

TRABALHO DE GRADUAÇÃO

**ANÁLISE DE CONVERSOR
ANALÓGICO-DIGITAL
BASEADO EM TRANSFORMADA WAVELET**

Isadora Freire Martins

Brasília, dezembro de 2014

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO
ANÁLISE DE CONVERSOR
ANALÓGICO-DIGITAL
BASEADO EM TRANSFORMADA WAVELET

Isadora Freire Martins

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. José Edil Guimarães de Medeiros, _____
ENE/UnB
Orientador

Prof. Eduardo Peixoto Fernandes da Silva, _____
ENE/UnB
Examinador interno

Prof. Sandro Augusto Pavlik Haddad, _____
FGA/UnB
Examinador interno

RESUMO

Este projeto analisa um conversor analógico-digital que propõe uma nova técnica de amostragem baseada nas propriedades da transformada wavelet. A primeira fase do trabalho consistiu em estudar o algoritmo do conversor proposto, e algumas modificações foram sugeridas. Em seguida, foi analisado o efeito da etapa de quantização no processo, além do desempenho do conversor para diferentes sinais de entrada aplicados e diferentes bases e escalas na transformada wavelet. Para essas verificações, foi desenvolvido um conversor digital-analógico. Com os resultados obtidos, foi possível perceber o efeito de cada parâmetro no processo de conversão. Por fim, é discutido o uso dos conversores propostos em aplicações de alta resolução e de baixo consumo.

ABSTRACT

This project consists of the analysis of an analog-to-digital converter which presents a novel wavelet-based sampling technique. First of all, the converter's algorithm was studied and some modifications were suggested. Afterwards, the effect of quantization in the conversion process, as well as the effects of applying different input signals and of computing the wavelet transform for different bases and scales, were analyzed. In order to verify the converter's behavior under these different conditions, a digital-to-analog converter was developed. The results show how the quantization and the changes in the wavelet transform parameters affect the outcome. In the end, the viability of applying the proposed converters for high resolution and low-power applications is discussed.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	OBJETIVOS DO PROJETO	2
1.4	APRESENTAÇÃO DO MANUSCRITO	2
2	REVISÃO BIBLIOGRÁFICA	4
2.1	CONVERSÃO A/D	4
2.2	TRANSFORMADA WAVELET	7
2.3	COEFICIENTE DE LIPSCHITZ	12
3	DESENVOLVIMENTO	16
3.1	ALGORITMO PROPOSTO PARA CONVERSÃO A/D	16
3.2	RECONSTRUÇÃO DO SINAL	18
3.3	PROCEDIMENTOS ADOTADOS NOS TESTES	22
3.3.1	SINAIS DE ENTRADA APLICADOS	22
3.3.2	ESCOLHA DAS ESCALAS NA TRANSFORMADA WAVELET	23
3.3.3	ESCOLHA DAS BASES WAVELET	24
3.3.4	QUANTIZAÇÃO	27
3.3.5	MÉTRICAS DE ERRO	28
4	RESULTADOS E ANÁLISE	30
4.1	RESULTADOS DAS ANÁLISES PARA O SINAL 1	31
4.2	RESULTADOS DAS ANÁLISES PARA O SINAL 2	41
4.3	RESULTADOS DAS ANÁLISES PARA O SINAL 3	53
4.4	RESULTADOS DAS ANÁLISES PARA O SINAL 4	59
4.5	ANÁLISE DOS RESULTADOS	65
5	CONCLUSÃO	67
5.1	TRABALHOS FUTUROS	68
	REFERÊNCIAS BIBLIOGRÁFICAS	70
	ANEXOS	71

I	<i>Scripts</i> DE MATLAB	72
I.1	FUNÇÕES UTILIZADAS	72
I.1.1	CÁLCULO DO COEFICIENTE DE LIPSCHITZ	72
I.1.2	QUANTIZADOR	73
I.1.3	RECONSTRUÇÃO DO SINAL	73
I.1.4	CÁLCULO DE ERROS	74
I.2	SINAIS DE ENTRADA	74
I.2.1	SINAL 1 [1]	74
I.2.2	SINAL 2	75
I.2.3	SINAL 3	75
I.2.4	SINAL 4	76
I.3	ADC	76
I.3.1	AMOSTRAGEM	76
I.3.2	QUANTIZAÇÃO	80
I.4	DAC	80

LISTA DE FIGURAS

2.1	Representação em bloco de um conversor A/D. [2]	4
2.2	Exemplos de (a) um sinal analógico (b) um sinal digital (c) um sinal amostrado por S/H. T é o período dos sinais digital e amostrado. [3]	5
2.3	Diagrama de blocos genérico para um conversor A/D. [3]	5
2.4	(a) Espectro do sinal analógico, com banda f_B . (b) Espectro do sinal amostrado a uma frequência f_S . (c) Espectro do sinal amostrado quando $f_B > f_S/2$, ilustrando o <i>aliasing</i> . (d) Efeito do filtro anti- <i>aliasing</i> . [3]	6
2.5	Efeito ideal da quantização em um ADC de 3 bits. [3]	8
2.6	A transformada wavelet compara o sinal à esquerda com versões da base wavelet em diferentes escalas e posições, mostradas à direita. [4]	9
2.7	Algumas bases wavelet: (a) gaus1 (b) gaus2 (c) Morlet (d) Daubechies (db6). [5]	10
2.8	Para escalas pequenas (esquerda), a base é comprimida e permite analisar detalhes. Para escalas maiores (direita), a base é expandida e permite analisar variações lentas no sinal. [4]	11
2.9	A convolução suaviza o sinal original. A primeira transformada, $W_1 f(u, s)$, é feita com $\psi = -\theta'$, e $W_2 f(u, s)$, com $\psi = \theta''$. [6]	13
2.10	Expoentes de Lipschitz variando de 0.2 até 2 para a função da Eq. 2.22 [5]	15
3.1	Detecção de singularidades e regularidades de um sinal definindo valores de limiar. [1]	18
3.2	Evolução do logaritmo na base 2 do módulo da transformada wavelet de um sinal como o da Figura 3.3 com uma gaussiana de segunda ordem[1].	19
3.3	Sinal de exemplo.	19
3.4	Diagrama de blocos do ADC originalmente proposto [1].	20
3.5	Ilustração do processo de reconstrução. Pn são os polinômios de reconstrução; os pontos de máximos, mínimos e singularidades são indicados por setas, e os pontos de inflexão, por pontos.	21
3.6	Diagrama de blocos do ADC modificado.	23
3.7	Sinal 1, originalmente usado na análise [1]	24
3.8	(a) Sinal 2, sinal arbitrário formado por informações de (b).....	25
3.9	Sinais de eletrocardiograma [7]. (a) Sinal 3 (ecg1). (b) Sinal 4 (ecg2).....	26
3.10	Bases gaussianas utilizadas.....	29
3.11	(a) Bases de Daubechies e biortogonais com 1 <i>vanishing moment</i> . Bases com 2 <i>vanishing moments</i> : (b) Daubechies (c) Biortogonal (d) Biortogonal reversa.....	29

4.1	Identificação de pontos máximos, mínimos e singularidades pelo ADC.....	33
4.2	Identificação de pontos de inflexão pelo ADC.....	34
4.3	Sinal 1: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.....	37
4.4	Sinal 1: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64. Apenas as amplitudes amostradas passaram pelo quantizador.	39
4.5	Sinal 1: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64. Apenas os coeficientes de Lipschitz passaram pelo quantizador.....	40
4.6	Sinal 1: reconstrução para transformadas feitas com as bases db1 e db2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.....	42
4.7	Sinal 1: reconstrução para transformadas feitas com as bases gaus 1 e mexh e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.....	44
4.8	Sinal 1: reconstrução para transformadas feitas com as bases gaus 1 e mexh e escalas de 1 a 64, para o caso sem quantização e para quantização com 4, 8 e 12 bits. Apenas as amplitudes foram quantizadas.	45
4.9	Sinal 1: reconstrução para transformadas feitas com as bases gaus 1 e mexh e escalas de 1 a 64, para o caso sem quantização e para quantização com 4, 8 e 12 bits. Apenas os coeficientes de Lipschitz passaram pelo quantizador.	46
4.10	Sinal 2: reconstrução para transformadas feitas com as bases bior1.1 e bior2.2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.	49
4.11	Sinal 2: reconstrução para transformadas feitas com as bases bior1.1 e bior2.2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.	50
4.12	Sinal 2: resultados da reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.....	51
4.13	<i>Zoom</i> do resultado da Figura 4.11(a), evidenciando a defasagem do sinal reconstruído em relação ao sinal original.	52
4.14	Identificação de pontos de inflexão pelo ADC para o sinal 2.	54
4.15	Sinal 3: reconstrução para transformadas feitas com as bases rbio1.1 e rbio2.2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.	56
4.16	Sinal 3: reconstrução para transformadas feitas com as bases rbio1.1 e rbio2.2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.	57
4.17	Sinal 3: reconstrução para transformadas feitas com as bases bior1.1 e bior2.2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.	58

4.18	Sinal 4: reconstrução para transformadas feitas com as bases rbio1.1 e rbio2.2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.....	62
4.19	Sinal 4: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.....	63
4.20	Sinal 4: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.....	64

LISTA DE TABELAS

3.1	Bases wavelet utilizadas.....	24
4.1	Número de pontos utilizados para reconstruir cada sinal de acordo com as bases aplicadas, para escalas de 16 a 64	30
4.2	Número de pontos utilizados para reconstruir cada sinal de acordo com as bases aplicadas, para escalas de 1 a 64	31
4.3	Erro RMS de reconstrução para o sinal 1 , para escalas de 16 a 64	36
4.4	Erro RMS de reconstrução para o sinal 1 , para escalas de 16 a 64 , quando apenas as amplitudes são quantizadas.....	38
4.5	Erro RMS de reconstrução para o sinal 1 , para escalas de 16 a 64 , quando apenas os coeficientes de Lipschitz são quantizados.	38
4.6	Erro RMS de reconstrução para o sinal 1 , para escalas de 1 a 64	43
4.7	Erro RMS de reconstrução para o sinal 1 , para escalas de 1 a 64 , quando apenas as amplitudes são quantizadas.....	43
4.8	Erro RMS de reconstrução para o sinal 1 , para escalas de 1 a 64 , quando apenas os coeficientes de Lipschitz são quantizados.	43
4.9	Erro RMS de reconstrução para o sinal 2 , para escalas de 16 a 64	47
4.10	Erro RMS de reconstrução para o sinal 2 , para escalas de 16 a 64 . Apenas as amplitudes foram quantizadas.	47
4.11	Erro RMS de reconstrução para o sinal 2 , para escalas de 16 a 64 . Apenas os coeficientes de Lipschitz foram quantizados.....	48
4.12	Erro RMS de reconstrução para o sinal 2 , para escalas de 1 a 64	48
4.13	Erro RMS de reconstrução para o sinal 2 , para escalas de 1 a 64 . Apenas as amplitudes foram quantizadas.	48
4.14	Erro RMS de reconstrução para o sinal 2 , para escalas de 1 a 64 . Apenas os coeficientes de Lipschitz foram quantizados.....	52
4.15	Erro RMS de reconstrução para o sinal 3 , para escalas de 16 a 64	53
4.16	Erro RMS de reconstrução para o sinal 3 , para escalas de 16 a 64 , para quantização das amplitudes.	53
4.17	Erro RMS de reconstrução para o sinal 3 , para escalas de 16 a 64 , para quantização dos coeficientes de Lipschitz.	55
4.18	Erro RMS de reconstrução para o sinal 3 , para escalas de 1 a 64	55
4.19	Erro RMS de reconstrução para o sinal 3 , para escalas de 1 a 64 , para quantização das amplitudes.....	55

4.20	Erro RMS de reconstrução para o sinal 3 , para escalas de 1 a 64 , para quantização dos coeficientes de Lipschitz.	59
4.21	Erro RMS de reconstrução para o sinal 4 , para escalas de 16 a 64	60
4.22	Erro RMS de reconstrução para o sinal 4 , para escalas de 16 a 64 , para quantização apenas das amplitudes.	60
4.23	Erro RMS de reconstrução para o sinal 4 , para escalas de 16 a 64 , para quantização apenas dos coeficientes de Lipschitz.	60
4.24	Erro RMS de reconstrução para o sinal 4 , para escalas de 1 a 64	61
4.25	Erro RMS de reconstrução para o sinal 4 , para escalas de 1 a 64 , para quantização apenas das amplitudes.	61
4.26	Erro RMS de reconstrução para o sinal 4 , para escalas de 1 a 64 , para quantização apenas dos coeficientes de Lipschitz.	61

LISTA DE SÍMBOLOS

Símbolos Latinos

b	Bit
f	Frequência
t	Tempo
v	Tensão
V	Tensão (DC)
A	Amplitude
D	Código digital
N	Número de bits de D

Símbolos Gregos

α	Coefficiente de Lipschitz
ν	Ponto de interesse no sinal avaliado
τ	Instante em que se identifica um ponto de interesse
μ	Suporte no tempo de um dos polinômios de reconstrução

Subscritos

0	Inicial
f	Final
i	Informação relacionada a um ponto de inflexão
n	Número de <i>vanishing moments</i> de uma wavelet
s	Informação relacionada a ponto de singularidade, máximo ou mínimo
B	Largura de banda (<i>Bandwidth</i>)
S	Amostragem (<i>Sampling</i>)
IN	Entrada
REF	Referência

Sobrescritos

* Complexo conjugado

Siglas

A/D	Analógico-Digital
D/A	Digital-Analógico
ADC	Conversor Analógico-Digital
DAC	Conversor Digital-Analógico
MSB	Bit mais significativo
LSB	Bit menos significativo
S/H	Sample-and-Hold
RMS	Valor quadrático médio (<i>root mean square</i>)

Capítulo 1

Introdução

Este projeto é baseado no trabalho desenvolvido pelo aluno José Alberto Alves de Andrade, da Faculdade UnB—Gama, na disciplina “Trabalho de Conclusão de Curso 1” [1], em que é proposto um conversor A/D idealizado para aplicações de alta resolução e baixo consumo a partir da aplicação de propriedades da transformada wavelet no processo de amostragem.

Este primeiro capítulo inclui uma breve contextualização do problema tratado e sua descrição. Em seguida, os objetivos do projeto são definidos e o restante do manuscrito é apresentado.

1.1 Contextualização

No início dos anos 1980, alguns processos que eram feitos de maneira analógica passaram a ser digitais, devido ao desenvolvimento de algoritmos para processamento digital de sinais. Os algoritmos digitais têm duas vantagens principais sobre seus correspondentes analógicos: a imunidade ao ruído e a compactação no armazenamento de dados e, desde então, o processamento digital passou a se popularizar. Por outro lado, os sinais naturais, como imagem e voz, são analógicos. Assim, uma parte importante no processamento de sinais é converter um sinal analógico para a forma digital e também converter o sinal digital novamente para analógico, o que é feito com os conversores analógico-digital (ADCs) e digital-analógico (DACs), respectivamente. A fabricação desses conversores apresenta alguns desafios, como alta velocidade, alta resolução, e baixa dissipação de potência [8, 2], cujas prioridades dependem da aplicação desejada.

O número de amostras necessárias para que se possa reconstruir um sinal que é aplicado em um ADC é um dos fatores que afetam os parâmetros resolução, velocidade e consumo de potência do conversor. Esse número varia de acordo com certas características do sinal analisado, como sua banda no domínio das frequências.

No final dos anos 1980 e início da década de 1990 começaram a ser publicados estudos sobre a transformada wavelet, que tem aplicações em compressão de dados e apresenta propriedades interessantes na detecção de singularidades. O conversor A/D estudado surge da ideia de aplicar tais propriedades da transformada wavelet ao processo de conversão para reduzir o número de

amostras do sinal necessárias para sua reconstrução, a fim de implementar um ADC para aplicações de alta resolução e com baixo consumo de potência.

Resumidamente, esse conversor A/D difere dos conversores mais conhecidos por eliminar do processo de amostragem a escolha arbitrária de pontos que devem ser amostrados; em vez disso, são selecionados apenas alguns pontos de interesse: os pontos críticos do sinal, que são identificados a partir dos coeficientes da transformada wavelet. Como será visto nos capítulos seguintes, o comportamento do sinal em torno desses pontos especiais pode ser caracterizado pelo coeficiente de Lipschitz, que também pode ser estimado com informações provenientes da transformada wavelet. Portanto, o ADC recebe o sinal analógico e o amostra apenas nos pontos críticos identificados, tendo como saída as informações de amplitude e tempo e a estimativa do coeficiente de Lipschitz para cada um desses pontos. Esses dados de saída devem ser suficientes para a reconstrução do sinal, que será realizada por aproximação polinomial. Dessa forma, esse conversor A/D é orientado a eventos, descartando a necessidade de um sinal de sincronização (*clock*) e possibilitando a variação da taxa de amostragem de acordo com o sinal de entrada.

1.2 Definição do problema

O problema abordado é a análise da estratégia de amostragem desenvolvida no trabalho da referência [1], verificando o funcionamento, as propriedades e as possíveis aplicações do conversor A/D proposto.

1.3 Objetivos do projeto

O objetivo do projeto é descrever a nível de sistema o conversor A/D proposto no trabalho da referência [1]. Para isso, será analisado o algoritmo desenvolvido no trabalho citado, buscando avaliar questões deixadas em aberto. São elas:

- Verificar o erro obtido utilizando o algoritmo implementado;
- Analisar a escolha da base wavelet, verificando resultados com outras bases;
- Investigar métodos para definir o melhor valor de limiar (*threshold*) para a detecção de pontos críticos e singularidades;
- Verificar se a aproximação pelo valor médio é a melhor estratégia de aproximação na amostragem dos valores da amplitude e na estimativa do expoente de Lipschitz do sinal.

1.4 Apresentação do manuscrito

O capítulo 2 apresenta uma breve revisão dos conceitos de conversão A/D e também aborda os conceitos e propriedades da transformada wavelet e do coeficiente de Lipschitz necessárias para

o entendimento do projeto. Em seguida, o capítulo 3 apresenta o desenvolvimento da solução para o problema, bem como a metodologia aplicada. Os resultados obtidos são mostrados e analisados no capítulo 4, seguido das conclusões no capítulo 5. Os anexos contêm os *scripts* de MATLAB desenvolvidos.

Capítulo 2

Revisão Bibliográfica

Neste capítulo são abordados os conceitos teóricos necessários para o entendimento do problema tratado e o desenvolvimento de sua solução. A primeira seção trata do processo de conversão analógico-digital; a segunda, da transformada wavelet, contendo definições e as propriedades pertinentes ao escopo do trabalho. Por último, são estudadas características e propriedades do Coeficiente de Lipschitz.

2.1 Conversão A/D

A Figura 2.1, adaptada de Baker [2], mostra um bloco correspondente a um conversor analógico-digital. Na figura, o ADC trabalha com sinais de tensão, sendo V_{REF} a tensão de referência e v_{IN} , a de entrada; D é uma palavra de N bits, em que b_0 é o bit mais significativo (MSB) e b_{N-1} é o bit menos significativo (LSB). Como seu nome já diz e a figura evidencia, a entrada do conversor A/D é um sinal analógico, e sua saída, um sinal digital. A diferença entre um sinal analógico e um sinal digital é que o analógico é definido em um intervalo contínuo de tempo e um intervalo contínuo de amplitudes, enquanto o digital é definido para intervalos discretos de amplitude e de tempo. A Figura 2.2(a) ilustra um sinal analógico, e a Figura 2.2(b), um sinal digital. [3].

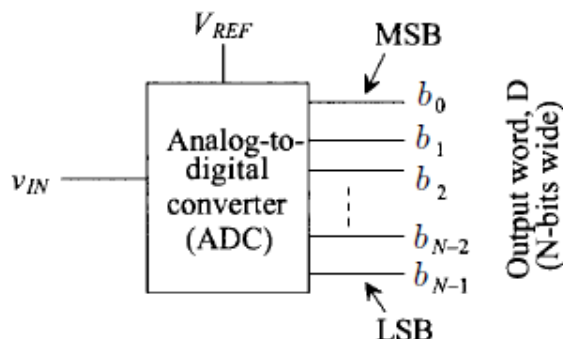


Figura 2.1: Representação em bloco de um conversor A/D. [2]

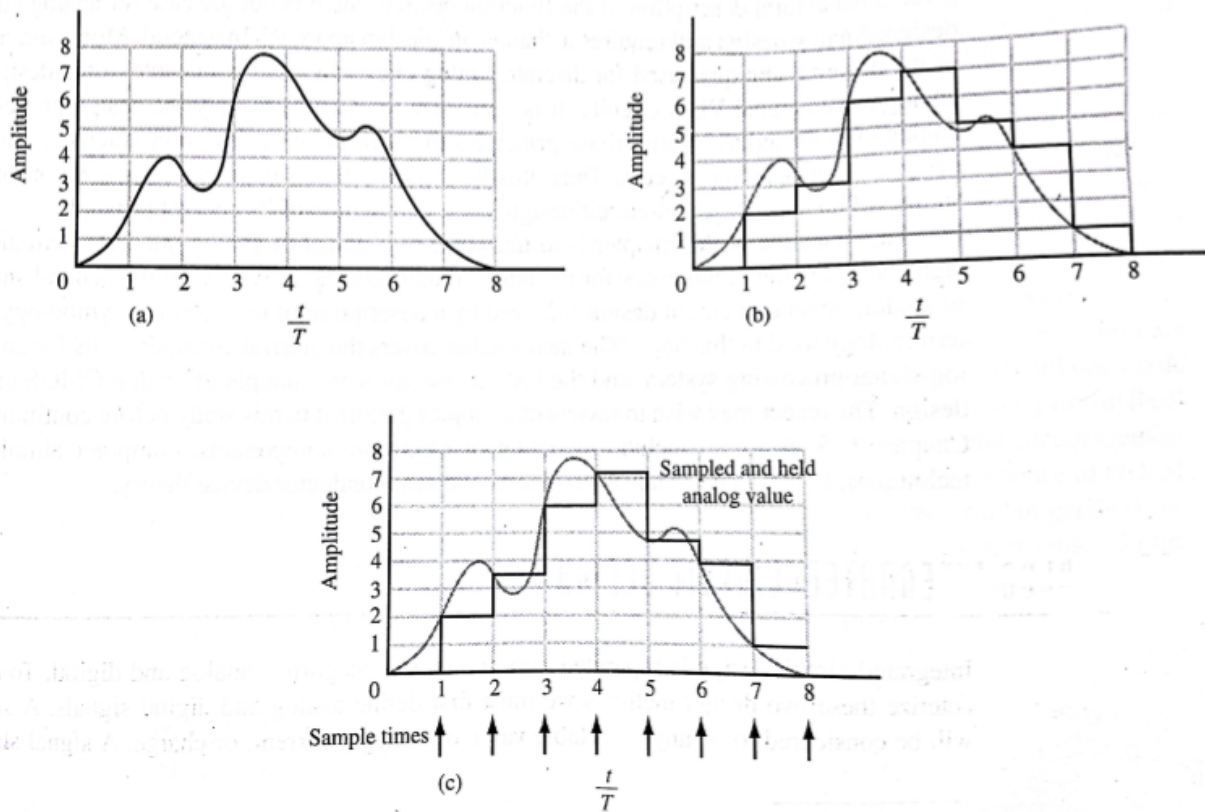


Figura 2.2: Exemplos de (a) um sinal analógico (b) um sinal digital (c) um sinal amostrado por S/H. T é o período dos sinais digital e amostrado. [3]

Outro diagrama de blocos para um conversor A/D é mostrado na Figura 2.3 [3]. O primeiro bloco é um filtro com função de evitar o *aliasing*, e os dois blocos seguintes mostram as etapas principais do processo de conversão A/D: a amostragem e a quantização.

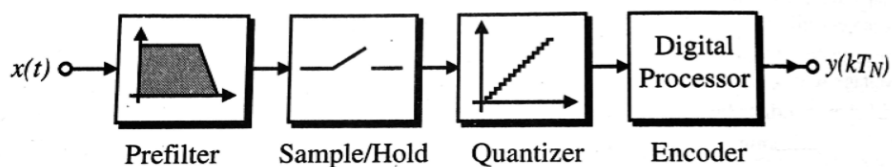


Figura 2.3: Diagrama de blocos genérico para um conversor A/D. [3]

Como a entrada dos ADCs é um sinal analógico e a saída é um sinal digital, é necessária uma etapa de amostragem para converter a entrada com informação contínua na saída com informação discretizada. Um sinal amostrado é definido para valores contínuos de amplitude, mas para instantes discretos no tempo. Normalmente para a amostragem são utilizados circuitos de *sample-and-hold* (S/H), como é o caso indicado no bloco da Figura 2.3. O modo *sample* amostra o sinal analógico, e o modo *hold* mantém o sinal constante no valor amostrado, isto é, o valor da amplitude do sinal analógico correspondente ao instante de amostragem é guardado e esse valor é atribuído ao sinal amostrado por todo o período até o próximo instante de amostragem, como

mostrado no sinal da Figura 2.2(c).

No domínio da frequência, a amostragem é representada pela resposta em frequência do sinal repetida em múltiplos de f_S , que é a frequência de amostragem. Isso é mostrado na Figura 2.4(a) e (b) [3]. Se há superposição dessas respostas em frequência, como na Figura 2.4(c), a informação de altas frequências é alterada pela de baixas frequências, e o sinal recuperado será diferente do sinal original: é o chamado *aliasing* ou dobramento espectral. Para evitar esse fenômeno é usado um filtro anti-*aliasing*, que deve eliminar os componentes que ocasionem a superposição dos espectros. Pela regra de Nyquist, se f_S é a frequência de amostragem e f_B é a banda do sinal, para que não haja *aliasing* e o sinal possa ser recuperado após a amostragem, deve-se ter

$$f_B \leq \frac{f_S}{2} \quad (2.1)$$

Portanto, seguindo a regra de Nyquist, o filtro anti-*aliasing* deve limitar a banda do sinal a $f_S/2$, como mostrado na Figura 2.4(c).

