62] AARON, A.; ZHANG, R.; GIROD, B. Wyner-ziv coding of motion video. In: *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002*. 2002. v. 1, p. 240–244 vol.1. ISSN 1058-6393.
- [63] AARON, A.; GIROD, B. Compression with side information using turbo codes. In: *Proceedings DCC 2002. Data Compression Conference*. 2002. p. 252–261. ISSN 1068-0314.
- [64] AARON, A.; RANE, S.; ZHANG, R.; GIROD, B. Wyner-ziv coding for video: applications to compression and error resilience. In: *Data Compression Conference, 2003. Proceedings. DCC 2003*. 2003. p. 93–102. ISSN 1068-0314.
- [65] AARON, A.; GIROD, B. Compression with side information using turbo codes. In: *Proceedings DCC 2002. Data Compression Conference*. 2002. p. 252–261. ISSN 1068-0314.
- [66] JASPAR, X.; VANDENDORPE, L. New iterative decoding of variable length codes with turbo-codes. In: *IEEE. 2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577)*. 2004. v. 5, p. 2606–2610.
- [67] BRITES, C. Advances on distributed video coding. *Lisbon, Portugal: Technical University of Lisbon*, 2005.
- [68] ABRANTES, S. A. *Códigos Correctores de Erros em Comunicações Digitais*. : FEUP edições, 2010.
- [69] TUN, H. M.; MOE, W. K.; NAING, Z. M. Analysis on ecg data compression using wavelet transform technique. *International Journal of Psychological and Brain Sciences*, Science Publishing Group, v. 2, n. 6, p. 127, 2017. ISSN 25751573.
- [70] Lee, S.; Kim, J.; Lee, M. A real-time ecg data compression and transmission algorithm for an e-health device. *IEEE Transactions on Biomedical Engineering*, v. 58, n. 9, p. 2448–2455, Sep. 2011. ISSN 0018-9294.
- [71] FATHI, A.; FARAJI-KHEIRABADI, F. Ecg compression method based on adaptive quantization of main wavelet packet subbands. *Signal, Image and Video Processing*, Springer, v. 10, n. 8, p. 1433–1440, 2016.
- [72] ABOY, M.; CRESPO, C.; MCNAMES, J.; BASSALE, J.; JENKINS, L.; GOLDSTEIN, B. A biomedical signal processing toolbox. 01 2002.

- [73] FILHO, E. B. L.; RODRIGUES, N. M. M.; SILVA\*, E. A. B. da; CARVALHO, M. B. de; FARIA, S. M. M. de; SILVA, V. M. M. da. On ecg signal compression with 1-d multi-scale recurrent patterns allied to preprocessing techniques. *IEEE Transactions on Biomedical Engineering*, v. 56, n. 3, p. 896–900, 2009.
- [74] BILGIN, A.; MARCELLIN, M. W.; ALTBACH, M. I. Compression of electrocardiogram signals using jpeg2000. *IEEE Transactions on Consumer Electronics*, v. 49, n. 4, p. 833–840, Nov 2003. ISSN 0098-3063.
- [75] DIVSALAR, D.; POLLARA, F. Turbo codes for pcs applications. In: IEEE. *Proceedings IEEE International Conference on Communications ICC'95*. 1995. v. 1, p. 54–59.
- [76] VUCETIC, B.; YUAN, J. *Turbo codes: principles and applications*. : Springer Science & Business Media, 2012.
- [77] AARON, A.; RANE, S. D.; SETTON, E.; GIROD, B. Transform-domain wyner-ziv codec for video. *Proceedings of SPIE - The International Society for Optical Engineering*, v. 5308, 11 2003.
- [78] BRITES, C.; PEREIRA, F. Correlation noise modeling for efficient pixel and transform domain wyner–ziv video coding. *IEEE Transactions on Circuits and systems for Video Technology*, IEEE, v. 18, n. 9, p. 1177–1190, 2008.
- [79] CARLSON PAUL B. CRILLY, J. C. R. A. B. *Communication Systems*. 2000.
- [80] VATIS, Y.; KLOMP, S.; OSTERMANN, J. Enhanced reconstruction of the quantised transform coefficients for wyner-ziv coding. In: IEEE. *Multimedia and Expo, 2007 IEEE International Conference on*. 2007. p. 172–175.
- [81] VCODEX. *H.264/AVC 4x4 Transform and Quantization*. 2019. Disponível em: <<https://www.vcodex.com/h264avc-4x4-transform-and-quantization/>>. Acesso em: 11 ago. 2019.
- [82] MOODY, G. B.; MARK, R. G. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, IEEE, v. 20, n. 3, p. 45–50, 2001. ISSN 07395175.

- [83] FILHO, E. B. L.; M., N. M.; SILVA\*, E. A. B. da; CARVALHO, M. B. de; FARIA, S. M. M. de; SILVA, V. M. M. da. On ecg signal compression with 1-d multiscale recurrent patterns allied to preprocessing techniques. *IEEE Transactions on Biomedical Engineering*, v. 56, n. 3, p. 896–900, March 2009. ISSN 0018-9294.
- [84] BILGIN, A.; MARCELLIN, M. W.; ALTBACH, M. I. Compression of electrocardiogram signals using jpeg2000. *IEEE Transactions on Consumer Electronics*, IEEE, v. 49, n. 4, p. 833–840, 2003.
- [85] Bilgin, A.; Marcellin, M. W.; Altbach, M. I. Compression of electrocardiogram signals using jpeg2000. *IEEE Transactions on Consumer Electronics*, v. 49, n. 4, p. 833–840, Nov 2003. ISSN 0098-3063.
- [86] DANZIGER, P. Big o notation. *Source internet: <http://www.scs.ryerson.ca/~mth110/Handouts/PD/bigO.pdf>*, Retrieve: April, 2010.
- [87] CHIOU, P. T.; SUN, Y.; YOUNG, G. A complexity analysis of the jpeg image compression algorithm. In: IEEE. *2017 9th Computer Science and Electronic Engineering (CEECE)*. 2017. p. 65–70.
- [88] HU, H.-T.; HSU, L.-Y. Exploring dwt–svd–dct feature parameters for robust multiple watermarking against jpeg and jpeg2000 compression. *Computers & Electrical Engineering*, v. 41, 08 2014.
- [89] AARON, A.; VARODAYAN, D.; GIROD, B. Wyner-ziv residual coding of video. In: *Proceedings Picture Coding Symposium*. 2006.
- [90] WANG, B.; YUAN, G. Compression of ecg data by vector quantization. *IEEE engineering in medicine and biology magazine*, IEEE, v. 16, n. 4, p. 23–26, 1997.

# Apêndice A

## Artigo desenvolvido nesta dissertação

- Souza P. E. B.; Filho, E. B. L.; Junior, W. S. S.; Cordeiro L. C. "Compressão de Sinais de Eletrocardiograma Utilizando Técnicas de Codificação Distribuída". In: Anais do XXXVII Simpósio Brasileiro de Telecomunicações, Setembro de 2019, Rio de Janeiro. SBrT'19, 2019.