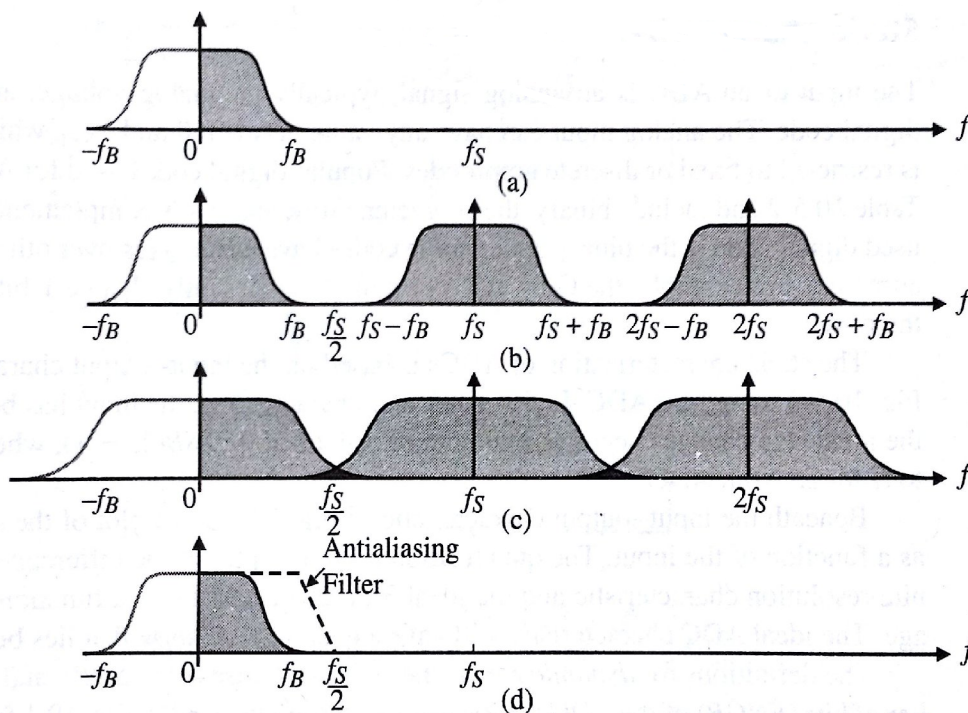


Figura 2.4: (a) Espectro do sinal analógico, com banda f_B . (b) Espectro do sinal amostrado a uma frequência f_S . (c) Espectro do sinal amostrado quando $f_B > f_S/2$, ilustrando o *aliasing*. (d) Efeito do filtro anti-*aliasing*. [3]

Uma das maneiras de classificar os conversores A/D é quanto à frequência de amostragem, que costuma ser fixa. Se essa frequência for próxima à de Nyquist, o conversor é classificado com um A/D de Nyquist. Conversores com taxas de amostragem muito superiores à de Nyquist

são chamados *oversampled ADCs*, ou ADCs sobreamostrados. Tipicamente, os conversores que utilizam sobreamostragem têm resolução maior, mas também maior consumo de potência.

A etapa seguinte à amostragem é a quantização. O quantizador divide a referência do sinal em intervalos e, ao receber as amostras do sinal, as aloca no valor possível—o nível de quantização—mais próximo. A cada nível de quantização é atribuída uma codificação, normalmente binária de N bits. As distâncias entre os níveis de quantização podem ser iguais ou não, mas normalmente tem-se 2^N intervalos uniformes, sendo N o número de bits do código digital de saída. A Figura 2.5 [3] mostra o efeito da quantização para um ADC ideal com codificação de 3 bits. O erro ou ruído de quantização corresponde à diferença entre o valor original (no gráfico, reta de resolução infinita) e o valor atribuído ao nível de quantização. Para o caso ilustrado, o erro máximo corresponde a metade do intervalo entre dois níveis de quantização, isto é, $\pm 0.5LSB$.

Note que o sinal digital pode ser definido também como um sinal quantizado para valores discretos, escrito na forma de uma soma de fatores binários [3]:

$$D = \sum_{i=1}^N b_{N-i} 2^{-i} = b_{N-1} 2^{-1} + b_{N-2} 2^{-2} + \dots + b_0 2^{-N} \quad (2.2)$$

Na equação 2.2, D é o sinal digital e N é o número de bits da palavra. Os coeficientes b_i têm valores binários; tipicamente, 0 ou 1.

2.2 Transformada wavelet

Uma transformada passa um sinal de um domínio para outro. Matematicamente, a transformada é o resultado da convolução de uma base com a função que se deseja analisar. A integral de convolução de um sistema linear invariante ao deslocamento (SLID) com sinal de entrada $x(t)$ e saída $y(t)$ é a seguinte:

$$y(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau \quad (2.3)$$

Na equação 2.3, $h(t)$ é a resposta do sistema ao impulso.

Portanto, se fizermos a convolução entre a base de Fourier— $e^{j\omega t}$ —e a função analisada— $f(t)$ —temos a transformada de Fourier desta função:

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (2.4)$$

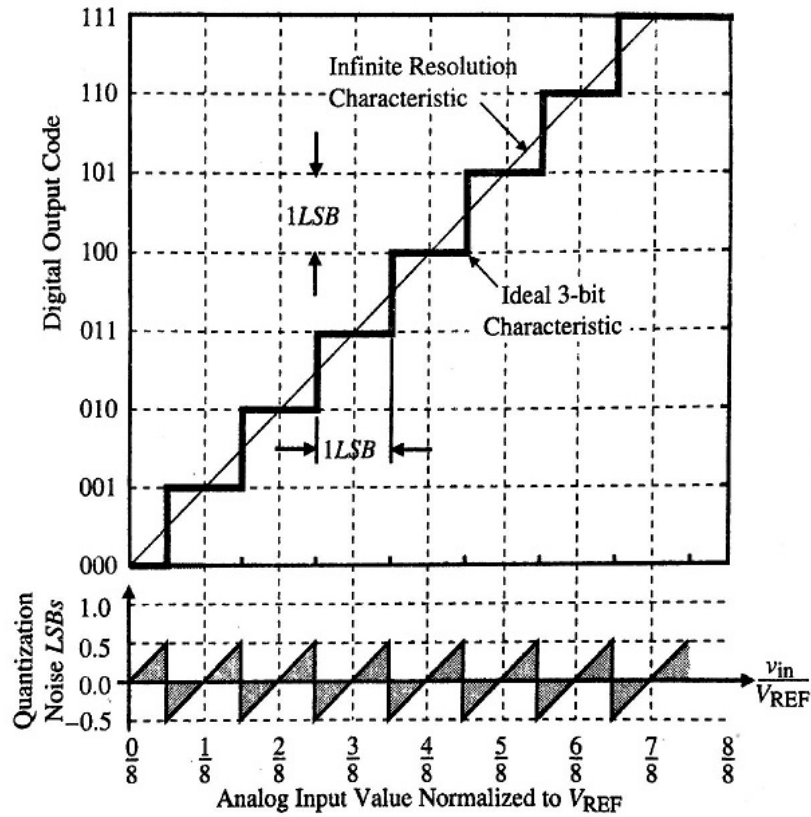


Figura 2.5: Efeito ideal da quantização em um ADC de 3 bits. [3]

Analogamente, a transformada wavelet em tempo contínuo pode ser expressa pela integral do produto interno do sinal— $f(t)$ —com a base escolhida— $\psi(t)$:

$$W_f(\tau, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t - \tau}{a} \right) dt \quad (2.5)$$

Sendo que $a > 0$ e o sobrescrito * indica conjugado complexo, que é retirado da fórmula quando a base é real.

Pela equação 2.5, nota-se que os parâmetros da transformada wavelet são a escala a e a posição τ . Assim, a transformada compara o sinal $f(t)$ com versões da base wavelet $\psi(t)$ deslocadas no tempo e também comprimidas ou expandidas de acordo com o fator de escala aplicado. A Figura 2.6 [4] ilustra esse processo.

A transformada wavelet leva esse nome porque sua base é uma “onda pequena”, um sinal finito e não necessariamente simétrico. Para que uma função possa ser uma base wavelet, ela deve cumprir alguns requisitos:

- Deve ter energia finita, ser oscilatória e ter média zero, obedecendo à Eq. 2.6:

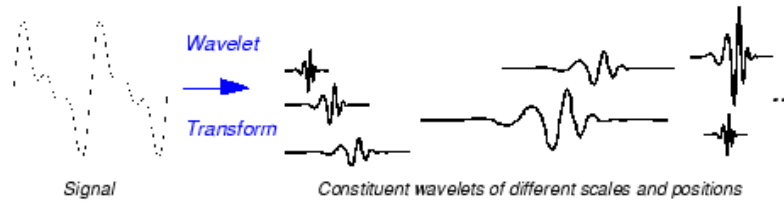


Figura 2.6: A transformada wavelet compara o sinal à esquerda com versões da base wavelet em diferentes escalas e posições, mostradas à direita. [4]

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (2.6)$$

- Sua transformada de Fourier deve ter um componente na frequência zero:

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega = C_{\Psi} < \infty \quad (2.7)$$

Uma consequência da definição da Eq. 2.7 é que as transformadas wavelet podem ser interpretadas como filtros passa-banda no domínio de Fourier [5].

Algumas bases comuns são a primeira e a segunda derivadas da gaussiana (“gaus1” e “gaus2”, respectivamente), a wavelet de Morlet e as wavelets de Daubechies (“db”). Na Figura 2.7 [5] são mostrados exemplos dessas bases.

Ao contrário da transformada de Fourier, que permite resolução do sinal apenas na frequência, a transformada wavelet permite analisar um sinal com resolução tanto no tempo quanto na frequência. Em outras palavras, com o espectro de Fourier de um sinal, temos todas as informações sobre ele no domínio da frequência, mas perdemos as informações no domínio do tempo, enquanto a wavelet permite analisá-lo de acordo com os parâmetros tempo (τ) e escala (a). Conforme varia-se a escala, tem-se resoluções diferentes no tempo e na frequência, permitindo descrever o sinal. [5].

De acordo com [6], a transformada wavelet tem sua resolução no tempo, Δt , e sua resolução na frequência, $\Delta \omega$, relacionadas pelo princípio da incerteza de Heisenberg, condição expressa pelas equações 2.8 a 2.10, nas quais $\Psi(t)$ é a derivada da base wavelet, $\psi(t)$. Assim, quanto maior a resolução na frequência, menor a resolução no tempo, e vice-versa. Por outro lado, a transformada wavelet permite que se varie a escala para estudar o comportamento do sinal em diferentes situações, dependendo do interesse da análise. Uma escala menor corresponde a computar a transformada com a base wavelet comprimida, o que permite analisar detalhes, ou seja, as componentes em altas frequências. Analogamente, para uma escala a grande, a transformada é feita com a base expandida, permitindo verificar características mais gerais, ou as componentes de baixas frequências. Isso é ilustrado na Figura 2.8 [4]. Assim, para avaliar o comportamento do sinal em um ponto

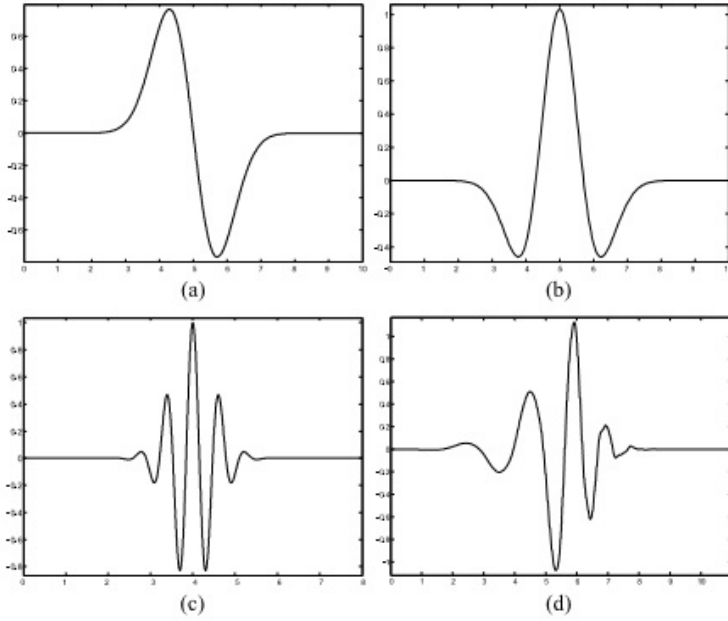


Figura 2.7: Algumas bases wavelet: (a) gaus1 (b) gaus2 (c) Morlet (d) Daubechies (db6). [5]

de interesse, pode-se computar sua transformada wavelet reduzindo gradativamente sua escala, o que implica focar cada vez mais em um detalhe do sinal analisado. Essa propriedade é conhecida como o *wavelet zoom*.

$$\Delta t \Delta \omega > \frac{1}{2} \quad (2.8)$$

$$\Delta t = \sqrt{\frac{\int t^2 |\psi(t)|^2 dt}{\int |\psi(t)|^2 dt}} \quad (2.9)$$

$$\Delta \omega = \sqrt{\frac{\int \omega^2 |\Psi(\omega)|^2 d\omega}{\int |\Psi(\omega)|^2 d\omega}} \quad (2.10)$$

Dessa forma, o *wavelet zoom* pode ser uma ferramenta útil para caracterizar singularidades: sabe-se que, no processamento de sinais, as singularidades e discontinuidades podem armazenar informações relevantes, como, por exemplo, informações de bordas no caso de imagens, sons consonantais no caso da fala, e até transições econômicas bruscas no caso do sistema financeiro [9], então deve ser interessante poder analisar de perto esses pontos especiais.

Pode-se mostrar que a regularidade do sinal se caracteriza pela redução dos coeficientes da transformada wavelet conforme se reduz a escala—afinal, a regularidade deve ter uma correlação



Figura 2.8: Para escalas pequenas (esquerda), a base é comprimida e permite analisar detalhes. Para escalas maiores (direita), a base é expandida e permite analisar variações lentas no sinal. [4]

menor com a base wavelet que o ponto de transição no qual se tem interesse. Portanto, as singularidades podem ser encontradas analisando-se os pontos de máximo local nas diferentes escalas. O conjunto de informações sobre os máximos locais para diferentes escalas permite que se trace uma “linha de máximos” (*maxima line*). Por definição, um *modulus maximum* é um ponto (t_0, a_0) tal que $|Wf(t, a_0)|$ tem um máximo local em $t = t_0$, o que deve implicar [6]

$$\frac{\partial Wf(t_0, a_0)}{\partial t} = 0 \quad (2.11)$$

A *maxima line* mostra os *modulus maxima* ao longo do plano escala-espaco, permitindo visualizar o comportamento do módulo da transformada wavelet para os diferentes instantes e escalas.

Outra propriedade interessante da transformada wavelet aplicada na descrição da regularidade de um sinal são os *vanishing moments*. Um *vanishing moment* é um momento igual a zero, e pode ser expresso matematicamente como mostrado na Eq. 2.12, em que k é a ordem do momento.

$$m[k] = \int t^k \psi(t) dt = 0 \quad (2.12)$$

Para o caso da transformada wavelet, o número de *vanishing moments* corresponde ao número de zeros em π para a representação da wavelet em transformada Z. Nesse caso, se a wavelet tem n *vanishing moments*, ela tem n zeros em π e pode ser reescrita como [10]:

$$H(z) = \left(\frac{1 + z^{-1}}{2} \right) Q(z) \quad (2.13)$$

Ainda, de acordo com Mallat [6], se a wavelet tem n *vanishing moments*, ela pode ser escrita como a n -ésima derivada de outra função θ e a sua transformada wavelet funciona como um operador diferencial multiescala de ordem n , como mostrado no Teorema 1, em que u é o parâmetro posição e s é a escala.

Teorema 1 *Uma wavelet ψ com decaimento rápido tem n vanishing moments se e somente se existe uma função θ de decaimento rápido tal que*

$$\psi(t) = (-1)^n \frac{d^n \theta(t)}{dt^n} \quad (2.14)$$

Consequentemente,

$$Wf(u, s) = s^n \frac{d^n}{du^n} (f * \bar{\theta}_s)(u) \quad (2.15)$$

$$\bar{\theta}_s = s^{-1/2} \theta(-t/s) \quad (2.16)$$

Ainda, ψ não tem mais de n vanishing moments se e somente se $\int_{-\infty}^{\infty} \theta(t) dt \neq 0$.

Portanto, se a base wavelet escolhida tiver $n = 1$ vanishing moments, a transformada wavelet funcionará como um operador diferencial de primeira ordem e os *modulus maxima* deverão identificar pontos de descontinuidade; além disso, os pontos onde o módulo da transformada for igual a zero corresponderão aos pontos críticos do sinal, isto é, seus máximos e mínimos locais. Analogamente, para uma base com $n = 2$ vanishing moments, a transformada corresponde a um operador diferencial de segunda ordem, e os *modulus maxima* identificam pontos de curvatura máxima, enquanto os pontos onde $|Wf(u, s)| = 0$ indicam os pontos de inflexão do sinal f . Essas propriedades são mostradas na Figura 2.9 [6].

Os *vanishing moments* têm aplicações que são dependentes do conhecimento das propriedades do coeficiente de Lipschitz na descrição de regularidades, e por isso serão explicadas na seção seguinte.

2.3 Coeficiente de Lipschitz

O coeficiente de Lipschitz (conhecido também como coeficiente de Hölder, na Matemática) analisa a regularidade do sinal e tem aplicações na sua caracterização, classificação e identificação [5]. Ele pode ser usado para analisar a regularidade do sinal em um intervalo ou em um ponto específico. Se o sinal tem uma singularidade em um ponto ν , isto é, não é diferenciável nesse ponto, a singularidade é caracterizada pelo coeficiente de Lipschitz em ν .

Definição 1 *Uma função f é Lipschitz $\alpha \geq 0$ em um ponto ν se existe $K > 0$ e um polinômio p_ν de grau $m = \lfloor \alpha \rfloor$ tais que, para qualquer t real,*

$$|f(t) - p_\nu(t)| \leq K |t - \nu|^\alpha \quad (2.17)$$

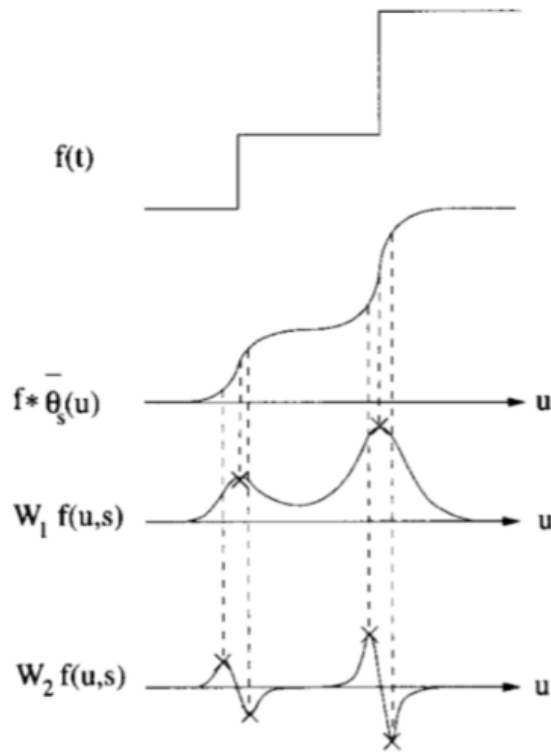


Figura 2.9: A convolução suaviza o sinal original. A primeira transformada, $W_1 f(u, s)$, é feita com $\psi = -\theta'$, e $W_2 f(u, s)$, com $\psi = \theta''$. [6]

A função é uniformemente Lipschitz α em $[a, b]$ se satisfaz a equação 2.17 para todo $\nu \in [a, b]$, sendo K independente de ν .

A regularidade Lipschitz de f em ν ou no intervalo $[a, b]$ é o supremum de α tal que f é Lipschitz α .

Na definição acima, o polinômio de aproximação é único para cada ponto ν [6]. Esse polinômio de aproximação pode ser o polinômio de Taylor: para um polinômio m vezes diferenciável, $m = \lfloor \alpha \rfloor$, a expansão de Taylor de f em ν é expressa pela Eq. 2.18.

$$p_\nu = \sum_{k=0}^{m-1} \frac{f^{(k)}(\nu)}{k!} (t - \nu)^k \quad (2.18)$$

Ainda, o erro de aproximação é definido como

$$\epsilon_\nu(t) = f(t) - p_\nu(t) \quad (2.19)$$

e deve satisfazer

$$\forall t \in \mathbb{R}, \quad |\epsilon_\nu(t)| \leq \frac{|t - \nu|^m}{m!} \sup_{u \in [\nu-h, \nu+h]} |f^m(u)| \quad (2.20)$$

Assim, quando t tende a ν , m gera um limite superior para o erro da aproximação de Taylor com f na vizinhança de ν . De fato, aplicando a Definição 1 para o caso em que $p_\nu(t)$ é o polinômio de Taylor, esse limite superior é expresso em função do coeficiente de Lipschitz, α , e depende da constante K :

$$|\epsilon_\nu(t)| = |f(t) - p_\nu(t)| \leq K|t - \nu|^\alpha \quad (2.21)$$

Como um exemplo para análise, seja f um sinal tal que

$$f(t) = |1 - t|^\alpha \quad (2.22)$$

Na Figura 2.10 é mostrado esse sinal para α assumindo valores entre 0.2 e 2. O sinal é regular com Lipschitz α nos intervalos $0 \leq t < 1$ e $1 < t \leq 2$, mas apresenta uma singularidade em $t = 1$: à esquerda da singularidade, o sinal tem comportamento crescente, e à direita, decrescente. Assim, conhecendo-se o instante em que há singularidade, bem como o valor do coeficiente de Lipschitz e o comportamento do sinal em cada intervalo, tem-se informação suficiente para descrever o sinal. Em outras palavras, o coeficiente de Lipschitz permite compactar a informação do sinal em torno de seu ponto de singularidade. [5]

A regularidade Lipschitz do sinal em um ponto ν depende do decaimento em escalas pequenas do módulo da transformada wavelet na vizinhança de ν [6]. Por sua vez, o decaimento do módulo da transformada na vizinhança de ν é controlado pelo decaimento dos *modulus maxima* que estão no cone de influência $|u - \nu| \leq Cs$. A função f é Lipschitz α se e somente se existe $A > 0$ tal que cada *modulus maximum* no cone satisfaz

$$|Wf(u, s)| \leq As^{\alpha + \frac{1}{2}} \quad (2.23)$$

Tirando o logaritmo dos dois lados,

$$\log_2 |Wf(u, s)| \leq \log_2 A + \left(\alpha + \frac{1}{2} \right) \log_2 s \quad (2.24)$$

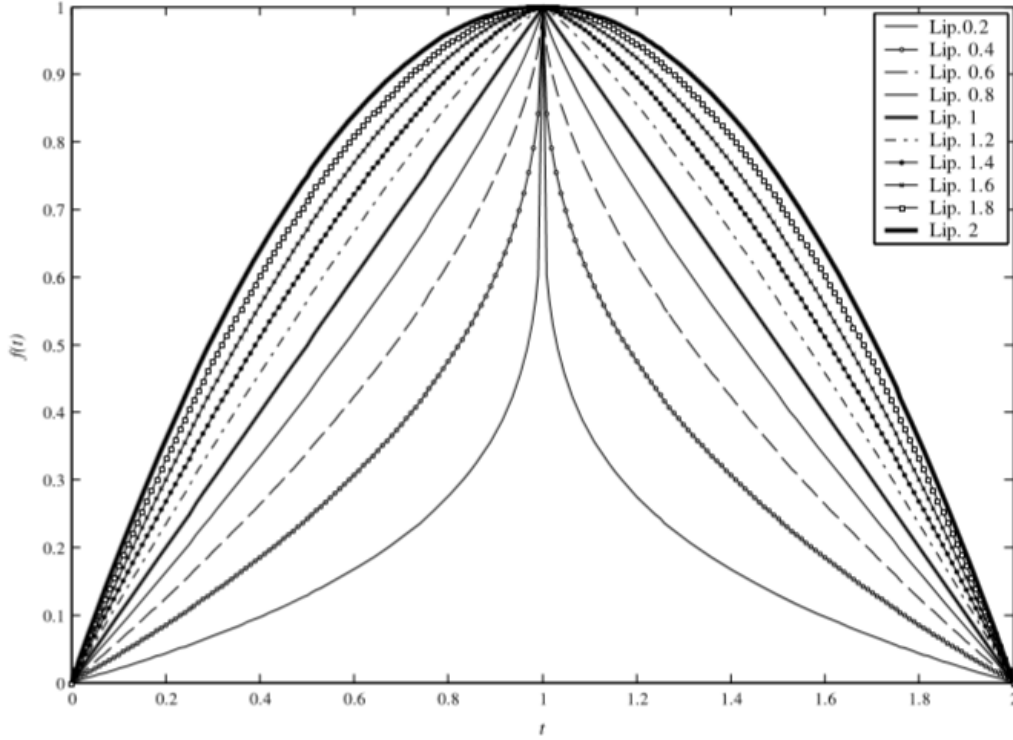


Figura 2.10: Exponentes de Lipschitz variando de 0.2 até 2 para a função da Eq. 2.22 [5]

Pela Eq. 2.24, conclui-se que a regularidade Lipschitz em ν corresponde à máxima inclinação de $\log_2|Wf(u, s)|$ como uma função de $\log_2 s$ pelas linhas de máximo convergindo para ν [5].

Por outro lado, a propriedade de supressão do polinômio dos *vanishing moments* indica que, para calcular o coeficiente de Lipschitz de um sinal, é necessário utilizar como base uma wavelet que tenha $n > \alpha$ *vanishing moments*

$$\int_{-\infty}^{\infty} t^k \psi(t) dt = 0 \quad , \quad 0 \leq k < n \quad (2.25)$$

Na aproximação polinomial do sinal como $f(t) = p_\nu(t) + \epsilon_\nu(t)$, a transformada wavelet de $f(t)$ resulta

$$Wf(u, s) = Wp_\nu(t) + W\epsilon_\nu(t) = W\epsilon_\nu(t) \quad (2.26)$$

o que significa que a transformada wavelet estima α ignorando p_ν (supressão do polinômio). Essa propriedade é interessante porque, pela Eq. 2.21, o coeficiente de Lipschitz está relacionado ao erro de aproximação do polinômio de Taylor.

Capítulo 3

Desenvolvimento

O ponto de partida deste projeto foi o algoritmo desenvolvido até então pelo aluno José Alberto, da Faculdade UnB—Gama, na disciplina “Trabalho de Conclusão de Curso 1” [1]. Portanto, a primeira tarefa realizada foi a análise do algoritmo proposto a fim de avaliar o desempenho do ADC descrito. Para atingir esse objetivo, foi necessário desenvolver um DAC próprio para essa aplicação, o que também permitiu testar mudanças no algoritmo inicial.

Com o conversor D/A pronto, o sinal de entrada pôde ser reconstruído, permitindo avaliar o desempenho do conversor A/D. Os seguintes casos foram considerados, de acordo com os objetivos do projeto:

- Aplicar diferentes sinais de entrada ao conversor proposto, incluindo sinais naturais. Foram escolhidos dois sinais de eletrocardiograma, disponíveis no endereço <http://www.librow.com/cases/case-2> [7];
- Utilizar diferentes bases e escalas na transformada Wavelet;
- Incluir uma etapa de quantização e repetir as análises, comparando os resultados.

As simulações foram feitas em ambiente MATLAB/Simulink, com auxílio do *Wavelet Toolbox*. Para a aplicação desejada, foi escolhida a opção *Continuous Wavelet 1-D* (transformada wavelet contínua 1-D), que equivale a utilizar a função `cwt` na linha de comando.

Neste capítulo, os algoritmos adotados na solução do problema são explicados. A primeira seção define e avalia o algoritmo inicial da referência [1], propondo uma modificação. As seções seguintes tratam das soluções desenvolvidas neste projeto, apresentando o algoritmo para o DAC proposto e explicando detalhes dos diferentes testes realizados.

3.1 Algoritmo Proposto para Conversão A/D

O algoritmo proposto no projeto anterior [1] se baseia na reconstrução de sinais a partir da informação de pontos críticos, como é feito nas disciplinas de Cálculo, identificando máximos e

mínimos da função igualando sua primeira derivada a zero, e seus pontos de inflexão, onde há mudança de concavidade, igualando sua segunda derivada a zero.

Como foi visto no Capítulo 2, a transformada wavelet pode ser tratada como um operador diferencial de ordem n , sendo n o número de *vanishing moments* da base wavelet. Por essa lógica, escolhendo uma base com $n = 1$, é possível identificar os pontos de máximos e mínimos locais como aqueles em que o coeficiente da transformada vale zero; para identificar os pontos de inflexão, deve ser utilizada uma base com $n = 2$ e identificar também os pontos para os quais o coeficiente da transformada wavelet vale zero. Vale ressaltar que, como mostrado na Figura 2.9, a transformada identifica os pontos críticos do sinal original suavizado por uma função θ , o que significa que é possível identificar também pontos de singularidade e descontinuidade com a análise da derivada primeira, isto é, com a base de $n = 1$ *vanishing moments*.

O primeiro método escolhido para identificar os pontos críticos foi verificar os pontos em que, para cada escala, o coeficiente da transformada fica próximo de zero, de acordo com um *threshold*—limiar— ρ : se $|Wf(u, s)| < \rho$, o nível lógico da saída é alto; caso contrário, o nível lógico é baixo. Para unificar os sinais de detecção de cada escala, é utilizada a operação lógica OU, isto é, se é identificada uma região em torno de um ponto crítico em apenas uma escala, essa informação faz parte dos sinais de detecção de pontos críticos, e a posição exata de cada um desses pontos é aproximada com a média dos pontos no intervalo identificado. Um exemplo é mostrado na Figura 3.1 [1]. No entanto, a definição de valores de limiar é empírica e representa uma dificuldade de implementação do algoritmo apresentado. Para contornar esse problema, tendo como base a propriedade do *wavelet zoom*, os pontos de singularidade passaram a ser identificados como os pontos em que a linha de coeficientes da transformada wavelet de primeira ordem para a **menor escala** utilizada tem valor igual a zero. Analogamente, os pontos de inflexão devem ser aqueles em que a linha de coeficientes da transformada wavelet de segunda ordem para a **maior escala** utilizada cruza o eixo horizontal. Assim, o novo processo de detecção de pontos elimina a necessidade de valores de *threshold* e da aproximação das posições de pontos críticos pelo cálculo da média em um intervalo, e pode ser implementado com um comparador.

Tendo a informação da posição dos pontos críticos do sinal, ainda é necessário amostrar as amplitudes e calcular o coeficiente de Lipschitz nos pontos de máximo e mínimo e nas singularidades, para que se possa descrever o comportamento do sinal em torno desses pontos. A partir da Eq. 2.24, repetida abaixo, pode-se estimar o valor do coeficiente de Lipschitz em um ponto fixando-se a sua posição u e derivando-se o logaritmo da magnitude da transformada com relação ao logaritmo das escalas, e em seguida subtraindo-se $1/2$ do resultado.

$$\log_2|Wf(u, s)| \leq \log_2 A + \left(\alpha + \frac{1}{2}\right) \log_2 s$$

A estimativa do coeficiente de Lipschitz é feita utilizando os coeficientes da transformada wavelet com $n = 2$ *vanishing moments*. A Figura 3.2 [1] mostra, para um sinal com comportamento semelhante ao da Figura 3.3, a evolução dos logaritmos na base 2 da escala e do módulo da

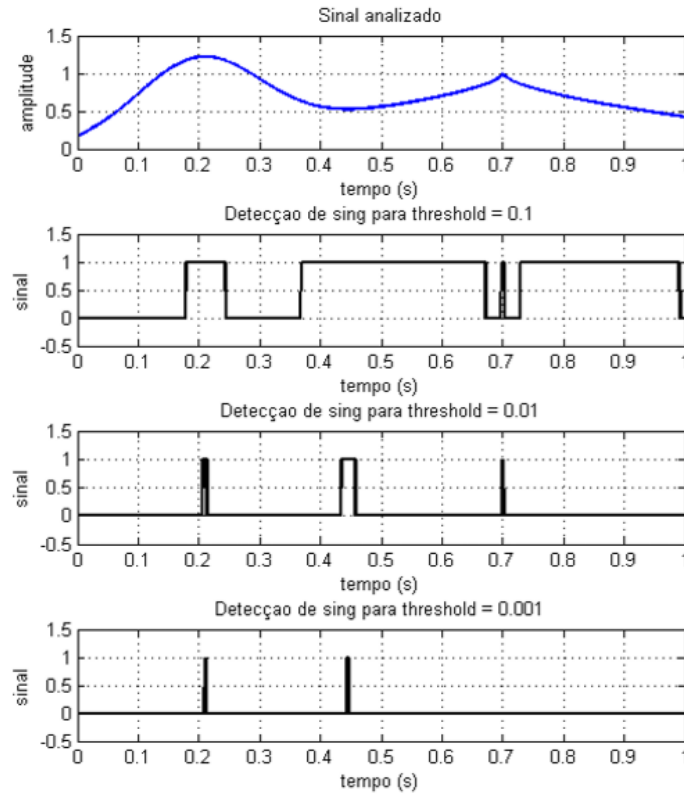


Figura 3.1: Detecção de singularidades e regularidades de um sinal definindo valores de limiar. [1]

transformada wavelet feita com uma base gaussiana de segunda ordem. Nesse caso, a estimativa do coeficiente de Lipschitz em um ponto específico do sinal é feita a partir da inclinação da evolução do logaritmo das escalas, que é destacada nas retas em vermelho. A inclinação até os primeiros 600 pontos é maior que nos pontos restantes, indicando um coeficiente de Lipschitz menor na porção final do sinal, isto é, comportamento menos regular nesse trecho.

Em resumo, as informações resultantes da conversão A/D serão as posições dos pontos críticos—máximos e mínimos locais, inflexões—, além da amplitude e da estimativa do coeficiente de Lipschitz apenas para os máximos e mínimos locais, que incluem pontos de singularidade. O diagrama de blocos do conversor proposto originalmente é mostrado na Figura 3.4 [1].

3.2 Reconstrução do Sinal

Com o algoritmo de conversão A/D definido, ele deve ser testado. Para tanto, foi necessário desenvolver um DAC. A proposta inicial do trabalho da referência [1] é a reconstrução polinomial do sinal utilizando como base a função da Eq 2.22, que será repetida a seguir e cujo comportamento foi mostrado na Figura 2.10.

$$f(t) = |1 - t|^\alpha$$

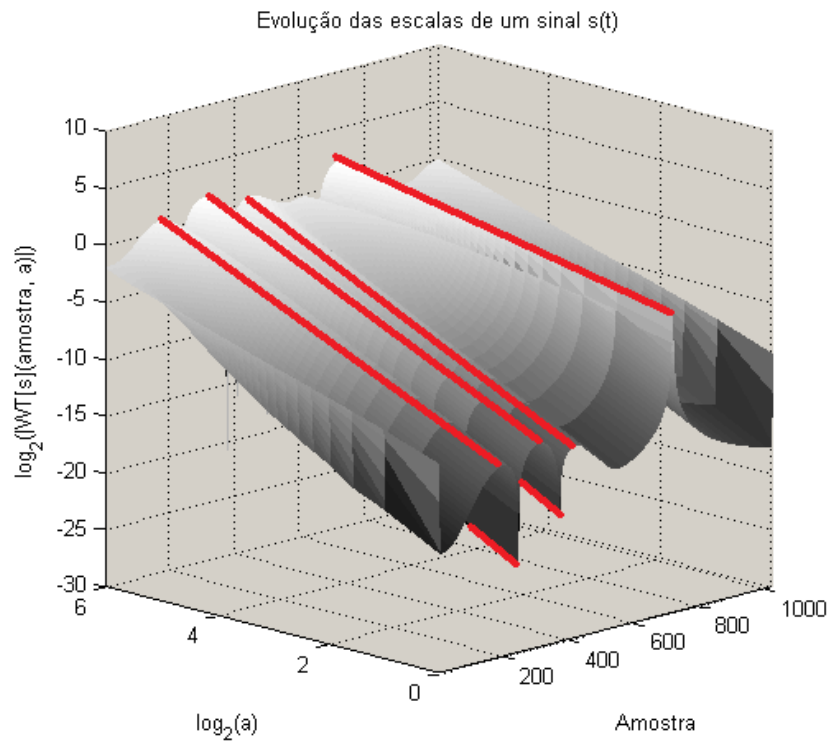


Figura 3.2: Evolução do logaritmo na base 2 do módulo da transformada wavelet de um sinal como o da Figura 3.3 com uma gaussiana de segunda ordem[1].

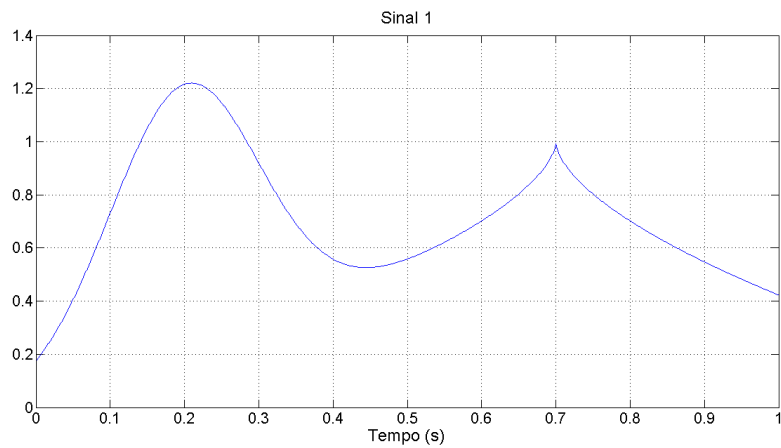


Figura 3.3: Sinal de exemplo.

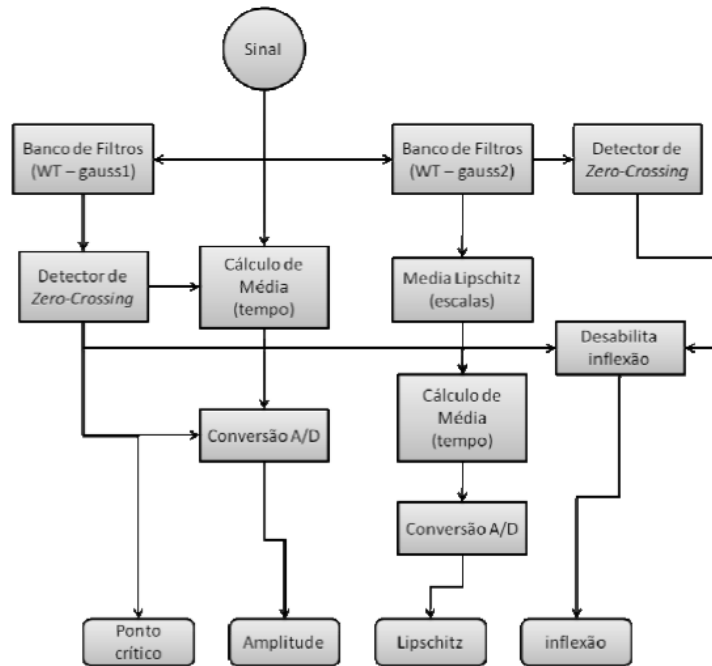


Figura 3.4: Diagrama de blocos do ADC originalmente proposto [1].

A estratégia de reconstrução consiste em utilizar as informações de singularidades e inflexões fornecidas pelo conversor A/D para aproximar o sinal original por trechos, isto é, entre dois pontos identificados, o comportamento do sinal é aproximado pelo sinal da Figura 2.10 a partir do valor estimado para seu coeficiente de Lipschitz. O ADC fornece informação das posições dos pontos críticos do sinal, portanto, neste algoritmo de reconstrução, os dois pontos identificados como os limites de cada trecho devem ser um ponto de inflexão e um ponto de máximo, mínimo ou singularidade. Além disso, o ADC fornece informação da amplitude e do coeficiente de Lipschitz apenas nos pontos de máximos, mínimos e singularidades. No entanto, para a estratégia de reconstrução aplicada, é necessário saber a amplitude do sinal nos pontos de inflexão; por isso, o algoritmo do ADC foi modificado para que a amplitude do sinal fosse amostrada não só nos pontos de máximos e mínimos locais, mas também nos pontos de inflexão identificados. O esboço na Figura 3.5 ilustra o processo para um sinal de exemplo, destacando os pontos utilizados.

Algumas aproximações foram necessárias para tornar o código em MATLAB genérico e permitir avaliação do processo para diferentes bases e sinais de entrada. Pela Figura 3.5, nota-se que também é necessário incluir os pontos inicial e final do sinal de entrada do ADC, que são considerados pontos de máximo ou mínimo locais e, portanto, devem ter suas amplitudes amostradas e o coeficiente de Lipschitz estimado. Ainda, o DAC proposto requer que haja um ponto de inflexão entre um máximo e um mínimo locais identificados, pois consiste em reconstruir o sinal entre um máximo ou mínimo local e um ponto de inflexão por aproximação polinomial. Por isso, se um ponto de inflexão for detectado na mesma posição que um ponto de singularidade, a posição de inflexão deverá ser desconsiderada. Caso o algoritmo de identificação de pontos críticos reconheça mais de um ponto de inflexão entre um máximo e um mínimo local, o algoritmo faz a média das posições dos pontos de inflexão detectados e considera essa a posição do único ponto de inflexão no trecho.

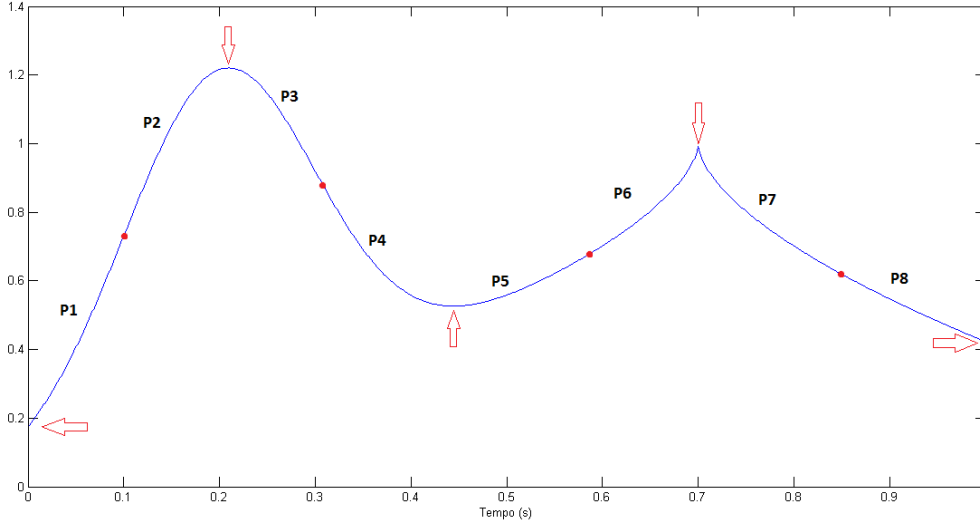


Figura 3.5: Ilustração do processo de reconstrução. P_n são os polinômios de reconstrução; os pontos de máximos, mínimos e singularidades são indicados por setas, e os pontos de inflexão, por pontos.

Por outro lado, se não for detectado nenhum ponto de inflexão entre um máximo e um mínimo local, o algoritmo considera que ocorre um ponto de inflexão na posição média do trecho analisado.

Além disso, para que a função da Eq. 2.22 possa ser utilizada como um polinômio de reconstrução, é necessário modificá-la a fim de incluir o **deslocamento no tempo** e também o **deslocamento na amplitude**. A função de aproximação tem a forma mostrada na Eq. 3.1.

$$P_n(t) = A_i + (A_s - A_i) \left(1 - \left| \frac{\tau - t}{\mu} \right|^{\alpha_s} \right) \quad , \quad t_0 \leq t \leq t_f \quad (3.1)$$

Sendo:

- A_i : a amplitude do sinal amostrada no ponto de inflexão;
- A_s : a amplitude do sinal amostrada no ponto de máximo ou mínimo local;
- t_0 : instante inicial do trecho que será reconstruído por $P_n(t)$. Deve ser um ponto de máximo ou mínimo local, ou um ponto de inflexão;
- t_f : instante final do trecho que será reconstruído por $P_n(t)$. Deve ser um ponto de máximo ou mínimo local, ou um ponto de inflexão;
- τ : instante do ponto de máximo ou mínimo local, que deve ser ou t_0 ou t_f ;
- μ : o suporte de $P_n(t)$, isto é, $\mu = t_f - t_0$;
- α_s : o coeficiente de Lipschitz calculado no máximo ou mínimo local analisado.

Na Eq. 3.1, o deslocamento no tempo é realizado incluindo os parâmetros τ e μ à expressão original da Eq. 2.22. Para o deslocamento na amplitude, a expressão, que originalmente tinha valor pico-a-pico igual a 1, é multiplicada pela diferença entre os pontos extremos do trecho avaliado: $(A_s - A_i)$. O sinal de $(A_s - A_i)$ permite também identificar a concavidade da função no trecho: se $A_s > A_i$, esse sinal é positivo, indicando um máximo local no trecho avaliado, e o resultado da reconstrução no trecho deve ser côncavo para baixo; de maneira análoga, se $A_s < A_i$, o sinal de $(A_s - A_i)$ é negativo, indicando um mínimo local, e os polinômios de reconstrução em torno deste mínimo devem representar uma função côncava para cima. Finalmente, soma-se a amplitude do ponto de inflexão a esse resultado, garantindo que, ao contrário do que ocorre na expressão original, um dos extremos do polinômio no trecho não seja sempre igual a zero.

Com base no exposto nesta seção e na anterior, nota-se que são feitas algumas modificações no algoritmo do conversor A/D originalmente proposto para que a reconstrução por aproximação polinomial possa ser realizada da maneira descrita acima. Essas modificações foram:

- Substituir o uso de valores de limiar na identificação dos pontos críticos do sinal pela detecção dos *zero-crossings* das linhas de maior e menor escalas usando um comparador;
- Passar a amostrar a amplitude do sinal também nos pontos de inflexão detectados;
- O algoritmo de reconstrução só permite um ponto de inflexão detectado entre um máximo e um mínimo locais. Dessa forma, são feitas aproximações nos seguintes casos:
 - Se mais de um ponto de inflexão for detectado em um trecho, é considerado que há apenas um ponto de inflexão, cuja posição é a média das posições dos pontos originalmente identificados;
 - Se não for identificado nenhum ponto de inflexão em um trecho, é considerado que há um ponto de inflexão na posição média do trecho entre o máximo e o mínimo locais.

O diagrama de blocos do ADC considerando as modificações feitas ao algoritmo do conversor A/D originalmente proposto é o mostrado na Figura 3.6.

3.3 Procedimentos adotados nos testes

Nesta seção são detalhados os parâmetros utilizados nos testes com vistas a avaliar o funcionamento do conversor A/D.

3.3.1 Sinais de entrada aplicados

Foram utilizados quatro sinais de teste para avaliar o desempenho do conversor A/D. Para facilitar o processamento, todos os sinais foram normalizados no tempo, tendo duração de 0 a 1 segundo. São eles:

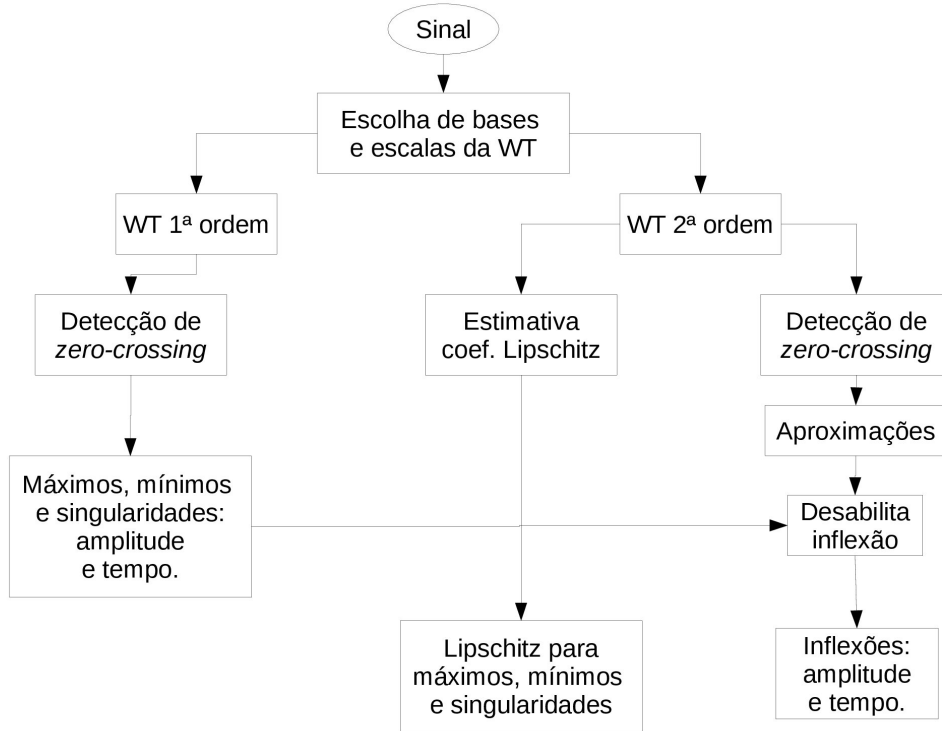


Figura 3.6: Diagrama de blocos do ADC modificado.

- O sinal originalmente utilizado [1], denominado daqui em diante **sinal 1**, mostrado na Figura 3.7 e definido pela expressão

$$s(t) = e^{-\frac{1}{2}\left(\frac{-0.2}{0.1}\right)^2} + \frac{(0.75^{0.6} - |0.7 - t|^{0.6})}{0.75^{0.6}} \quad (3.2)$$

- Um sinal arbitrário (**sinal 2**), mostrado na Figura 3.8(a), formado pela informação das duas primeiras colunas da componente vermelha da imagem mostrada na Figura 3.8(b);
- Dois sinais naturais de eletrocardiograma (**sinal 3**, ou **ecg1** e **sinal 4**, ou **ecg2**) [7], mostrados na Figura 3.9.

3.3.2 Escolha das escalas na transformada Wavelet

Considerando-se a situação real de implementação da transformada com bancos de filtros wavelet, primeiro foram feitas simulações com um conjunto de 4 escalas: 16, 32, 48 e 64, que também foram as escalas escolhidas originalmente. Para verificar se o número de escalas afeta a precisão na reconstrução, foram refeitas as simulações com todas as escalas de 1 a 64, ou seja, 64 escalas.

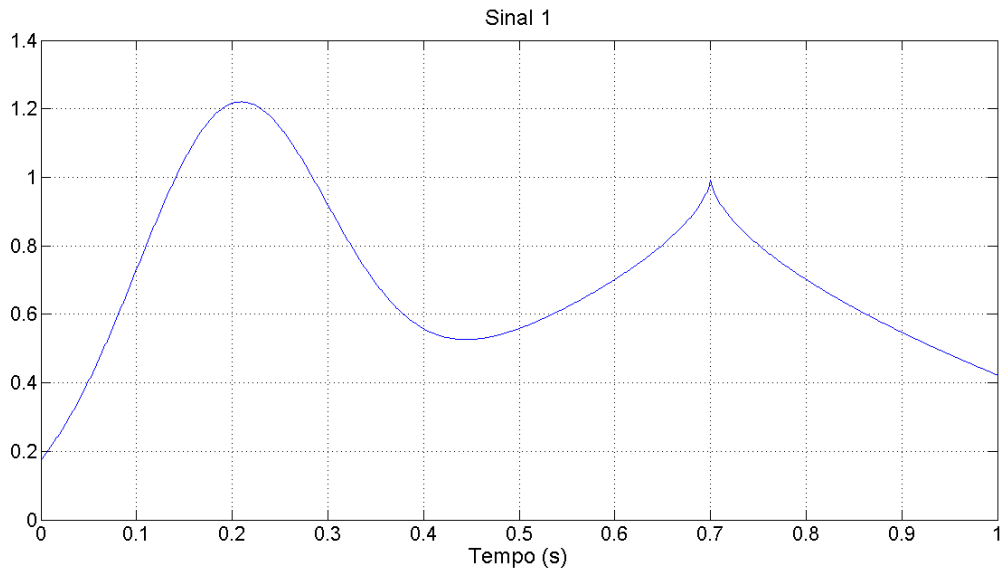


Figura 3.7: Sinal 1, originalmente usado na análise [1]

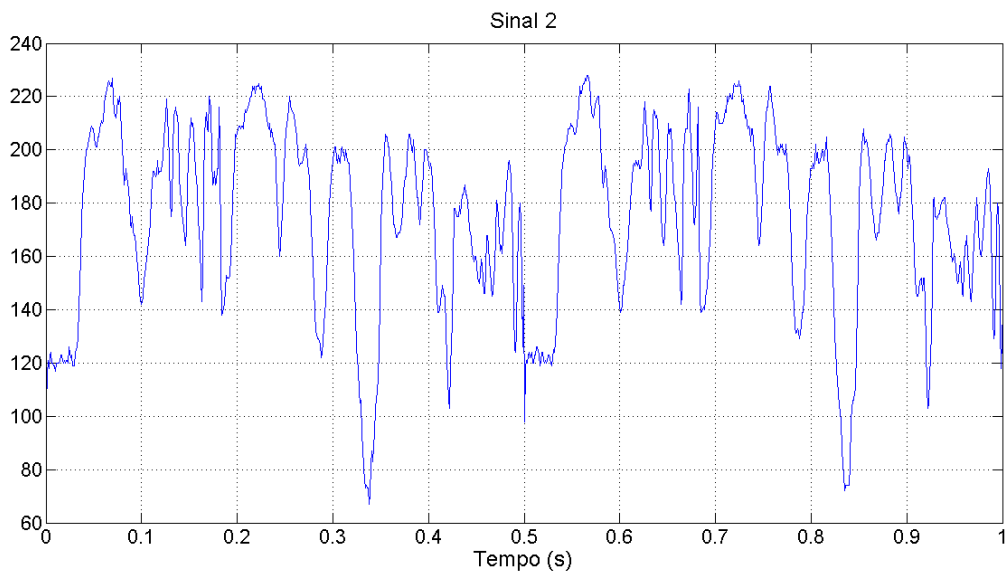
3.3.3 Escolha das bases wavelet

Seguindo o algoritmo de conversão, só foram utilizadas bases wavelet com 1 ou 2 *vanishing moments*; além disso, foram escolhidas wavelets das mesmas famílias. Foram escolhidas todas as bases disponíveis no *Wavelet Toolbox* que obedecessem a esses critérios; elas são mostradas na Tabela 3.1. As bases utilizadas no trabalho original [1] foram as gaussianas *gaus1* e *gaus2*, que também foram aplicadas nas simulações deste trabalho e, assim, podem funcionar como parâmetro de comparação dos resultados.

Tabela 3.1: Bases wavelet utilizadas

Família	1 <i>vanishing moment</i>	2 <i>vanishing moments</i>
Gaussianas	<i>gaus1</i>	<i>gaus2</i> <i>mexh</i>
Daubechies	<i>db1</i>	<i>db2</i>
Biortogonais	<i>bior1.1</i>	<i>bior2.2</i>
Biortogonais Reversas	<i>rbio1.1</i>	<i>rbio2.2</i>

As bases gaussianas são formadas pela p -ésima derivada da função gaussiana, descrita pela Eq. 3.3. A família das gaussianas é representada pelas bases *gausN*, em que N é o número de *vanishing moments* da base. As bases *gaus1* e *gaus2* correspondem, respectivamente, às primeira e segunda derivadas de uma gaussiana. A base *mexh* (*mexican hat*—chapéu mexicano) é proporcional à segunda derivada da gaussiana, a base *gaus2*, e portanto tem dois *vanishing moments*. As três bases utilizadas são mostradas na Figura 3.10.

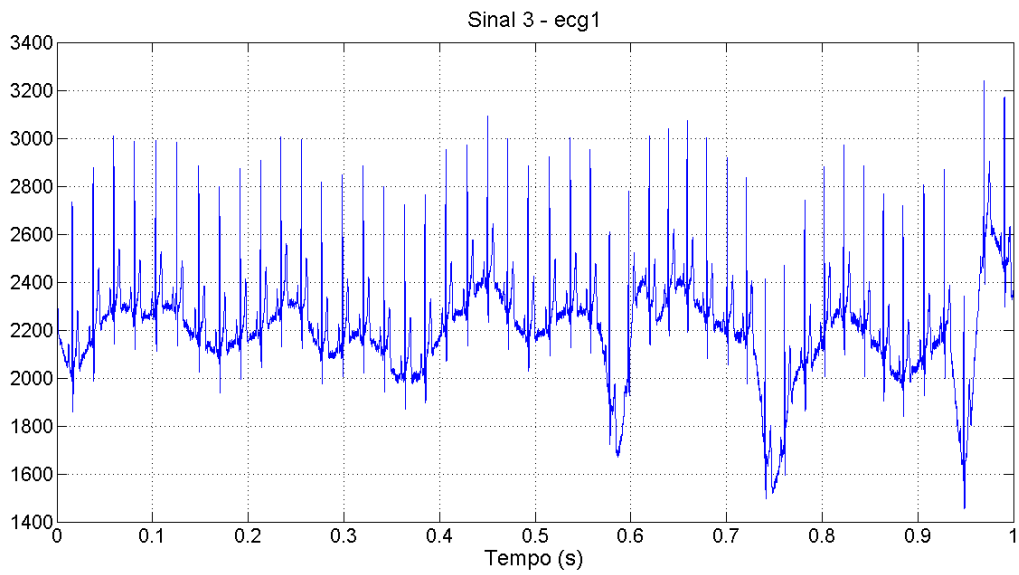


(a)

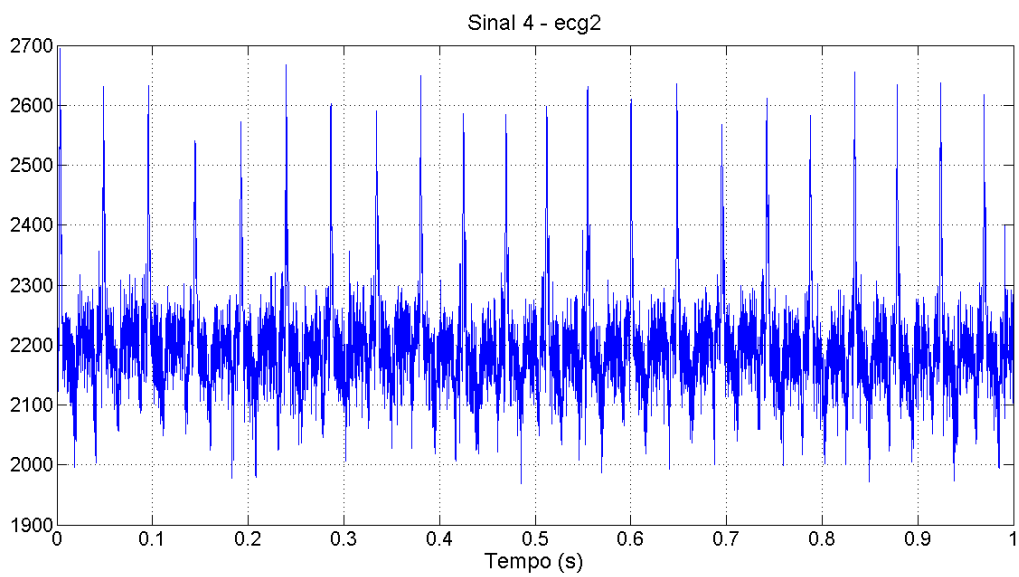


(b)

Figura 3.8: (a) Sinal 2, sinal arbitrário formado por informações de (b).



(a)



(b)

Figura 3.9: Sinais de eletrocardiograma [7]. (a) Sinal 3 (ecg1). (b) Sinal 4 (ecg2).

$$f(x) = C_p e^{-x^2} \quad (3.3)$$

$$\|f^{(p)}\|^2 = 1 \quad (3.4)$$

As bases de Daubechies foram descritas por Ingrid Daubechies [11]. Como são bases ortonormais de suporte compacto, sua forma genérica pode ser expressa pela Eq. 3.5 [12], em que ψ é a wavelet, ϕ é a função escala correspondente (*scaling function*), e h_n é a expressão tanto do filtro de reconstrução quanto do filtro de decomposição associado à base. Essas bases são representadas como dbN , em que N é o número de *vanishing moments* e pode ser qualquer número inteiro positivo.

Já as bases biortogonais têm associados a elas dois filtros não necessariamente iguais: um de reconstrução, expresso por h_n , e um de decomposição, expresso por g_n , apresentando tanto para sua função wavelet ψ quanto para sua função escala ϕ uma forma de síntese e uma de análise. No caso deste trabalho, é aplicada a forma de análise da função wavelet. As bases biortogonais utilizadas são bases de *splines* biortogonais, cuja forma genérica é indicada na Eq. 3.7 [12, 13]. Também é utilizada a base biortogonal reversa, derivada da biortogonal com a inversão das funções de análise e síntese. As bases biortogonais são referidas como $biorNr.Nd$, e as biortogonais reversas, $rbioNr.Nd$, sendo Nr e Nd o número de *vanishing moments* das funções de síntese e de análise, respectivamente.

Essas bases são mostradas na Figura 3.11. As bases de 1 *vanishing moment* das wavelets de Daubechies e das biortogonais são iguais, sendo equivalentes à wavelet de Haar, mostrada na Figura 3.11(a). As restantes são as bases de 2 *vanishing moments* para as três famílias: em (b), é mostrada a $db2$; em (c), a $bior2.2$; e, em (d), a $rbio2.2$.

$$\psi(x) = \sqrt{2} \sum_n (-1)^n h_{-n+1} \phi(2x - n) \quad (3.5)$$

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n) \quad (3.6)$$

$$\psi(x) = \sqrt{2} \sum_n g_n \phi(2x - n) \quad (3.7)$$

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n) \quad (3.8)$$

3.3.4 Quantização

Para avaliar o efeito da quantização no processo de conversão A/D, a reconstrução do sinal foi refeita quantizando as amplitudes amostradas nos pontos críticos do sinal de entrada do ADC e

também os valores estimados para o coeficiente de Lipschitz. Primeiro, as simulações foram feitas quantizando tanto as amplitudes quanto o coeficiente de Lipschitz e, em seguida, foram verificados os resultados quantizando esses valores separadamente.

As resoluções escolhidas foram de 4, 8 e 12 bits, explorando casos de baixa, média e alta resolução.

3.3.5 Métricas de erro

Na comparação entre o sinal reconstruído e o sinal de entrada no ADC, foram utilizadas duas métricas de erro: o erro relativo e o erro RMS. O erro relativo é um vetor cujos elementos consistem da diferença entre os dois sinais avaliados com relação ao sinal original, permitindo avaliar a semelhança entre os sinais ponto por ponto. Este erro pode ser expresso em porcentagem, pela fórmula da Eq. 3.9, sendo y o sinal original e \hat{y} o sinal reconstruído.

$$\text{erro relativo (\%)} = 100 \frac{|y - \hat{y}|}{y} \quad (3.9)$$

O erro RMS é o erro quadrático médio, que é expresso pela Eq. 3.10, em que l é o comprimento dos sinais y e \hat{y} . O erro RMS é representado por um escalar, permitindo analisar a semelhança entre os sinais comparados de forma geral.

$$\text{erro RMS} = \sqrt{\frac{\sum_{k=1}^l (\hat{y}_k - y_k)^2}{l}} \quad (3.10)$$

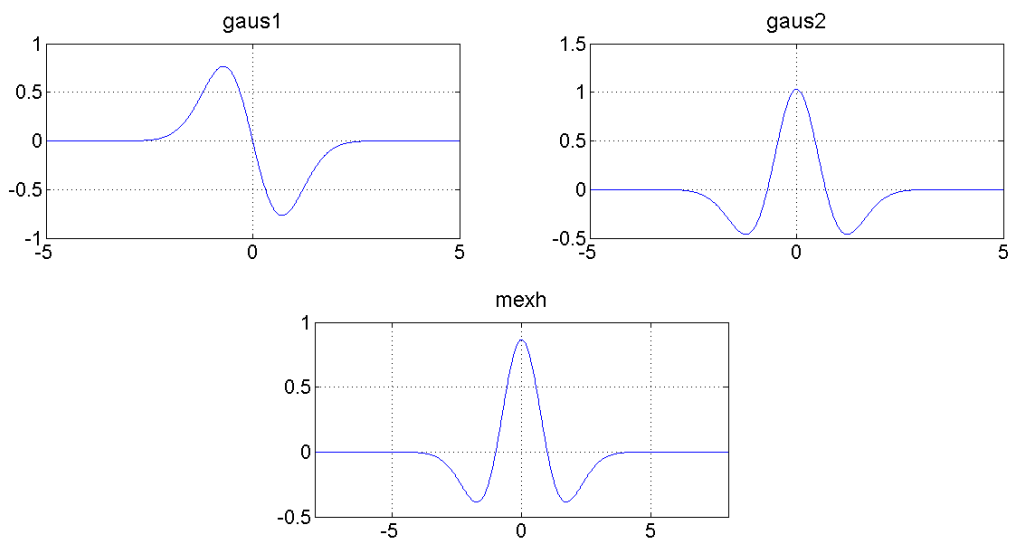


Figura 3.10: Bases gaussianas utilizadas.

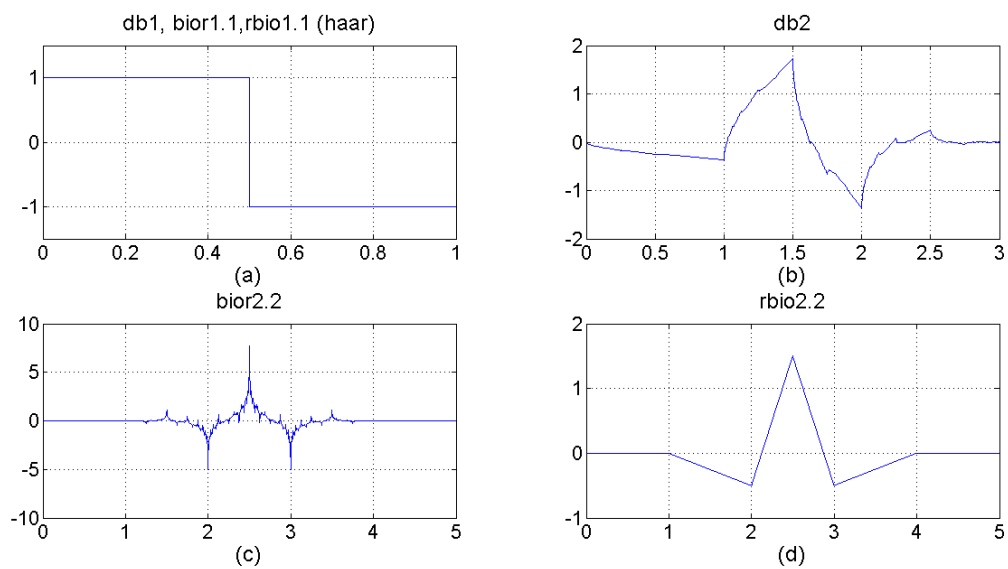


Figura 3.11: (a) Bases de Daubechies e biortogonais com 1 *vanishing moment*. Bases com 2 *vanishing moments*: (b) Daubechies (c) Biortogonal (d) Biortogonal reversa.

Capítulo 4

Resultados e Análise

Neste capítulo, são mostrados e avaliados os resultados obtidos nas simulações feitas, dispostos em tabelas. Também são mostrados gráficos dos resultados de algumas das simulações feitas em ambiente MATLAB para destacar etapas dos algoritmos aplicados e alguns resultados interessantes, além de facilitar a análise. Este capítulo está organizado em subseções de acordo com o sinal de entrada aplicado no ADC. Em cada subseção, são verificados os dados obtidos nas simulações para cada conjunto de escalas e cada par de bases wavelet.

Antes de iniciar essa análise, no entanto, é importante avaliar os dados relativos ao número de pontos identificados em cada sinal sob as diferentes condições aplicadas. Para as escalas de 16 a 64, essa informação está disponível na Tabela 4.1 e, para as escalas de 1 a 64, na Tabela 4.2. Essas tabelas também mostram o comprimento dos sinais, que precisaram ser discretizados no MATLAB.

Tabela 4.1: Número de pontos utilizados para reconstruir cada sinal de acordo com as bases aplicadas, para **escalas de 16 a 64**.

Sinal	Comprimento	Número de amostras para reconstrução				
		gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
1	1000	9	9	9	9	9
2	1024	65	65	129	129	129
3	44604	1837	1837	4165	4165	4165
4	16999	893	893	3449	3449	3449

A comparação das tabelas mostra que, ao se utilizar um conjunto de escalas que abrange escalas menores, o número de pontos identificados pode aumentar bastante, como verificado para os sinais 2, 3 e 4: o número de pontos amostrados para o segundo conjunto de escalas nesses casos apresentou aumentos de aproximadamente 3 a 6 vezes com relação aos dados para o primeiro conjunto de escalas. Essa mudança não é verificada para o sinal 1, que é mais suave e bem comportado que os restantes.

Tabela 4.2: Número de pontos utilizados para reconstruir cada sinal de acordo com as bases aplicadas, para escalas de 1 a 64.

Sinal	Comprimento	Número de amostras para reconstrução				
		gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
1	1000	9	9	9	9	9
2	1024	357	357	615	615	615
3	44604	15061	15061	18775	18775	18775
4	16999	8275	8275	8645	8645	8645

Avaliando os dados para o mesmo conjunto de escalas, nota-se que a escolha das bases também afeta o número de amostras: são identificados menos pontos com as bases gaussianas, isto é, os pares gaus1,gaus2 e gaus1,mexh, para os sinais 2, 3 e 4. Por outro lado, a diferença entre o número de amostras para as bases gaussianas e as outras bases não segue um comportamento uniforme. Na Tabela 4.1, nota-se que, com os sinais 2 e 3, o número de pontos para as bases gaussianas são próximos da metade do número de pontos para os outros três pares de bases, e com o sinal 4, o número de amostras para as bases gaussianas passa a ser cerca de quatro vezes menor que o número para as bases restantes. Já na Tabela 4.2, para o sinal 2, esse número para as gaussianas é quase metade do número para as outras bases, enquanto esses valores para os sinais 3 e 4 são próximos.

Essas comparações podem influenciar a escolha das bases e escalas a serem utilizadas na transformada wavelet, juntamente com a informação do erro de reconstrução para essas configurações. Os resultados das reconstruções e seus respectivos erros serão mostrados nas quatro seções seguintes. Com os dados de todos os resultados expostos, será feita, na última seção, uma análise geral das informações obtidas.

4.1 Resultados das análises para o sinal 1

A primeira simulação com o sinal 1 foi feita com as bases gaus1 e gaus2 para as escalas de 16 a 64. Essas são as mesmas condições aplicadas nas avaliações feitas no primeiro projeto [1], o que permite comparar os resultados obtidos na saída do ADC. A Figura 4.1 ilustra o processo de identificação de pontos de máximos e mínimos locais do sinal de entrada, que é mostrado nas Figuras 4.1(a) e 4.2(a). Primeiro, na Figura 4.1 estão os sinais que detectam esses pontos: em (b) estão a linha de coeficientes da transformada wavelet para a menor escala e também a saída do comparador, uma onda quadrada cujas bordas de subida e descida indicam os pontos de interesse; (c) e (d) mostram as saídas do ADC relacionadas a esses pontos: respectivamente, suas amplitudes e as estimativas para o coeficiente de Lipschitz, além de suas posições. Já a Figura 4.2 ilustra o processo de identificação de pontos de inflexão: os sinais de detecção obtidos com os dados da transformada wavelet são mostrados em (b) e, em (c), estão as amplitudes e posições obtidas para

os pontos de inflexão detectados. Pela Figura 4.2(b) nota-se que, neste caso, são identificados três pontos de inflexão entre o ponto de singularidade e o último ponto do sinal 1. Portanto, o ponto de inflexão nesse trecho tem sua posição aproximada pelo ponto médio dos três pontos originalmente identificados. Esse processo é percebido na Figura 4.2(c), que mostra as amplitudes dos pontos identificados antes e depois dessa aproximação.

A comparação das Figuras 4.1 e 4.2 com a Figura 3.1 e o procedimento descrito no Capítulo 3 evidenciam as mudanças aplicadas ao algoritmo original: a eliminação de valores de limiar para gerar os sinais de detecção de pontos e a amostragem da amplitude também nos pontos de inflexão.

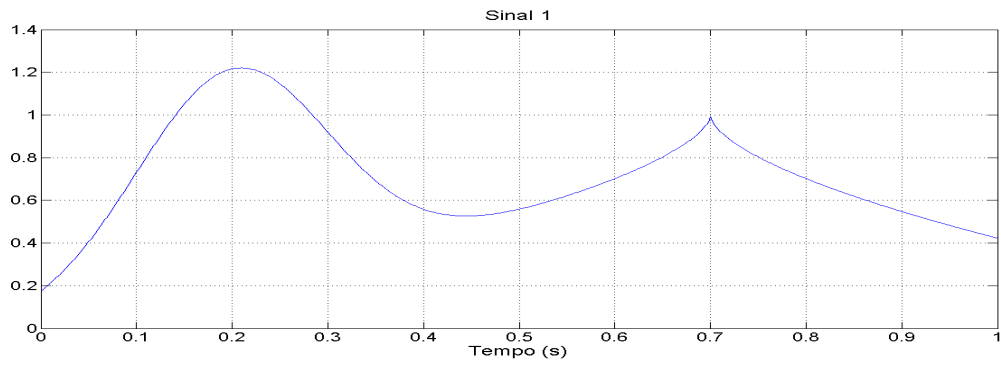
Com os dados dos gráficos das Figuras 4.1 e 4.2, é possível montar os polinômios de aproximação do sinal, seguindo a Eq. 3.1. Os *scripts* em MATLAB, disponíveis no Anexo I, utilizam cinco vetores formados a partir dos dados de saída do ADC para montar cada polinômio:

1. Vetor **tempos**: indica os instantes no tempo em que ocorrem os pontos identificados na amostragem, isto é, máximos, mínimos, singularidades e pontos de inflexão. Neste caso, são os instantes identificados nas Figuras 4.1(b) e 4.2(b). Esta informação permite identificar os instantes final e inicial de cada polinômio de reconstrução, isto é, t_0 e t_f , além do suporte de cada trecho, μ ;
2. Vetor **tau**: vetor com os instantes em que são identificados apenas pontos de máximos, mínimos ou singularidades, correspondentes a τ na Eq. 3.1 e mostrados na Figura 4.1(b);
3. Vetor **A_s**: mostra as amplitudes amostradas nos pontos de máximo/mínimo/singularidade identificados, A_s , como mostrado na Figura 4.1(c);
4. Vetor **A_i**: mostra as amplitudes amostradas nos pontos de inflexão, A_i , como mostrado na Figura 4.2(c);
5. Vetor **am_lpz**: vetor com os coeficientes de Lipschitz estimados nos pontos de máximo/mínimo/singularidade, α_s , como mostrado na Figura 4.2(d).

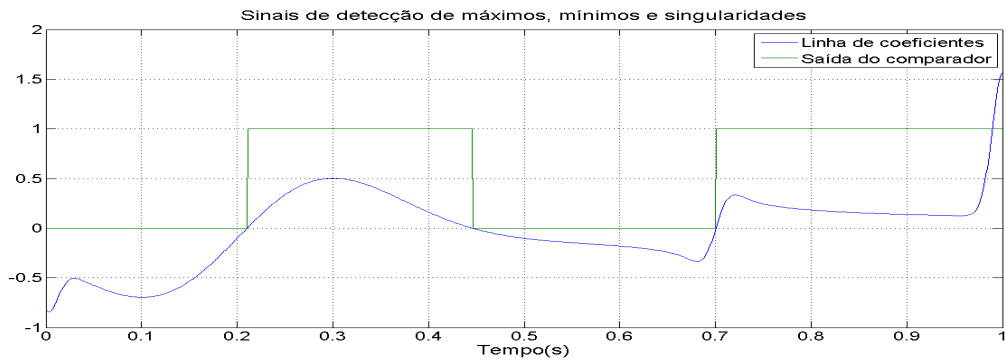
Sem quantizar as amplitudes amostradas, foram observados os seguintes valores para os vetores citados:

$$\begin{aligned} \mathbf{tempos} &= [0 \quad 0.0831 \quad 0.2112 \quad 0.3093 \quad 0.4464 \quad 0.6236 \quad 0.7007 \quad 0.8829 \quad 1] \\ \mathbf{tau} &= [0 \quad 0.2112 \quad 0.4464 \quad 0.7007 \quad 1] \\ \mathbf{A_s} &= [0.1759 \quad 1.2203 \quad 0.5263 \quad 0.9848 \quad 0.4229] \\ \mathbf{A_i} &= [0.6155 \quad 0.8740 \quad 0.7462 \quad 0.5712] \\ \mathbf{am_lpz} &= [1.2464 \quad 1.7658 \quad 2.0006 \quad 0.6013 \quad 1.0232] \end{aligned}$$

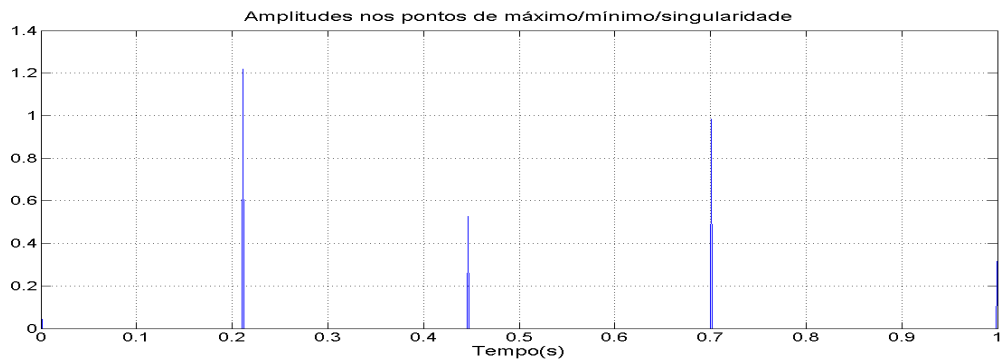
Como são identificados 9 pontos, serão necessários oito polinômios de reconstrução. Utilizando os valores obtidos imediatamente após a amostragem, isto é, sem passá-los pelo quantizador, as expressões de cada um desses polinômios são:



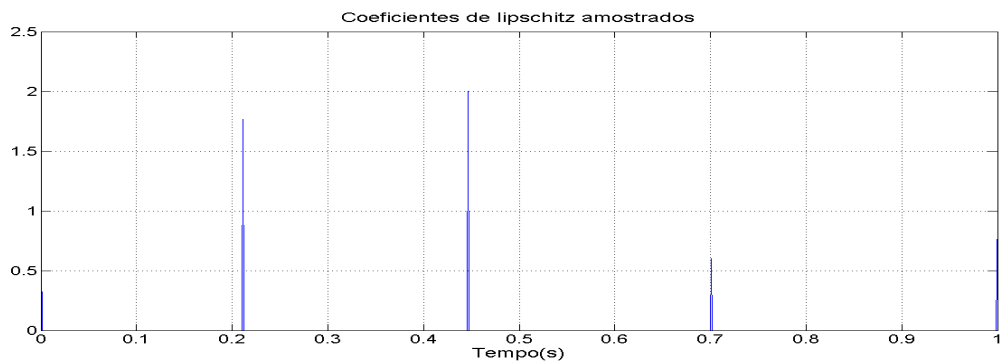
(a) Sinal de entrada



(b) Sinais de detecção de pontos de máximos, mínimos e singularidades

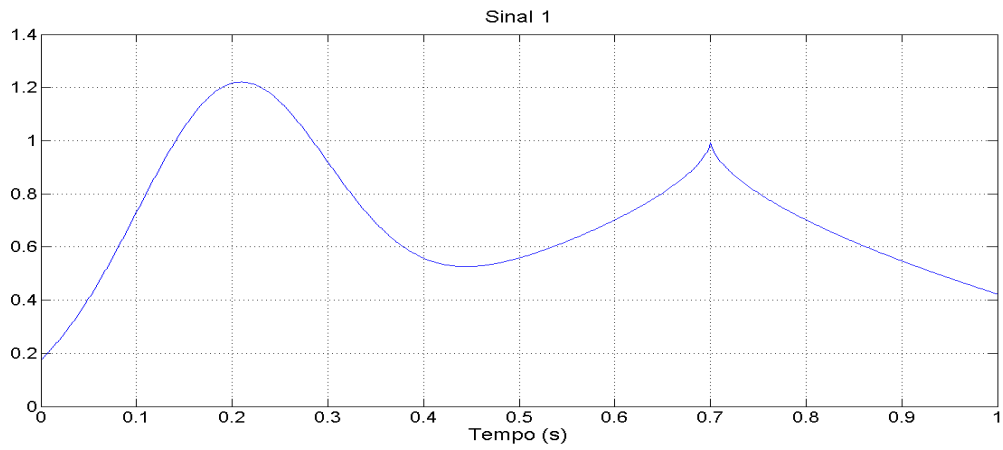


(c) Amplitudes amostradas nos máximos e mínimos locais identificados

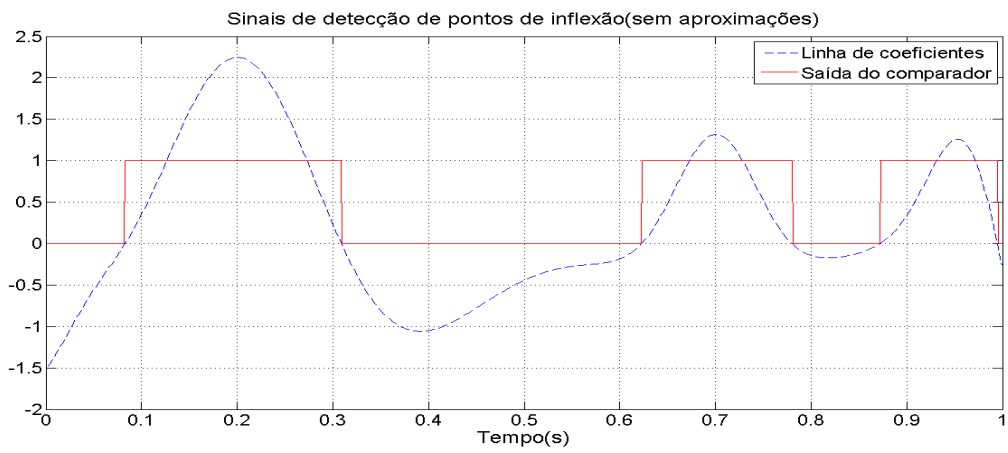


(d) Valores estimados para o coeficiente de Lipschitz.

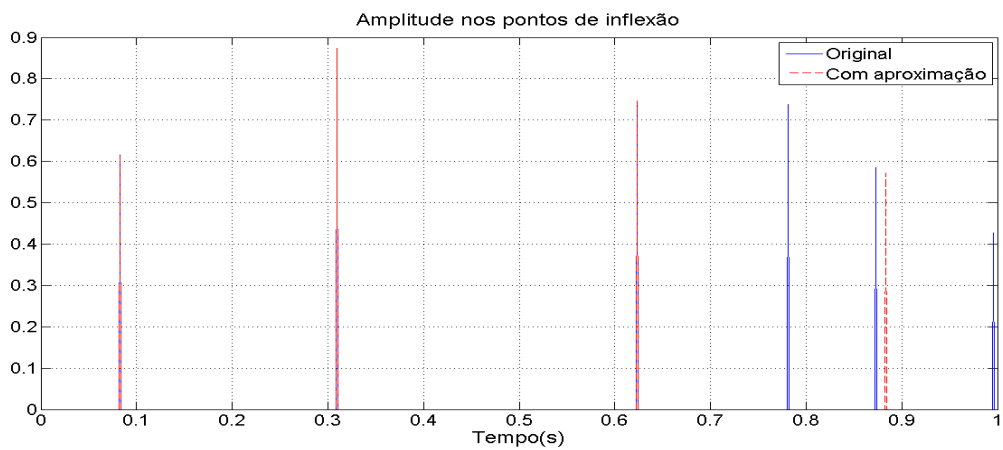
Figura 4.1: Identificação de pontos máximos, mínimos e singularidades pelo ADC



(a) Sinal de entrada



(b) Sinais de detecção de pontos de inflexão



(c) Amplitudes amostradas nos pontos de inflexão antes e depois das aproximações de posição

Figura 4.2: Identificação de pontos de inflexão pelo ADC

$$P_1(t) = 0.6155 + (0.1759 - 0.6155) \left(1 - \left| \frac{0-t}{0.0831} \right|^{1.2464} \right) , \quad 0s \leq t \leq 0.0831s$$

$$P_2(t) = 0.6155 + (1.2203 - 0.6155) \left(1 - \left| \frac{0.2112-t}{0.1281} \right|^{1.7658} \right) , \quad 0.0831s \leq t \leq 0.2112s$$

$$P_3(t) = 0.8740 + (1.2203 - 0.8740) \left(1 - \left| \frac{0.2112-t}{0.0981} \right|^{1.7658} \right) , \quad 0.2112s \leq t \leq 0.3093s$$

$$P_4(t) = 0.8740 + (0.5263 - 0.8740) \left(1 - \left| \frac{0.4464-t}{0.1371} \right|^{2.0006} \right) , \quad 0.3093s \leq t \leq 0.4464s$$

$$P_5(t) = 0.7462 + (0.5263 - 0.7462) \left(1 - \left| \frac{0.4464-t}{0.1772} \right|^{2.0006} \right) , \quad 0.4464s \leq t \leq 0.6236s$$

$$P_6(t) = 0.7462 + (0.9848 - 0.7462) \left(1 - \left| \frac{0.7007-t}{0.0771} \right|^{0.6013} \right) , \quad 0.6236s \leq t \leq 0.7007s$$

$$P_7(t) = 0.5712 + (0.9848 - 0.5712) \left(1 - \left| \frac{0.7007-t}{0.1822} \right|^{0.6013} \right) , \quad 0.7007s \leq t \leq 0.8829s$$

$$P_8(t) = 0.5712 + (0.4229 - 0.5712) \left(1 - \left| \frac{1-t}{0.1171} \right|^{1.0232} \right) , \quad 0.8829s \leq t \leq 1s$$

Após a quantização, os vetores de amplitudes (A_s e A_i) e dos coeficientes de Lipschitz (am_{lpz}) são modificados. Os novos vetores são nomeados acrescentando à direita a resolução em bits. Os vetores com os valores das amplitudes amostradas e dos coeficientes de Lipschitz quantizados para as diferentes resoluções são os seguintes:

$$A_{s4} = [0.1525 \quad 1.1440 \quad 0.4576 \quad 0.9152 \quad 0.3813]$$

$$A_{s8} = [0.1716 \quad 1.2155 \quad 0.5243 \quad 0.9819 \quad 0.4195]$$

$$A_{s12} = [0.1758 \quad 1.2200 \quad 0.5261 \quad 0.9846 \quad 0.4227]$$

$$A_{i4} = [0.6009 \quad 0.8194 \quad 0.7102 \quad 0.5463]$$

$$A_{i8} = [0.6146 \quad 0.8706 \quad 0.7443 \quad 0.5702]$$

$$A_{i12} = [0.6154 \quad 0.8738 \quad 0.7460 \quad 0.5710]$$

$$am_{lpz4} = [1.1253 \quad 1.7505 \quad 1.8756 \quad 0.5002 \quad 1.0003]$$

$$am_{lpz8} = [1.2426 \quad 1.7583 \quad 1.9928 \quad 0.5939 \quad 1.0159]$$

$$am_{lpz12} = [1.2460 \quad 1.7657 \quad 2.0001 \quad 0.6008 \quad 1.0228]$$

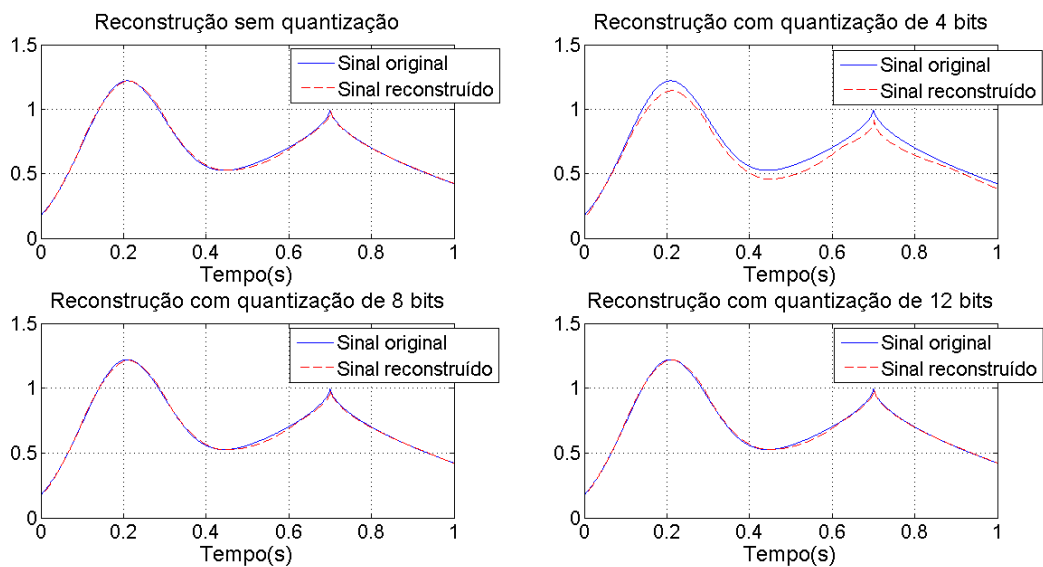
Com esses valores, ficam definidos os polinômios para as amplitudes e os coeficientes de Lipschitz quantizados e também para os outros dois casos, em que o efeito da quantização sobre cada um desses parâmetros é avaliado separadamente.

A Figura 4.3(a) mostra os resultados da reconstrução a partir dos polinômios indicados, tanto para o caso sem quantização (resolução altíssima) quanto para a quantização com resoluções de 4, 8 e 12 bits. O erro relativo de reconstrução para cada um desses casos é mostrado, em porcentagem, na Figura 4.3(b). Os erros RMS também foram calculados, e são mostrados na Tabela 4.3. Pelos resultados, nota-se que, para as bases e escalas escolhidas, o sinal pôde ser reconstruído com erro relativo máximo de aproximadamente 3.5%, e erro RMS igual a 0.0095. Quando a quantização é levada em conta no processo, o pior resultado, como esperado, ocorre para o caso de baixa resolução (4 bits), quando o erro relativo máximo aumenta para cerca de 13.5%, e o erro RMS passa a ser 0.0565. Os outros dois casos de quantização apresentam resultados próximos ao observado sem quantização: para média resolução (8 bits), o erro relativo máximo fica em torno de 4.7%, com erro RMS igual a 0.0101; para alta resolução (12 bits), o erro relativo máximo é de pouco mais de 3.5%, e o erro RMS é calculado como 0.0096.

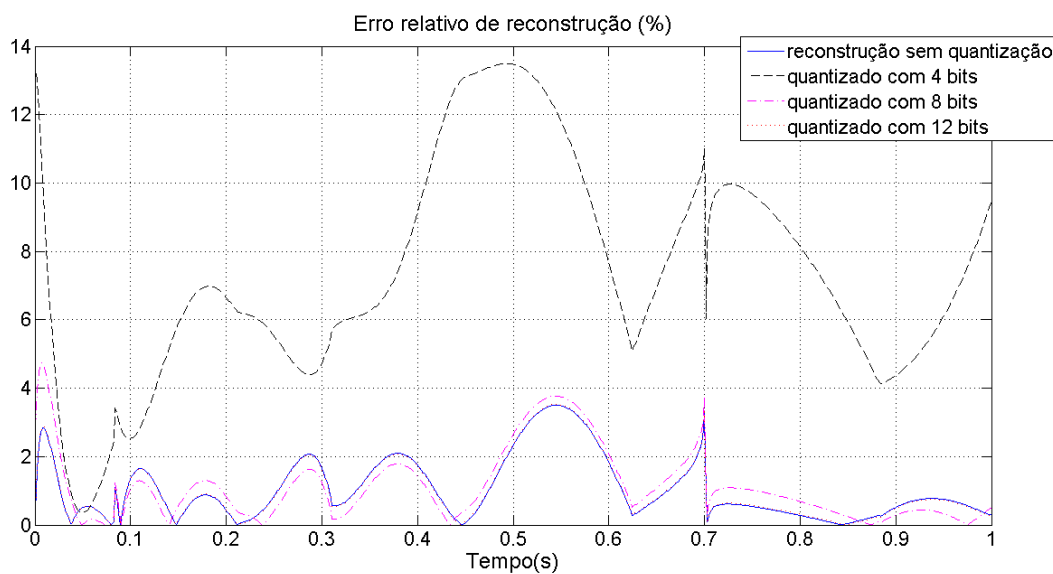
Tabela 4.3: Erro RMS de reconstrução para o **signal 1**, para **escalas de 16 a 64**.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	0.0095	0.0102	0.0450	0.0285	0.0213
12 bits	0.0096	0.0103	0.0449	0.0285	0.0213
8 bits	0.0101	0.0120	0.0441	0.0275	0.0222
4 bits	0.0565	0.0632	0.0629	0.0546	0.0651

Os testes foram repetidos para verificar o efeito da quantização sobre as amplitudes amostradas nos pontos críticos e sobre o coeficiente de Lipschitz estimado em cada ponto. Os erros RMS obtidos quando apenas as amplitudes são quantizadas estão dispostos na Tabela 4.4, para escalas de 16 a 64, e na Tabela 4.7, para escalas de 1 a 64. Os resultados da reconstrução para esse caso, com bases gaus1 e gaus2 e o primeiro conjunto de escalas, são mostrados na Figura 4.4. Já as Tabelas 4.5 e 4.8 contêm os erros RMS de reconstrução quando os coeficientes de Lipschitz são quantizados, para



(a) Resultado da reconstrução.



(b) Erros de reconstrução.

Figura 4.3: Sinal 1: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

o primeiro e o segundo conjunto de escalas, e a Figura 4.5 mostra os resultados da reconstrução para esse caso.

Comparando os resultados das Tabelas 4.4 e 4.5, nota-se que o erro RMS é menor quando apenas os coeficientes de Lipschitz passam pelo quantizador, indicando que o parâmetro que mais afeta o erro de reconstrução são as amplitudes amostradas nos pontos críticos do sinal. Nota-se que a amplitude exige maior resolução que o coeficiente de Lipschitz, já que os resultados nesse caso para resoluções média, alta e altíssima são muito próximos. Outra evidência disso surge da comparação das Figuras 4.3, 4.4: o erro relativo observado nessas duas figuras é muito próximo, indicando que a quantização do coeficiente de Lipschitz não afeta tanto o erro de reconstrução. Por outro lado, a Figura 4.5(b) apresenta comportamento diferente do das duas primeiras: a curva correspondente ao caso de baixa resolução está mais próxima das outras curvas, e o erro máximo é menor e se mantém abaixo de 5%. De fato, a distinção entre os quatro gráficos da Figura 4.5(a) é sutil, ao contrário do que ocorre nos outros dois casos.

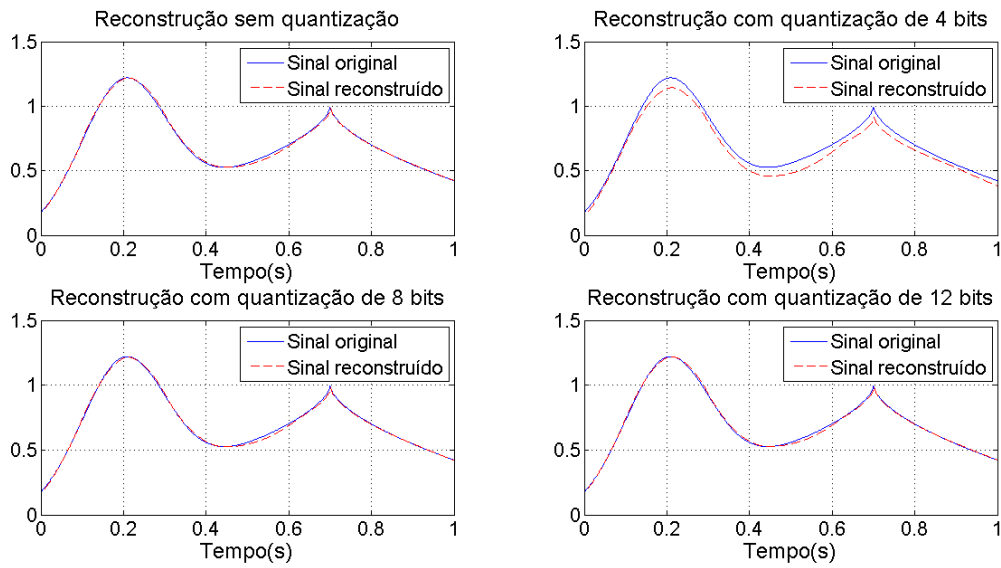
Tabela 4.4: Erro RMS de reconstrução para o **sinal 1**, para **escalas de 16 a 64**, quando apenas as amplitudes são quantizadas.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	0.0095	0.0102	0.0450	0.0286	0.0213
12 bits	0.0096	0.0103	0.0449	0.0286	0.0213
8 bits	0.0099	0.0116	0.0440	0.0279	0.0223
4 bits	0.0543	0.0554	0.0621	0.0599	0.0626

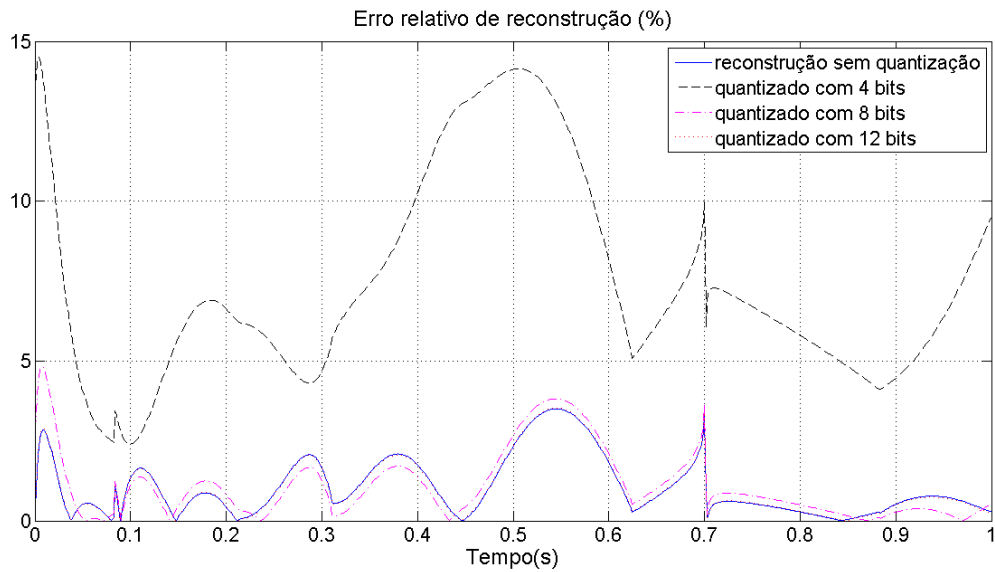
Tabela 4.5: Erro RMS de reconstrução para o **sinal 1**, para **escalas de 16 a 64**, quando apenas os coeficientes de Lipschitz são quantizados.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	0.0095	0.0102	0.0450	0.0286	0.0213
12 bits	0.0095	0.0102	0.0450	0.0286	0.0213
8 bits	0.0096	0.0106	0.0451	0.0284	0.0212
4 bits	0.0148	0.0226	0.0454	0.0233	0.0227

Como a transformada é uma medida da correlação da função analisada com uma base, espera-se que os melhores resultados sejam obtidos utilizando-se bases que tenham comportamento parecido com o do sinal ou o de trechos do sinal. As bases gaussianas são mais suaves, por isso identificam melhor o comportamento geral do sinal, enquanto as bases de Daubechies e as biortogonais apresentam transições bruscas, o que melhora a identificação de singularidades, descontinuidades

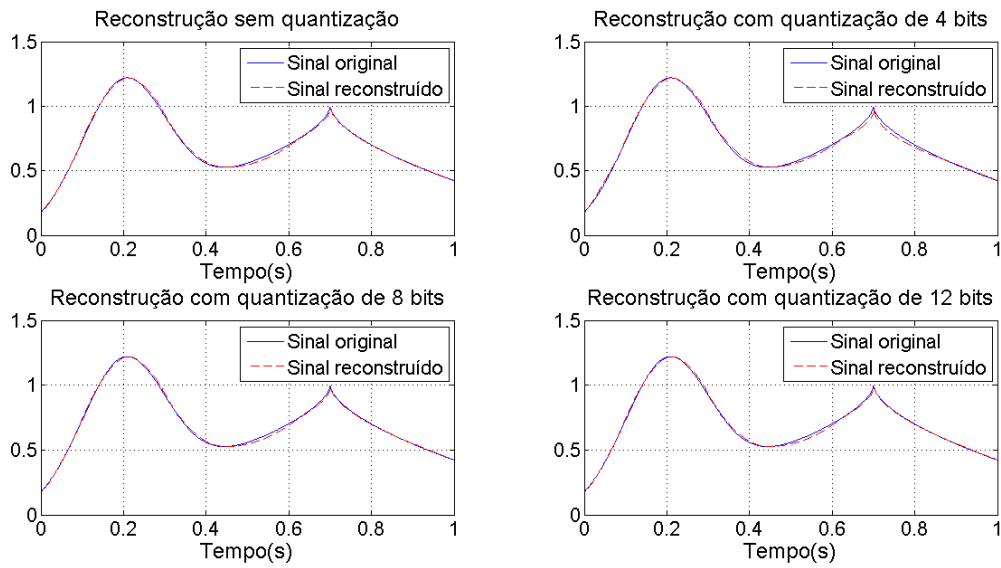


(a) Resultado da reconstrução.

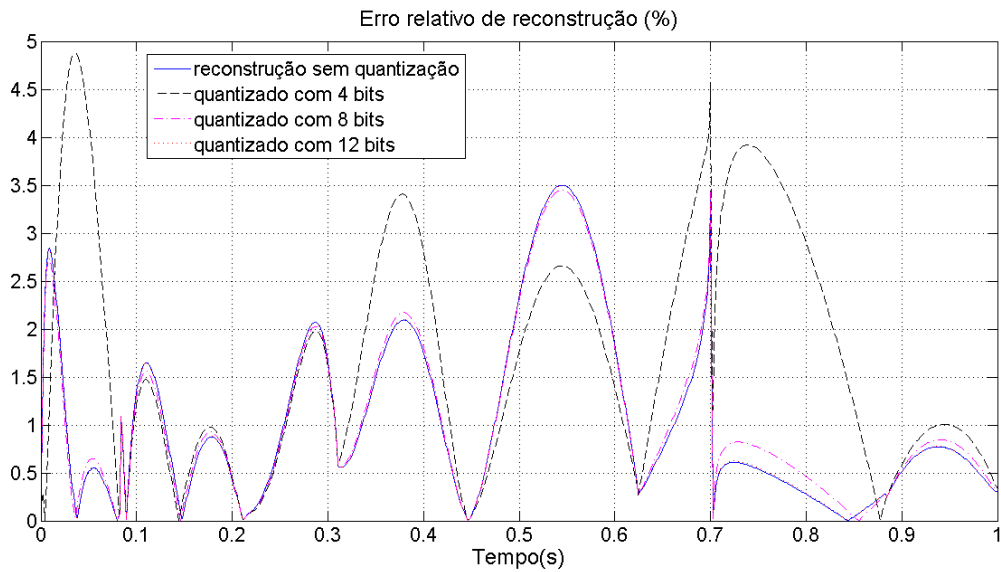


(b) Erros de reconstrução.

Figura 4.4: Sinal 1: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64. Apenas as amplitudes amostradas passaram pelo quantizador.



(a) Resultado da reconstrução.



(b) Erros de reconstrução.

Figura 4.5: Sinal 1: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64. Apenas os coeficientes de Lipschitz passaram pelo quantizador.

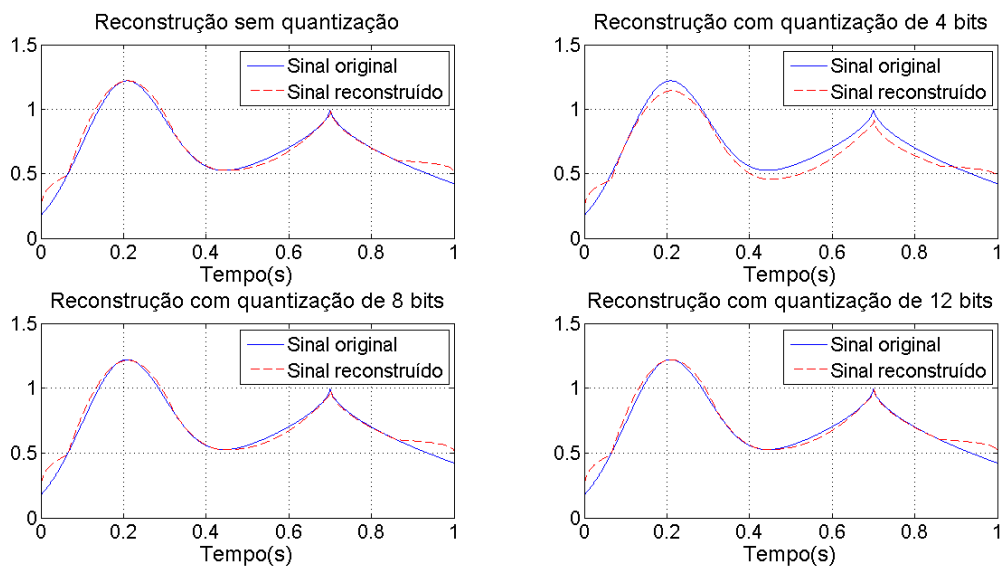
e variações rápidas do sinal avaliado. O sinal 1 é suave e constituído em parte por uma gaussiana, o que sugere que as bases gaussianas apresentem menor erro de reconstrução. Isso é confirmado pelos dados da Tabela 4.3, que indicam os maiores erros para a transformada feita com o par de bases de Daubechies, e os menores erros quando são utilizados os dois pares de bases derivadas da gaussiana. Esse resultado é mostrado na Figura 4.6, que mostra a reconstrução do sinal e o erro relativo associado para as bases db1 e db2 com escalas de 16 a 64.

Ao utilizar o segundo conjunto de escalas, a detecção de máximos e mínimos locais passa a ser feita com base nos coeficientes da transformada para escala $a = 1$ em vez de $a = 16$. Pela propriedade do *Wavelet Zoom*, conforme se diminui a escala, a transformada permite avaliar melhor os detalhes, entre eles, as posições de singularidades do sinal; portanto, utilizando-se uma escala menor, espera-se que os pontos de máximos e mínimos locais do sinal sejam identificados com maior exatidão, diminuindo o erro de reconstrução. Os erros RMS para as escalas de 1 a 64 estão na Tabela 4.6, para o caso em que as amplitudes e os coeficientes de Lipschitz são quantizados; na Tabela 4.7 para quando apenas as amplitudes passam pelo quantizador; e na Tabela 4.8 para quando somente os coeficientes de Lipschitz são quantizados. Comparando esses resultados com os das Tabelas 4.3, 4.4, e 4.5, nota-se que os erros diminuíram para as bases gaus1 e mexh, e houve pequenas variações para as bases restantes. Novamente, o maior erro de reconstrução é observado para o par de bases de Daubechies. Ainda, pelas Tabelas 4.1 e 4.2, a alteração das escalas não alterou a quantidade de pontos identificados, o que pode explicar a quase manutenção dos valores de erro RMS de reconstrução para os dois conjuntos de escalas. A Figura 4.7(a) mostra o resultado da reconstrução do sinal 1 com as bases gaus1 e mexh para as escalas de 1 a 64, que foi o caso com menor erro RMS para o sinal 1. O erro relativo é mostrado na Figura 4.7(b). Nota-se que o erro relativo apresenta valores elevados nos trechos inicial e final do sinal reconstruído, mas no restante do tempo permanece abaixo de 10% nos casos de média e alta resolução. Esse comportamento nos extremos do sinal pode ser devido à mudança dos coeficientes da transformada com a alteração da base, o que modifica o valor dos parâmetros calculados, como o expoente de Lipschitz, que define o comportamento do sinal no algoritmo de reconstrução polinomial adotado.

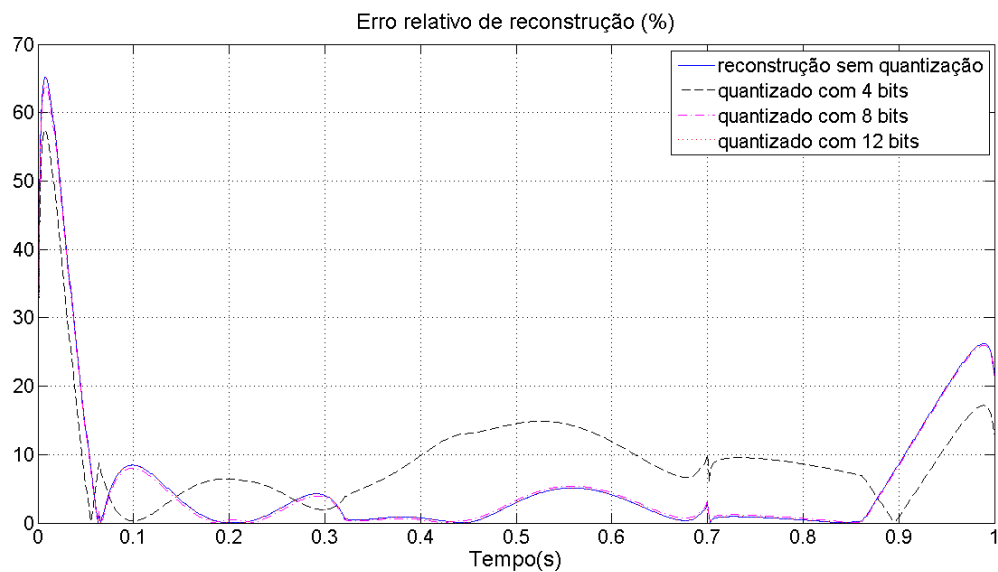
O efeito da quantização sobre as amplitudes e sobre o coeficiente de Lipschitz para esse caso pode ser observado também nos resultados das Figuras 4.8 e 4.9, respectivamente. Nota-se outra vez que a resolução das amplitudes tem maior peso sobre o resultado final que a resolução do coeficiente de Lipschitz, ou seja, que o coeficiente de Lipschitz pode gerar resultados de alta resolução com precisão menor que a dos dados de amplitude.

4.2 Resultados das análises para o sinal 2

O sinal 2 é um sinal com transições bruscas, por isso, espera-se que o erro de reconstrução para as bases de Daubechies e as biortogonais seja menor que o erro para as gaussianas. Os erros RMS para as escalas de 16 a 64 são mostrados nas Tabelas 4.9, 4.10 e 4.11. Nota-se que, de fato, as bases gaussianas são as que apresentam maior erro de reconstrução, mas esse erro não é muito superior ao observado para as outras bases. Para esse sinal, o número de amostras variou de acordo com a



(a) Resultado da reconstrução.



(b) Erros de reconstrução.

Figura 4.6: Sinal 1: reconstrução para transformadas feitas com as bases db1 e db2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

Tabela 4.6: Erro RMS de reconstrução para o **senal 1**, para **escalas de 1 a 64**.

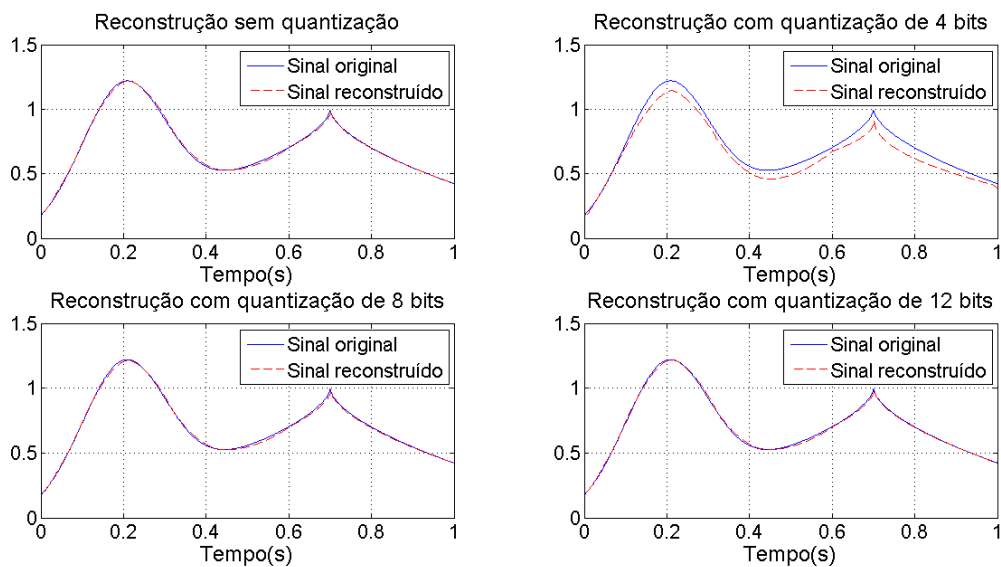
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	0.0112	0.0088	0.0493	0.0304	0.0105
12 bits	0.0111	0.0088	0.0493	0.0303	0.0105
8 bits	0.0108	0.0097	0.0483	0.0292	0.0101
4 bits	0.0557	0.0628	0.0633	0.0533	0.0598

Tabela 4.7: Erro RMS de reconstrução para o **senal 1**, para **escalas de 1 a 64**, quando apenas as amplitudes são quantizadas.

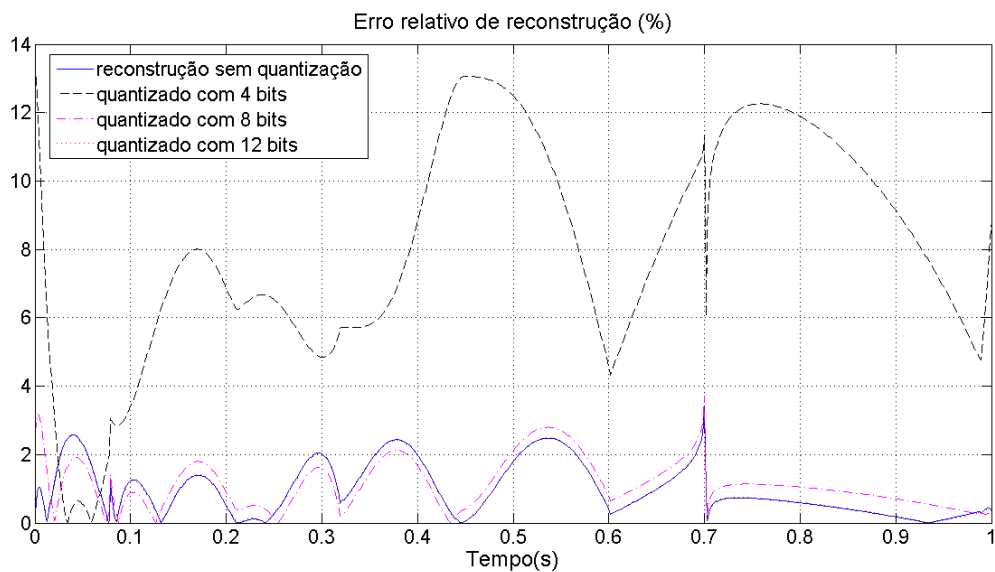
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	0.0112	0.0088	0.0493	0.0326	0.0105
12 bits	0.0111	0.0088	0.0492	0.0325	0.0105
8 bits	0.0108	0.0096	0.0483	0.0312	0.0102
4 bits	0.0524	0.0533	0.0611	0.0570	0.0564

Tabela 4.8: Erro RMS de reconstrução para o **senal 1**, para **escalas de 1 a 64**, quando apenas os coeficientes de Lipschitz são quantizados.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	0.0112	0.0088	0.0493	0.0326	0.0105
12 bits	0.0111	0.0088	0.0493	0.0326	0.0105
8 bits	0.0112	0.0089	0.0494	0.0324	0.0104
4 bits	0.0164	0.0218	0.0518	0.0294	0.0144

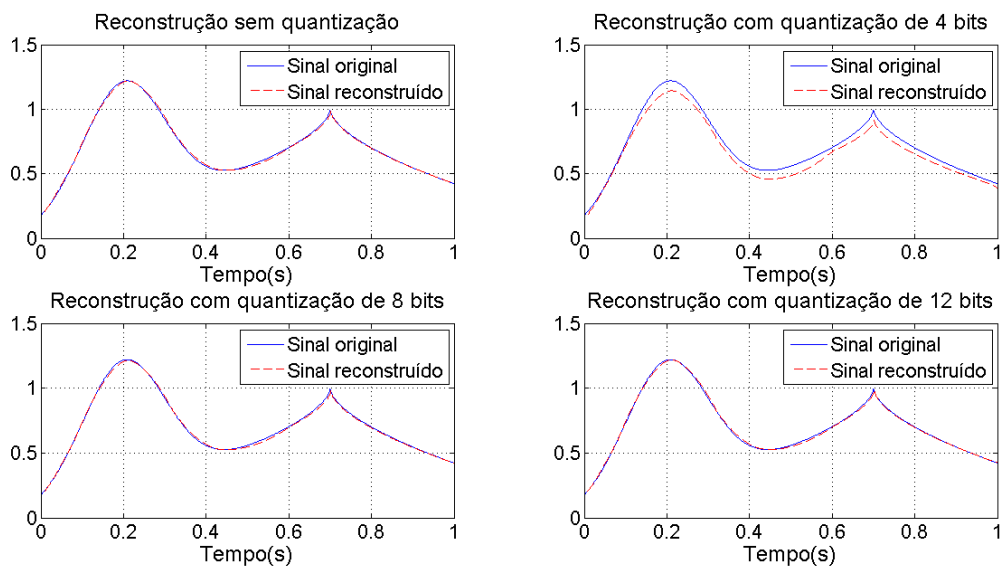


(a) Resultado da reconstrução.

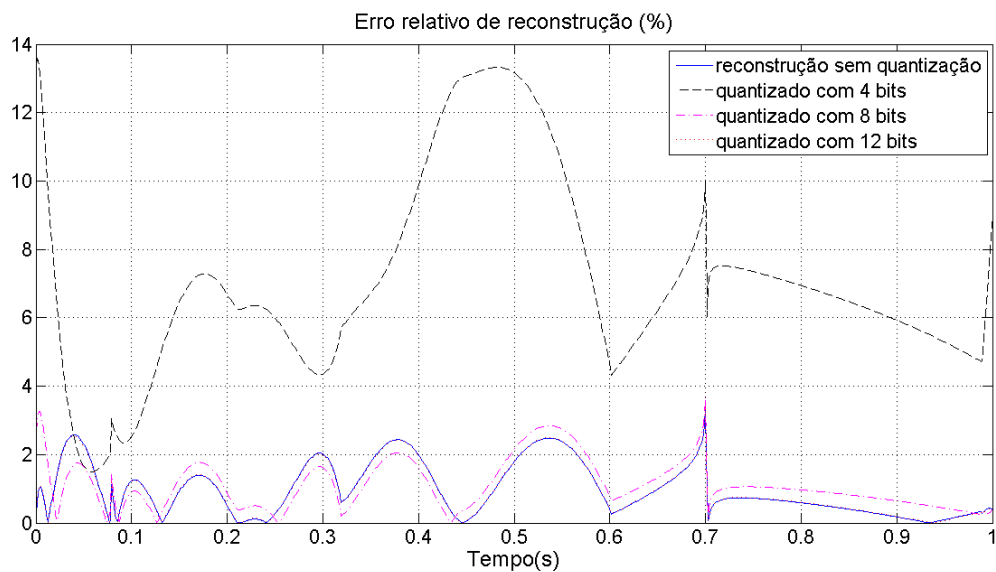


(b) Erros de reconstrução.

Figura 4.7: Sinal 1: reconstrução para transformadas feitas com as bases gauss 1 e mexh e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

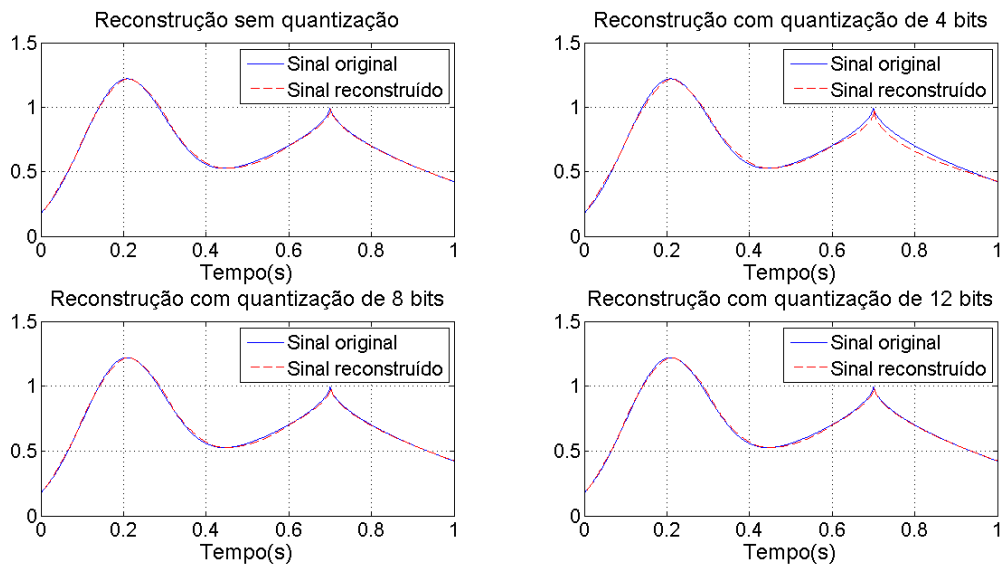


(a) Resultado da reconstrução.

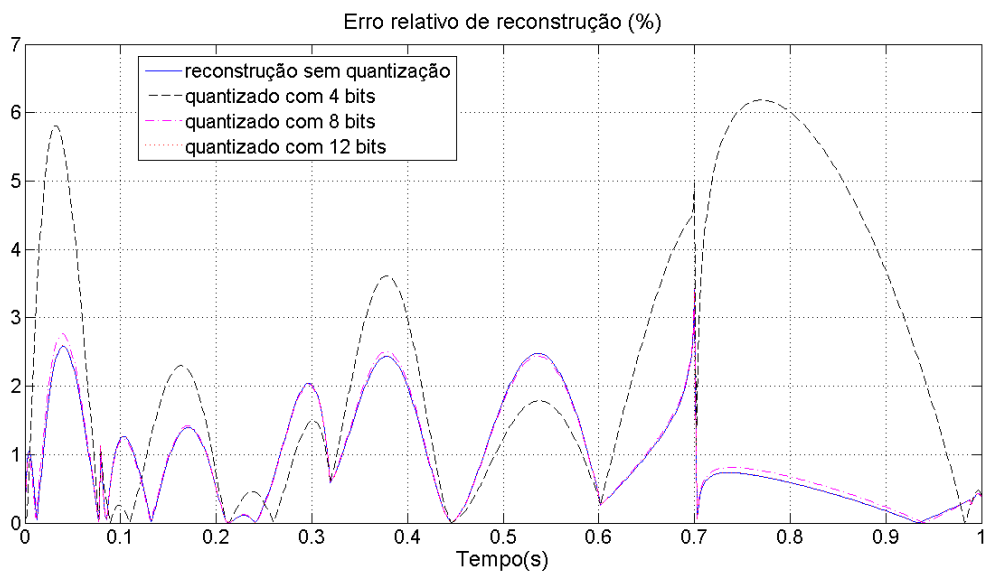


(b) Erros de reconstrução.

Figura 4.8: Sinal 1: reconstrução para transformadas feitas com as bases gauss 1 e mexh e escalas de 1 a 64, para o caso sem quantização e para quantização com 4, 8 e 12 bits. Apenas as amplitudes foram quantizadas.



(a) Resultado da reconstrução.



(b) Erros de reconstrução.

Figura 4.9: Sinal 1: reconstrução para transformadas feitas com as bases gaus 1 e mexh e escalas de 1 a 64, para o caso sem quantização e para quantização com 4, 8 e 12 bits. Apenas os coeficientes de Lipschitz passaram pelo quantizador.

base utilizada: para as gaussianas, foram necessárias 65 amostras para a reconstrução, enquanto para as outras bases foi utilizado quase o dobro desse valor, 129 amostras, como mostrado na Tabela 4.1.

Já os erros RMS para as escalas de 1 a 64 são mostrados nas Tabelas 4.12, 4.13 e 4.14. Nota-se que os erros foram reduzidos para todas as bases com relação ao resultado para o primeiro conjunto de escalas, e o par de bases com menor erro para esse sinal continuou sendo o de bases biortogonais; além disso, pela Tabela 4.2, o número de amostras utilizadas para as bases gaussianas foi de 357 amostras, e de 615 amostras para os outros pares de bases. Portanto, a mudança do conjunto de escalas contribuiu para a redução do erro de reconstrução do sinal, mas com o custo de aumentar o número de amostras utilizadas, que chegou a ser pouco mais da metade do número de pontos do sinal 2 para as bases de Daubechies e as biortogonais.

Tabela 4.9: Erro RMS de reconstrução para o **sinal 2**, para **escalas de 16 a 64**.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	22.0198	22.2514	18.7629	16.8464	19.6156
12 bits	22.0275	22.2673	18.7647	16.8517	19.6256
8 bits	22.1499	22.5042	18.8065	16.9048	19.7753
4 bits	26.1422	27.3219	21.7705	21.1148	27.3739

Tabela 4.10: Erro RMS de reconstrução para o **sinal 2**, para **escalas de 16 a 64**. Apenas as amplitudes foram quantizadas.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	22.0198	22.0810	18.5844	16.8464	17.7287
12 bits	22.0251	22.0916	18.5803	16.8426	17.7258
8 bits	22.1169	22.2370	18.5410	16.7375	17.6398
4 bits	25.3469	25.1860	19.5914	17.8079	18.2867

Para ilustrar o processo, a Figura 4.10(a) mostra o melhor resultado obtido para o primeiro conjunto de escalas, quando foram amostrados 129 pontos, e a Figura 4.10(b), o erro relativo associado. Para comparação, a Figura 4.11 mostra os resultados da reconstrução também para as bases bior1.1 e bior2.2, mas para as escalas de 1 a 64. Nesse caso, foram necessárias 615 amostras. Por fim, a Figura 4.12(a) mostra o pior resultado obtido com o segundo conjunto de escalas, que utilizou 357 amostras, e a Figura 4.12(b) mostra o erro relativo para essa reconstrução.

Mesmo sendo o resultado que apresentou menor erro RMS, a Figura 4.10(b) mostra que o erro

Tabela 4.11: Erro RMS de reconstrução para o **senal 2**, para **escalas de 16 a 64**. Apenas os coeficientes de Lipschitz foram quantizados.

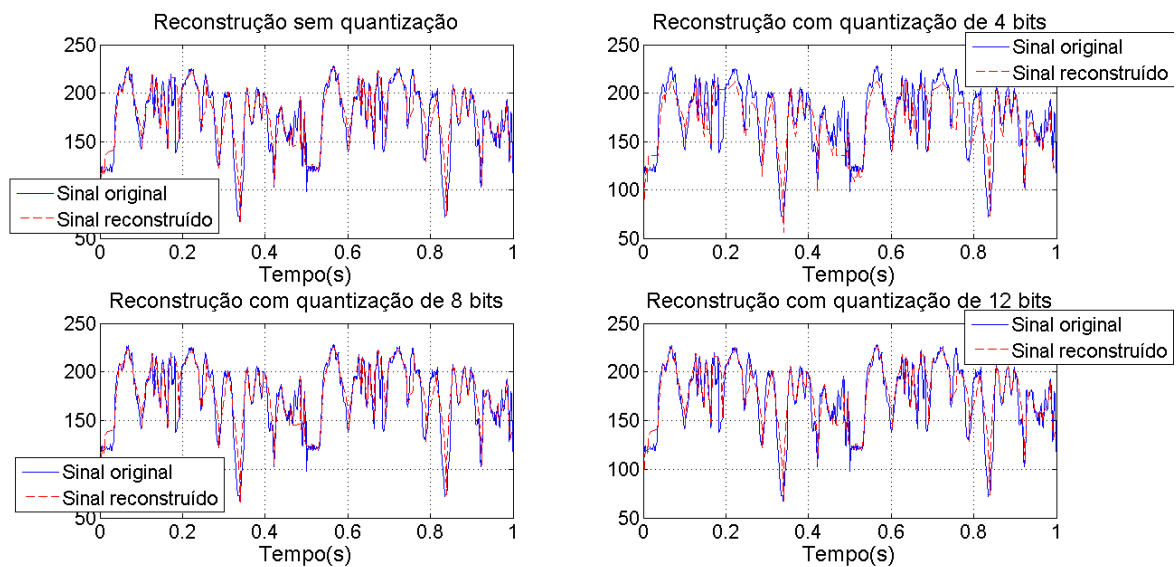
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	22.0198	22.0810	18.5844	16.8464	17.7287
12 bits	22.0222	22.0860	18.5911	16.8555	17.7399
8 bits	22.0534	22.1740	18.6802	17.0151	17.9375
4 bits	22.8572	24.2629	21.2102	20.6877	23.7835

Tabela 4.12: Erro RMS de reconstrução para o **senal 2**, para **escalas de 1 a 64**.

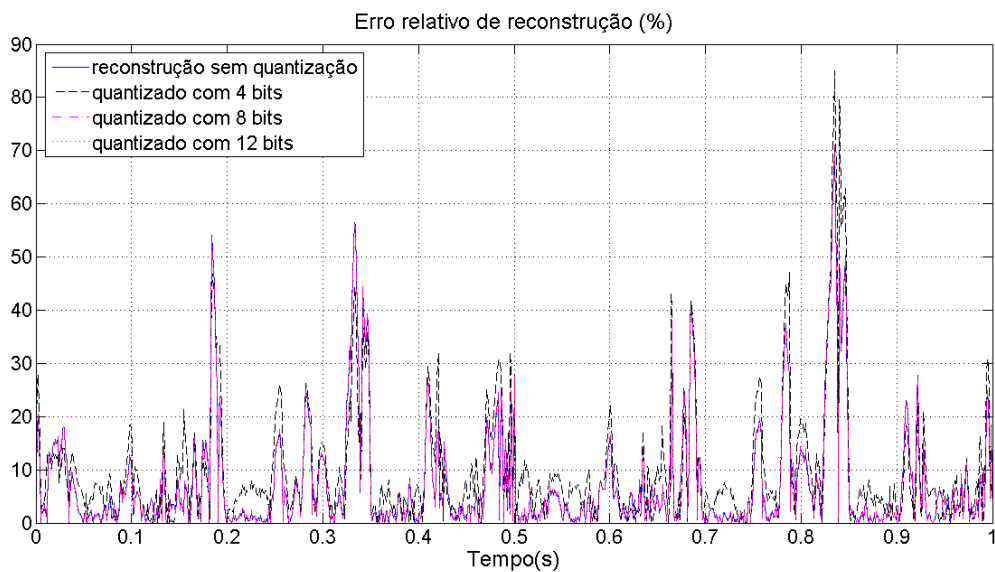
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	13.3912	12.1783	10.4635	9.5713	10.8229
12 bits	13.4021	12.1858	10.4673	9.5786	10.8304
8 bits	13.5784	12.3103	10.5786	9.7300	11.0538
4 bits	17.9713	17.1706	15.4546	14.4917	15.6156

Tabela 4.13: Erro RMS de reconstrução para o **senal 2**, para **escalas de 1 a 64**. Apenas as amplitudes foram quantizadas.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	13.3291	12.0811	10.4587	9.5713	10.5819
12 bits	13.3318	12.0821	10.4563	9.5728	10.5785
8 bits	13.3627	12.1083	10.4236	9.5856	11.5387
4 bits	16.1756	14.6068	12.6182	12.6729	12.5680

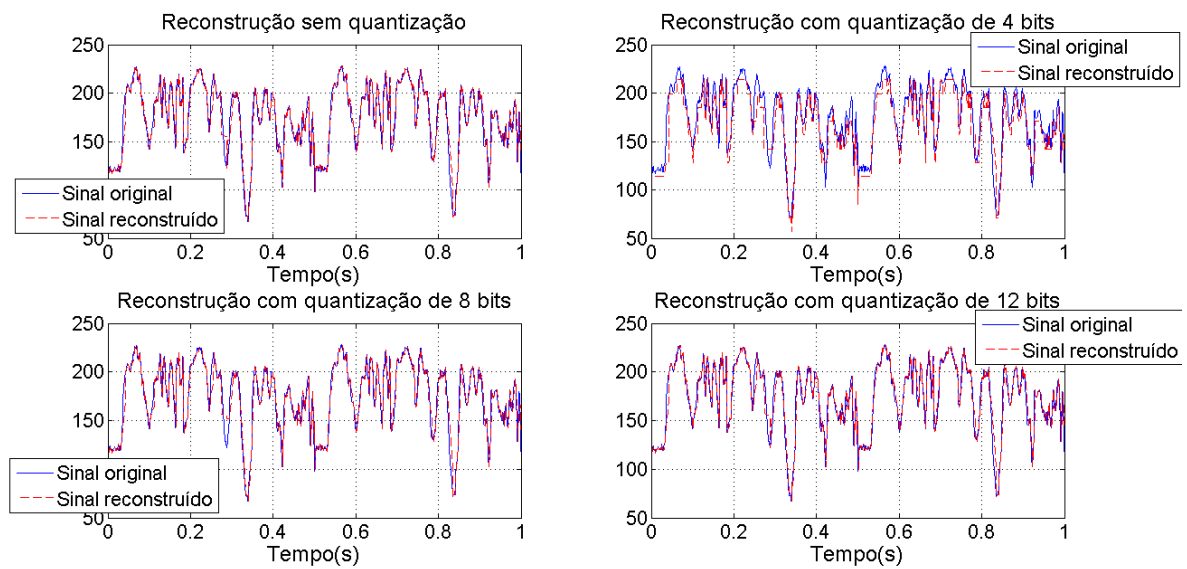


(a) Resultado da reconstrução.

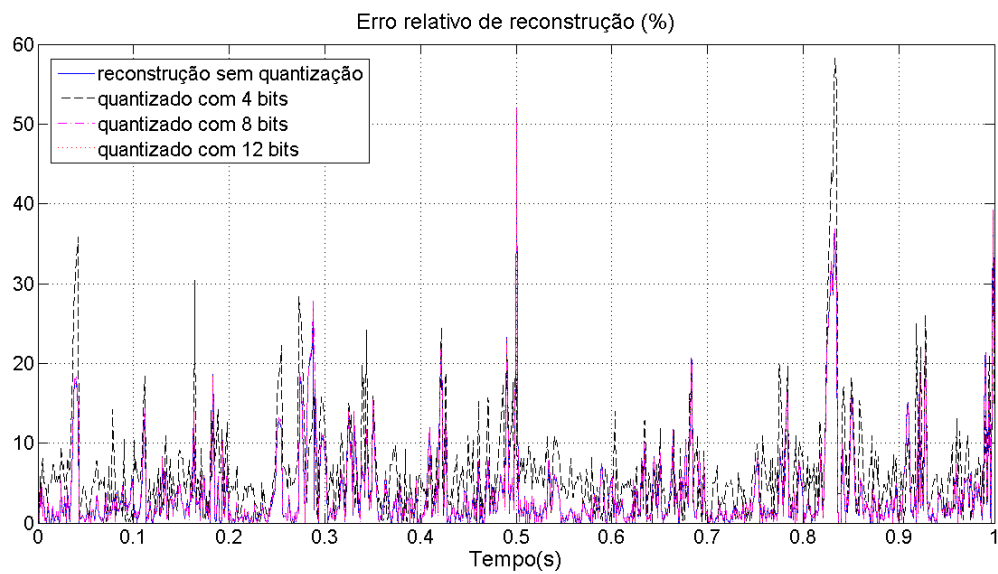


(b) Erros de reconstrução.

Figura 4.10: Sinal 2: reconstrução para transformadas feitas com as bases bior1.1 e bior2.2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

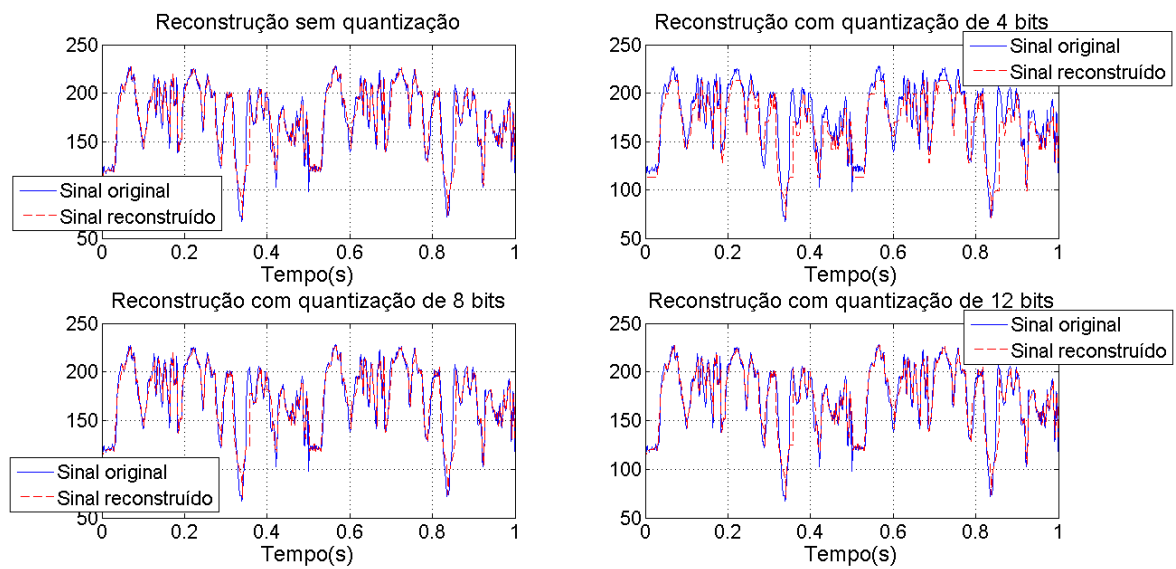


(a) Resultado da reconstrução.

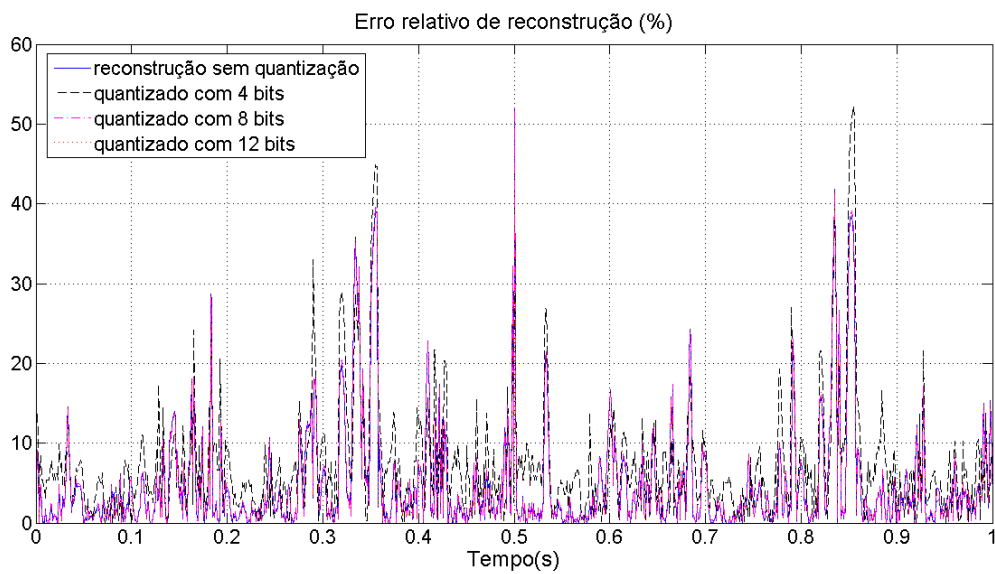


(b) Erros de reconstrução.

Figura 4.11: Sinal 2: reconstrução para transformadas feitas com as bases bior1.1 e bior2.2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.



(a) Resultados da reconstrução.



(b) Erros de reconstrução.

Figura 4.12: Sinal 2: resultados da reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

Tabela 4.14: Erro RMS de reconstrução para o **senal 2**, para **escalas de 1 a 64**. Apenas os coeficientes de Lipschitz foram quantizados.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	13.3291	12.0811	10.4587	9.5713	10.5819
12 bits	13.3371	12.0878	10.4649	9.5771	10.5913
8 bits	13.4612	12.1900	10.6115	9.7176	10.8493
4 bits	15.3192	14.9636	13.4864	11.5869	13.3215

da reconstrução para a base biortogonal com escalas de 16 a 64 apresenta erro relativo que se mantém abaixo de 30% em grande parte do sinal, mas apresenta alguns pontos em que ultrapassa 50%, chegando a 70% de erro em um dos casos. Quando é usada a mesma base mas são aplicadas as escalas de 1 a 64, o resultado fica visivelmente melhor, mas ainda há um pico de erro pouco acima de 50%, mostrado na Figura 4.11(b). Esse erro de 50% parece ser devido a um deslocamento do sinal reconstruído em relação ao original, que pode ser devido aos próprios filtros utilizados e é mostrado na Figura 4.13. Esse comportamento do erro é semelhante ao mostrado na Figura 4.12, quando se mantêm as escalas de 1 a 64, mas são utilizadas as bases gaus1 e gaus2. Assim, mesmo para o pior caso da reconstrução do sinal 2 com escalas de 1 a 64, a reconstrução é feita com menos erros em relação aos casos com escalas de 16 a 64.

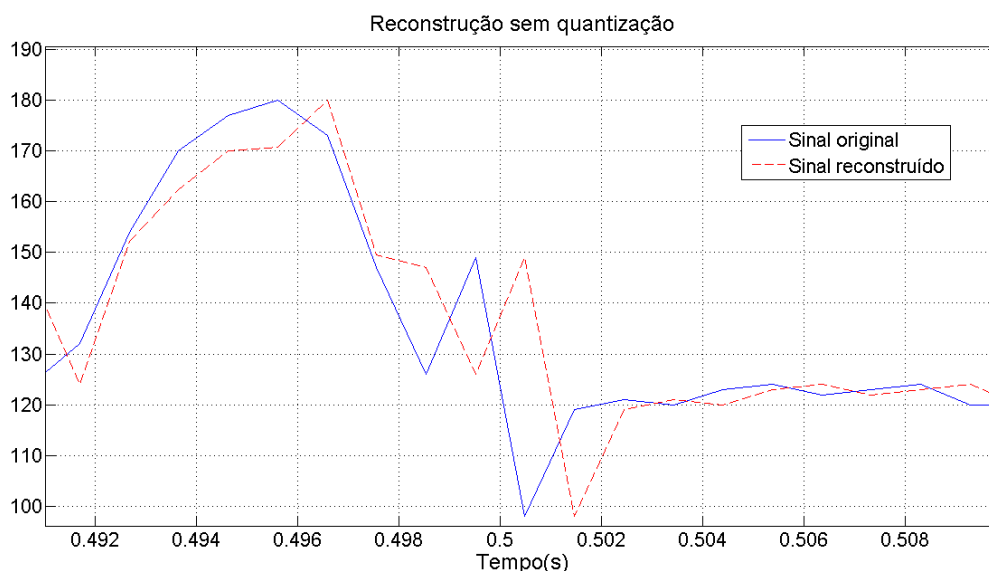


Figura 4.13: *Zoom* do resultado da Figura 4.11(a), evidenciando a defasagem do sinal reconstruído em relação ao sinal original.

A Figura 4.14 mostra o resultado da identificação dos pontos de inflexão para as bases gaus1 e gaus2 com escalas de 1 a 64, que permite observar a necessidade das aproximações do algoritmo.

Nota-se que o número de pontos de inflexão identificados apenas pela linha de coeficientes de maior escala é inferior ao número de pontos necessários à reconstrução, considerando a quantidade de pontos máximos e mínimos locais identificados. Como o sinal 2 apresenta muitas variações bruscas de amplitude em todo seu comprimento, os coeficientes de maior escala não são suficientes para identificar todos os detalhes do sinal.

4.3 Resultados das análises para o sinal 3

O sinal 3 é um sinal natural de eletrocardiograma. Novamente, espera-se que os piores resultados sejam obtidos para as bases gaussianas. No entanto, pelas Tabelas 4.15, 4.16 e 4.17, o maior erro RMS é observado para a base biortogonal reversa com as escalas de 16 a 64, que amostra informações de 4165 pontos. O resultado para esse caso é mostrado na Figura 4.15. Utilizando as escalas de 1 a 64, há uma redução considerável do erro RMS para todas as bases, conforme o disposto nas Tabelas 4.18, 4.19 e 4.20. O resultado para a base biortogonal reversa para esse conjunto de escalas e para quantização das amplitudes e dos coeficientes de Lipschitz é mostrado na Figura 4.16. De todas as simulações realizadas, o menor erro RMS foi obtido para a base biortogonal com as escalas de 1 a 64, com 18775 amostras. Esse resultado é mostrado na Figura 4.17.

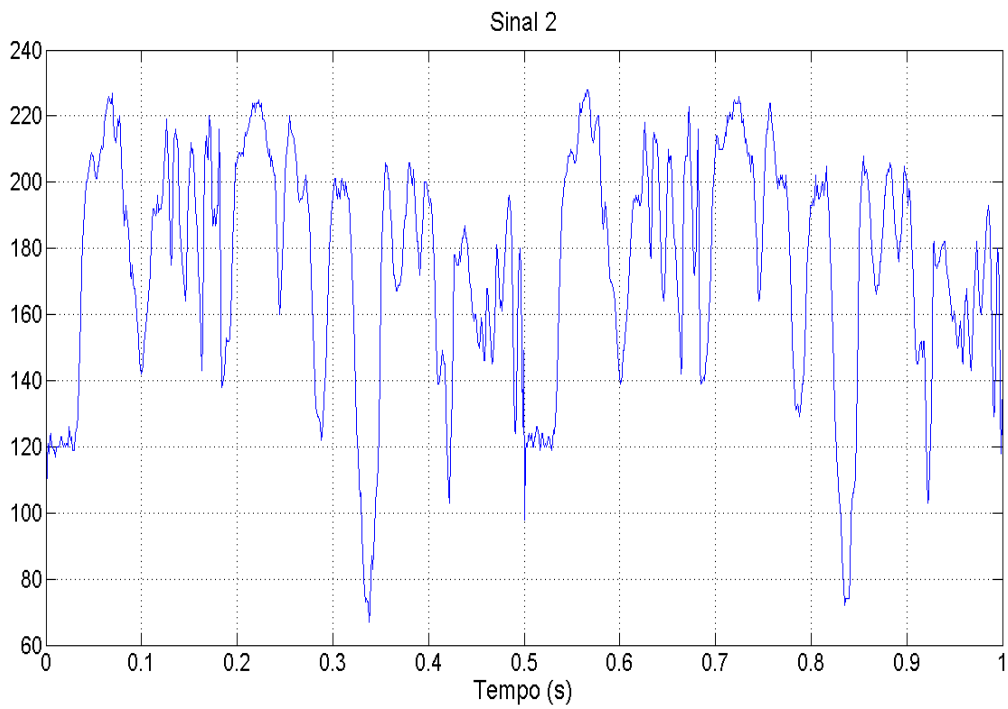
Tabela 4.15: Erro RMS de reconstrução para o **sinal 3**, para **escalas de 16 a 64**.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	51.8304	41.3652	28.2040	23.0982	77.1770
12 bits	51.9074	41.3901	28.3026	23.1489	77.4286
8 bits	53.5361	42.1871	30.6360	24.6831	81.4503
4 bits	127.4224	119.3568	121.2686	106.9198	152.0738

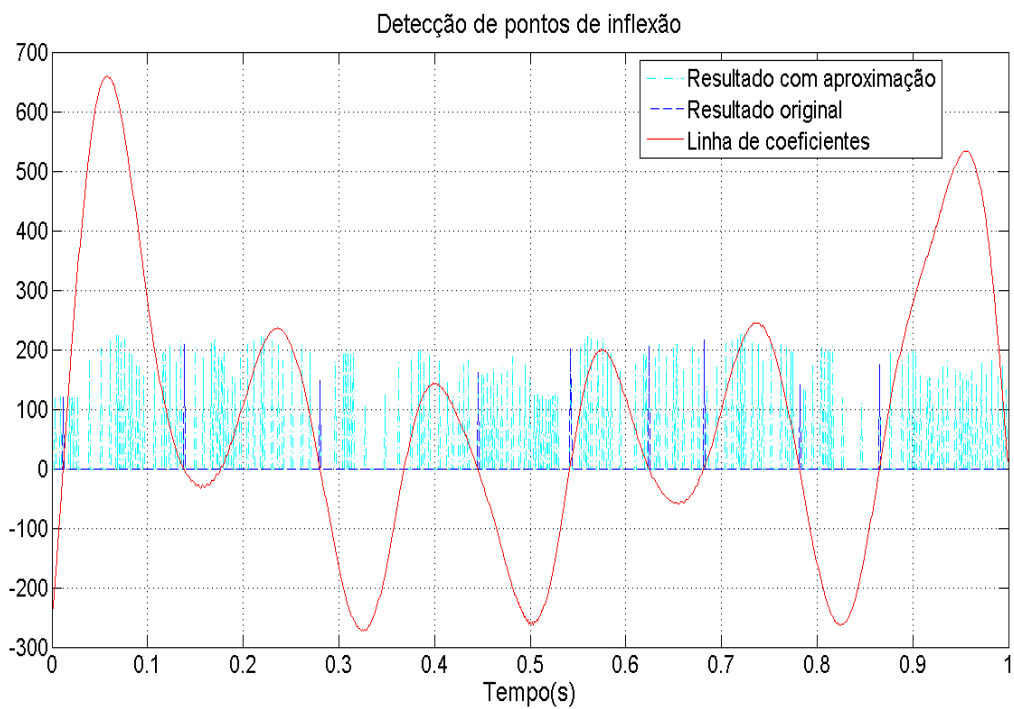
Tabela 4.16: Erro RMS de reconstrução para o **sinal 3**, para **escalas de 16 a 64**, para quantização das amplitudes.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	51.8304	41.3652	28.2040	23.0982	77.1770
12 bits	51.9027	41.3766	28.2524	23.1005	77.2277
8 bits	53.5024	41.8564	29.5966	23.9391	78.3412
4 bits	125.2962	110.1634	115.4594	104.6524	135.7075

A Figura 4.15(b) mostra que, para a base biortogonal reversa com escalas de 16 a 64, o erro relativo nos casos de média e alta resolução permanece bem abaixo de 5% nos trechos entre os picos



(a) Sinal de entrada



(b) Linha de coeficientes de maior escalas e amplitudes amostradas nos pontos de inflexão antes e depois das aproximações de posição

Figura 4.14: Identificação de pontos de inflexão pelo ADC para o sinal 2.

Tabela 4.17: Erro RMS de reconstrução para o **sinal 3**, para **escalas de 16 a 64**, para quantização dos coeficientes de Lipschitz.

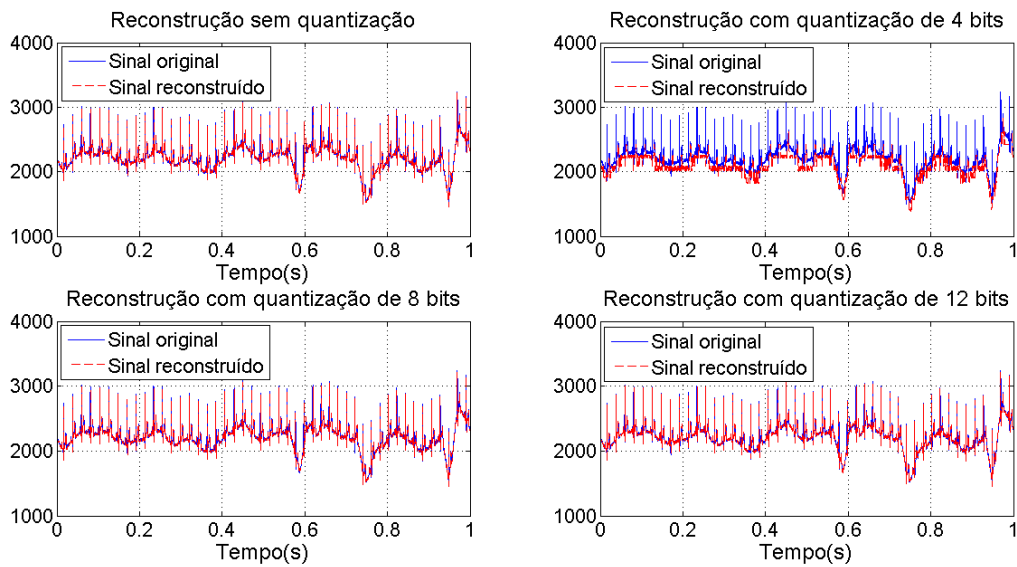
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	51.8304	41.3652	28.2040	23.0982	77.1770
12 bits	51.8349	41.3785	28.2541	23.1465	77.3780
8 bits	51.8016	41.6684	29.2554	23.8616	80.3008
4 bits	53.8782	53.8572	46.0371	34.6476	101.9186

Tabela 4.18: Erro RMS de reconstrução para o **sinal 3**, para **escalas de 1 a 64**.

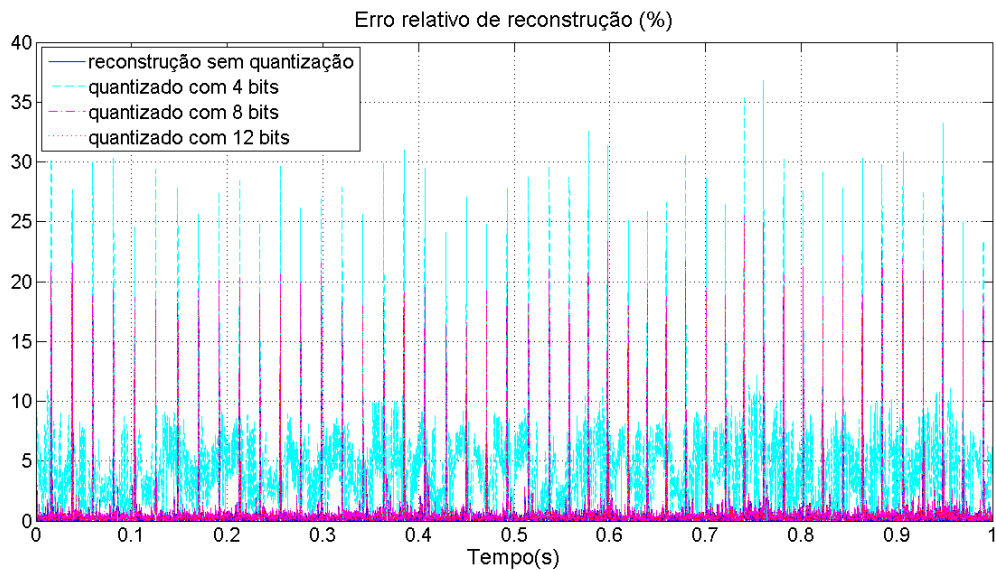
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	40.2495	35.9209	19.1436	13.4650	45.3871
12 bits	40.3600	35.9550	19.2051	13.5376	45.5468
8 bits	42.3603	37.1608	21.2583	16.1282	48.6302
4 bits	131.6136	114.7420	115.0198	112.4148	156.9837

Tabela 4.19: Erro RMS de reconstrução para o **sinal 3**, para **escalas de 1 a 64**, para quantização das amplitudes.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	40.2495	35.9209	19.1436	13.4650	45.3871
12 bits	40.2940	35.9070	19.1672	13.5067	45.4545
8 bits	41.3870	36.2475	20.6152	15.6899	46.9194
4 bits	116.0326	110.0522	113.5700	108.6299	122.7155

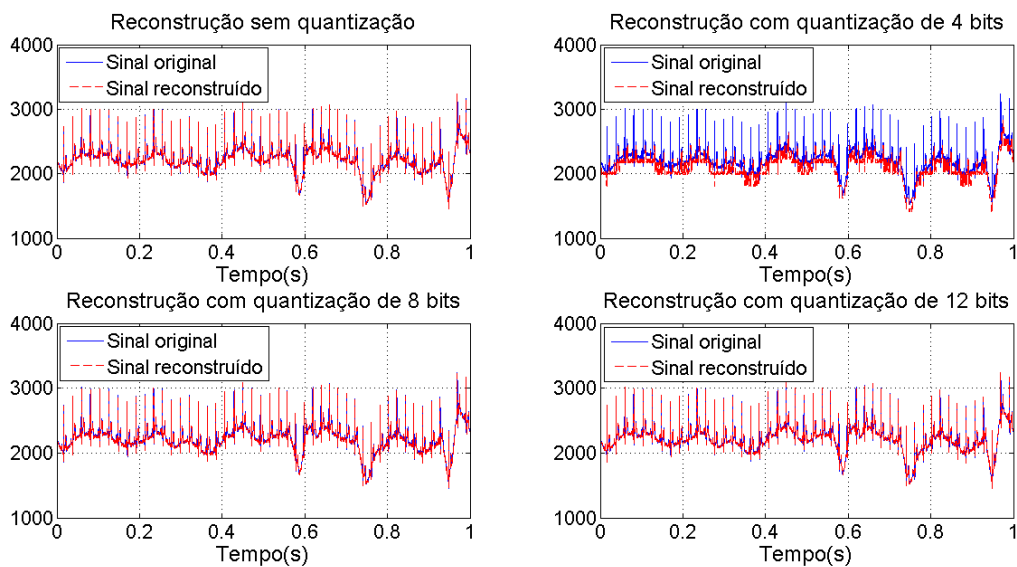


(a) Resultado da reconstrução.

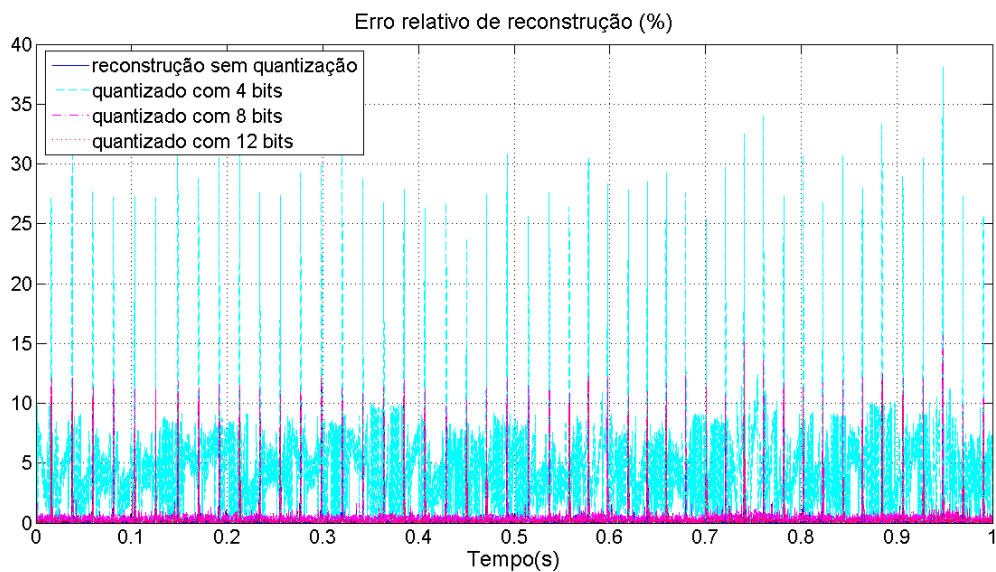


(b) Erros de reconstrução.

Figura 4.15: Sinal 3: reconstrução para transformadas feitas com as bases $rbio1.1$ e $rbio2.2$ e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

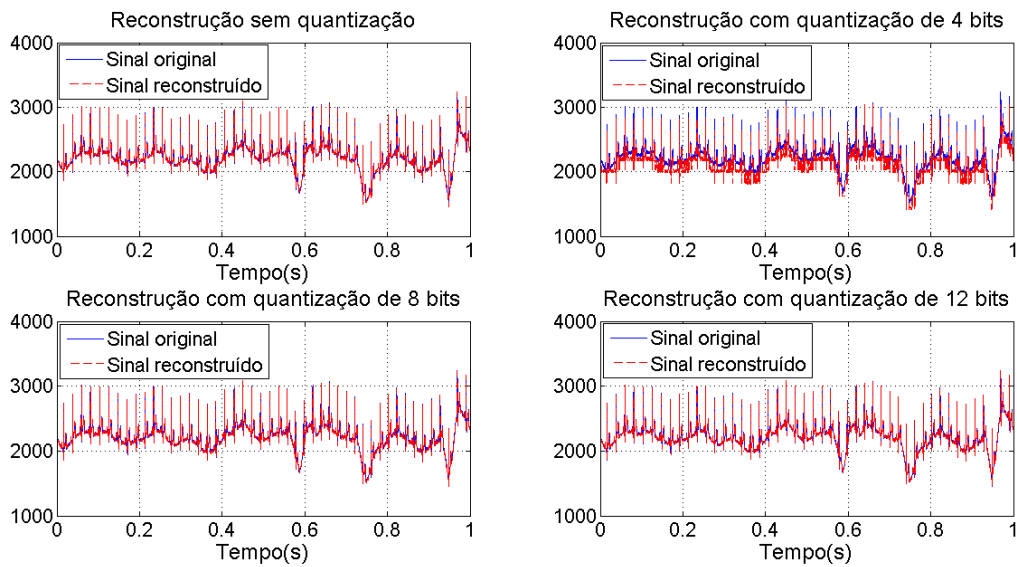


(a) Resultado da reconstrução.

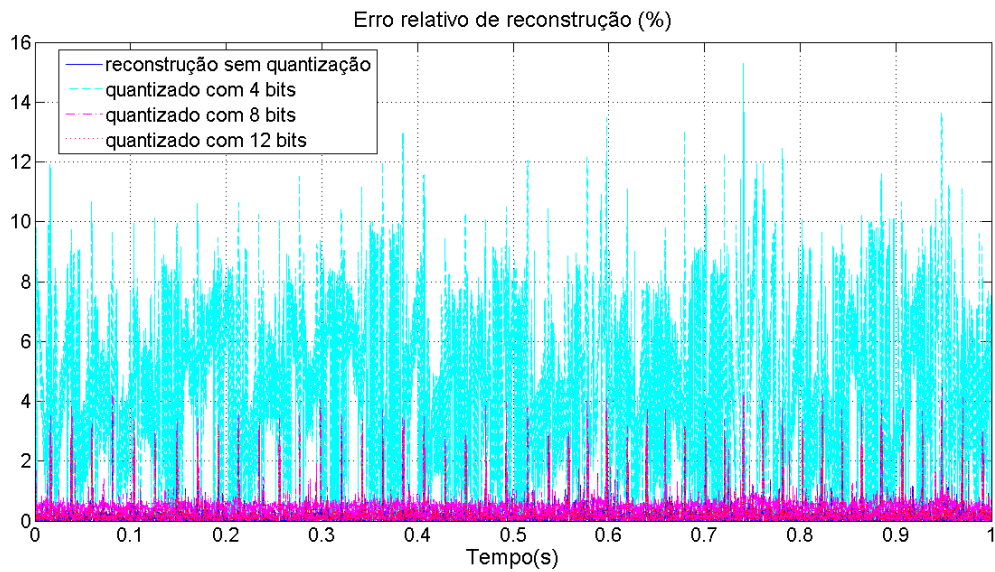


(b) Erros de reconstrução.

Figura 4.16: Sinal 3: reconstrução para transformadas feitas com as bases $rbio1.1$ e $rbio2.2$ e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.



(a) Resultado da reconstrução.



(b) Erros de reconstrução.

Figura 4.17: Sinal 3: reconstrução para transformadas feitas com as bases bior1.1 e bior2.2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

Tabela 4.20: Erro RMS de reconstrução para o **senal 3**, para **escalas de 1 a 64**, para quantização dos coeficientes de Lipschitz.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	40.2495	35.9209	19.1436	13.4650	45.3871
12 bits	40.3155	35.9688	19.1814	13.4958	45.4794
8 bits	41.2333	36.8381	19.7857	13.9400	47.1140
4 bits	72.2049	49.8266	25.6572	28.6520	100.2601

do sinal, e entre 20% e 30% nos picos. Quando são aplicadas as escalas de 1 a 64, o erro relativo nos picos é reduzido para valores entre 10% e 15%, mantendo-se menor nos outros trechos do sinal. Ainda assim, a diferença é gritante quando se altera a base para a biortogonal: na Figura 4.17(b), o erro relativo nos picos, para os casos de média e alta resolução, mantém-se abaixo de 5%, e o erro nos outros trechos do sinal, abaixo de 1%. Vale lembrar que as reconstruções das Figuras 4.16 e 4.17 foram feitas com o mesmo número de amostras do sinal original.

4.4 Resultados das análises para o sinal 4

O sinal 4 também é um sinal natural de eletrocardiograma. Como esperado, o erro de reconstrução para as bases gaussianas é superior ao erro obtido para as outras bases. Isso é notado nas Tabelas 4.21, 4.22 e 4.23, que mostram que o erro RMS observado para as bases gaussianas é superior ao das outras bases em aproximadamente 10 unidades, no mínimo, quando são utilizadas as escalas de 16 a 64. Para esse conjunto de escalas, a reconstrução com as bases gaussianas requer 893 amostras, enquanto a reconstrução com as bases de Daubechies e as biortogonais é feita com a amostragem de 3449 pontos. Quando se utiliza o segundo conjunto de escalas, o erro RMS para as bases gaussianas é reduzido e passa a ter valores próximos aos dos erros obtidos nas reconstruções com as outras bases, como mostrado nas Tabelas 4.24, 4.25 e 4.26. Observa-se menor erro RMS para a base biortogonal reversa com escalas de 1 a 64, que utiliza 8645 amostras na reconstrução do sinal. O resultado para esse caso é mostrado na Figura 4.18. Para esse mesmo conjunto de escalas, a reconstrução com as bases gaussianas é feita com 8275 amostras, quase dez vezes mais amostras que no caso anterior, quando foi utilizado o primeiro conjunto de escalas. Os resultados para as bases gaus1 e gaus2 são mostrados na Figura 4.19, para as escalas de 16 a 64, e na Figura 4.20 para as escalas de 1 a 64.

A Figura 4.18(b) confirma erro de reconstrução pequeno para base biortogonal reversa e escalas de 1 a 64: para os casos de média e alta resolução, o erro relativo mantém-se entre 4% e 6%, enquanto não passa de 13% para o caso de baixa resolução. Esse resultado é próximo ao obtido para bases gaussianas com o mesmo conjunto de escalas: o erro relativo apresenta alguns picos atingindo de 8% a 10% para média e alta resolução, mas mantém-se, em geral, abaixo de 5%. Para

Tabela 4.21: Erro RMS de reconstrução para o **senal 4**, para **escalas de 16 a 64**.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	58.3675	56.9867	39.4528	40.0384	47.4767
12 bits	58.4135	56.9904	39.4767	40.0822	47.5355
8 bits	59.4478	57.4736	40.3064	41.1256	48.8048
4 bits	125.0463	106.2203	104.0816	102.8324	107.7560

Tabela 4.22: Erro RMS de reconstrução para o **senal 4**, para **escalas de 16 a 64**, para quantização apenas das amplitudes.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	58.3675	56.9867	39.4528	40.0384	47.4767
12 bits	58.3626	56.9657	39.4544	40.0576	47.4941
8 bits	59.6504	56.9553	39.8833	40.7302	48.2390
4 bits	108.0906	98.2453	102.4706	99.7341	101.8605

Tabela 4.23: Erro RMS de reconstrução para o **senal 4**, para **escalas de 16 a 64**, para quantização apenas dos coeficientes de Lipschitz.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	58.3675	56.9867	39.4528	40.0384	47.4767
12 bits	58.4184	57.0114	39.4750	40.0630	47.5181
8 bits	59.1582	57.5098	39.8789	40.4393	48.0455
4 bits	77.7189	68.3959	45.2572	45.9202	57.8389

Tabela 4.24: Erro RMS de reconstrução para o **signal 4**, para **escalas de 1 a 64**.

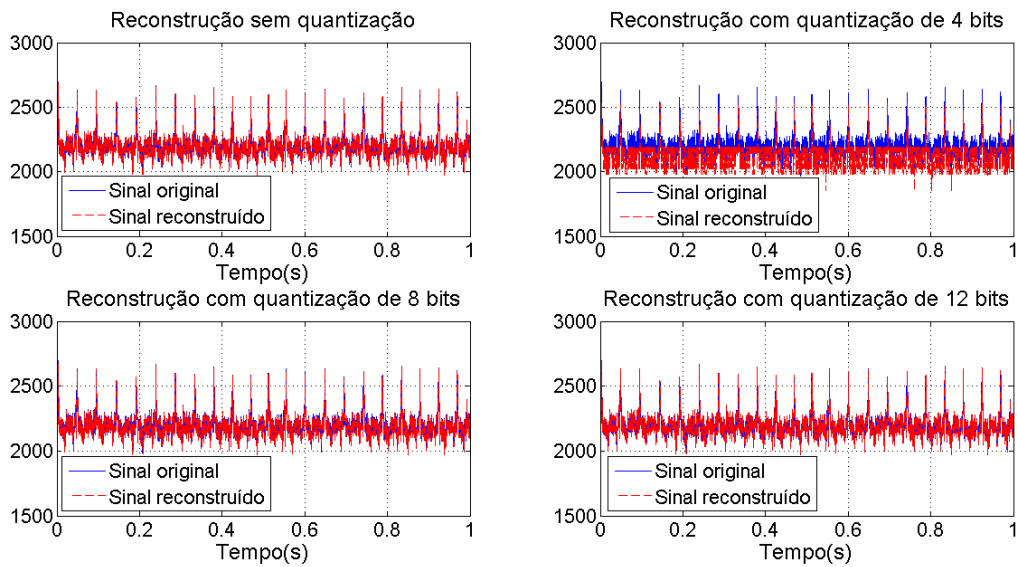
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	24.3212	24.3035	23.6966	24.6003	23.7375
12 bits	24.3228	24.3173	23.7398	24.6139	23.7549
8 bits	25.2197	25.0977	24.9077	25.5989	24.7265
4 bits	94.2186	92.8459	89.8396	90.2197	88.8713

Tabela 4.25: Erro RMS de reconstrução para o **signal 4**, para **escalas de 1 a 64**, para quantização apenas das amplitudes.

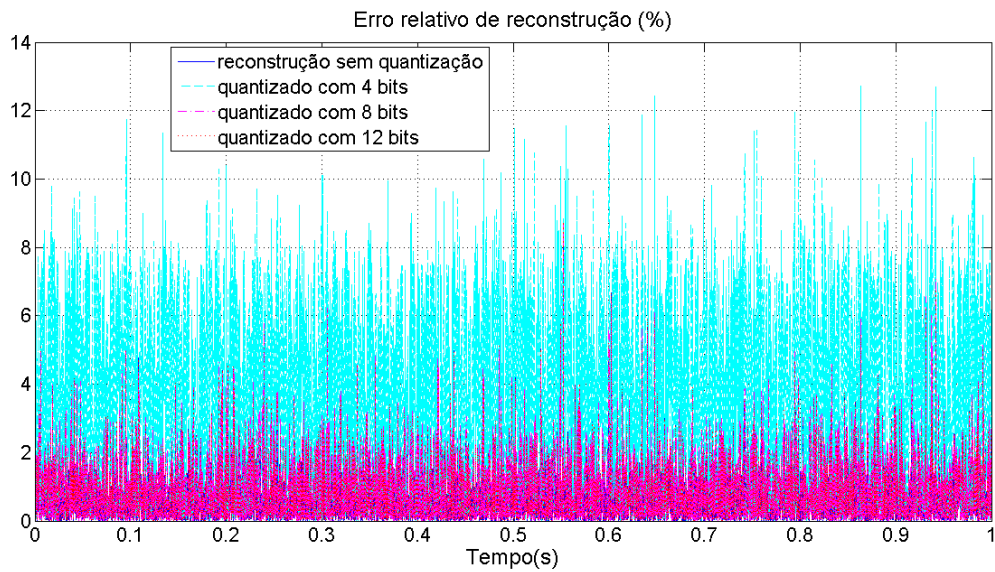
Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	24.3212	24.3035	23.6966	24.6003	23.7375
12 bits	24.3242	24.3063	23.7002	24.6016	23.7416
8 bits	25.0235	24.9954	24.4335	25.2682	24.4641
4 bits	82.4603	91.9765	90.6219	90.6350	89.9738

Tabela 4.26: Erro RMS de reconstrução para o **signal 4**, para **escalas de 1 a 64**, para quantização apenas dos coeficientes de Lipschitz.

Quantização	Bases utilizadas				
	gaus1, gaus2	gaus1, mexh	db1, db2	bior1.1, bior2.2	rbio1.1, rbio2.2
Altíssima resolução	24.3212	24.3035	23.6966	24.6003	23.7375
12 bits	24.3199	24.3145	23.7364	24.6126	23.7507
8 bits	24.5147	24.4091	24.1715	24.9381	24.0119
4 bits	26.8010	26.0587	31.4506	29.6948	28.3919

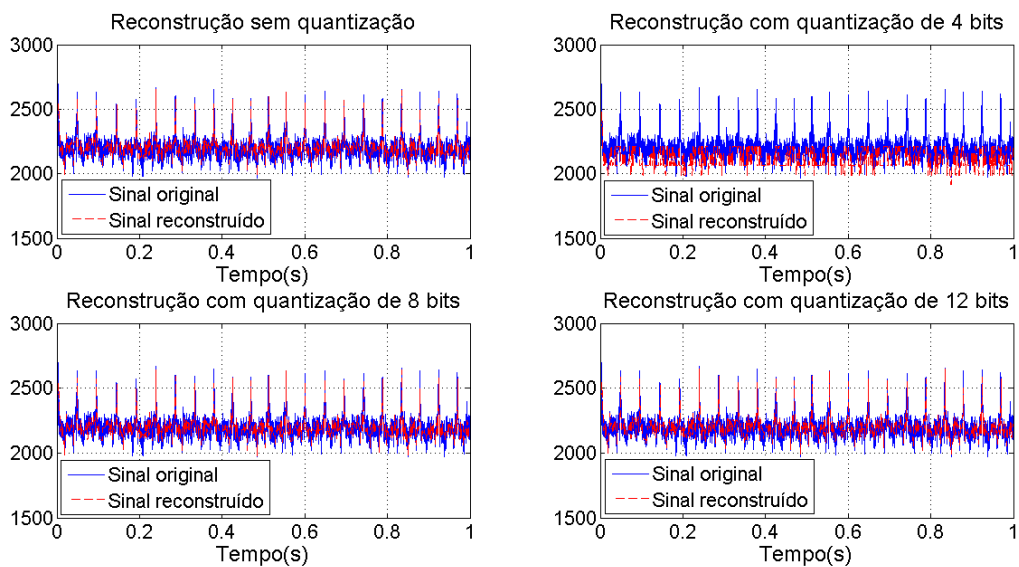


(a) Resultado da reconstrução

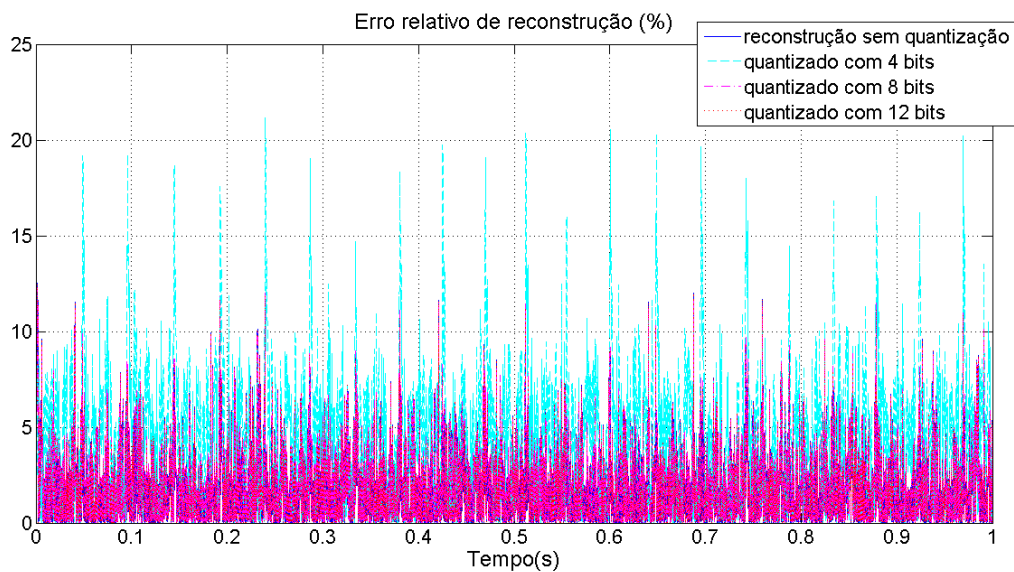


(b) Erros de reconstrução

Figura 4.18: Sinal 4: reconstrução para transformadas feitas com as bases $rbio1.1$ e $rbio2.2$ e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

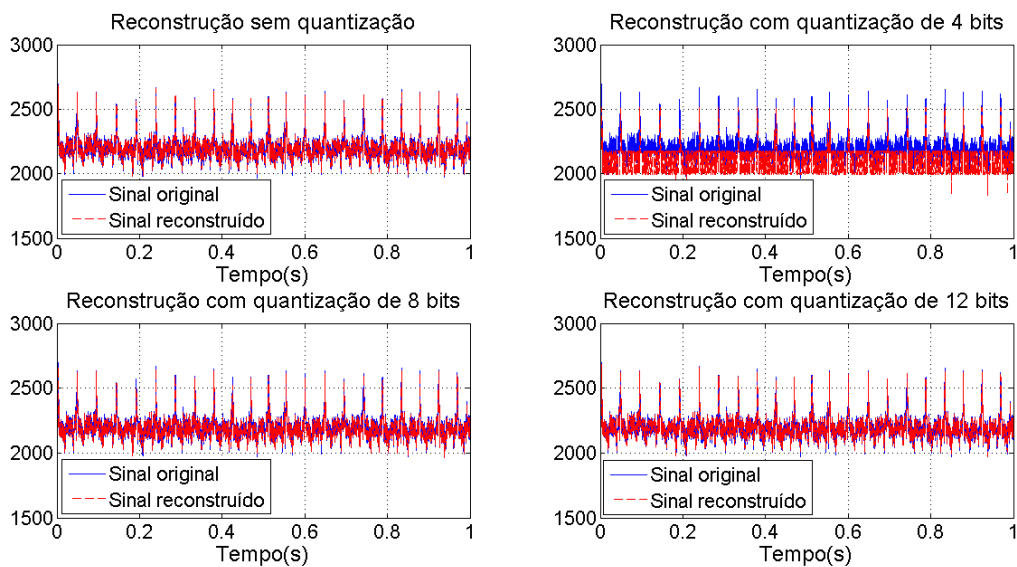


(a) Resultado da reconstrução.

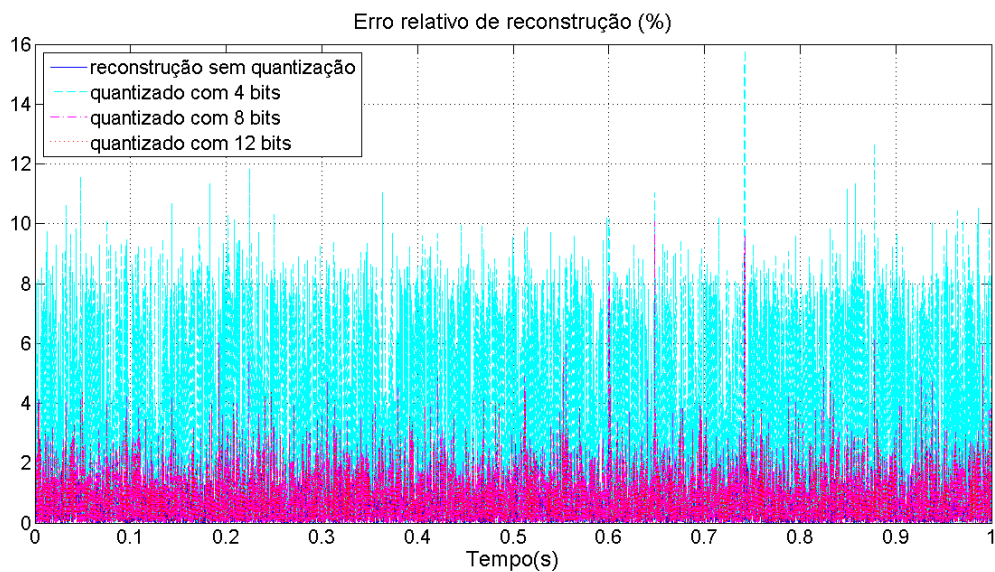


(b) Erros de reconstrução.

Figura 4.19: Sinal 4: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 16 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.



(a) Resultado da reconstrução.



(b) Erros de reconstrução.

Figura 4.20: Sinal 4: reconstrução para transformadas feitas com as bases gaus1 e gaus2 e escalas de 1 a 64, para o caso de resolução altíssima e para quantização com 4, 8 e 12 bits, tanto para as amplitudes quanto para os coeficientes de Lipschitz.

baixa resolução, há um pico do erro relativo em 16%, mas seu valor médio fica entre 8% e 10%. Já para as bases gaussianas em escalas de 16 a 64, a reconstrução apresenta resultado pior. Na Figura 4.19(a), nota-se que muitos pontos do sinal original não foram devidamente identificados, o que explica o resultado para o erro RMS nesse caso. O erro relativo aumenta: para média e alta resolução, atinge 10% em vários picos e mantém-se próximo a 5%. Para baixa resolução, seus picos atingem valores próximos de 20%.

4.5 Análise dos Resultados

Com os resultados, pode-se notar a influência das escolhas das escalas e bases no resultado final. A escolha das escalas está relacionada à resolução desejada na aplicação: para escalas menores, é possível perceber mais detalhes, o que acarreta maior precisão na identificação das posições dos pontos críticos. Foi visto que a quantidade de pontos amostrados aumentou quando se passou do conjunto de escalas de 16 a 64 para o conjunto de 1 a 64 para os sinais 2, 3 e 4: isso ocorre porque, com a escala menor, pontos críticos não identificados antes foram detectados. Já a escolha das bases está relacionada à correlação da base com o sinal avaliado: para o sinal 1, uma função suave, os melhores resultados foram obtidos com as bases gaussianas, enquanto os outros sinais, com variações bruscas, tiveram resultados melhores com as bases de Daubechies e as biortogonais (biortogonal e biortogonal reversa). Vale ressaltar que mesmo nos casos em que não se esperava que as bases gaussianas apresentassem resultado satisfatório, foram observados erros pequenos na reconstrução com essas bases para o segundo conjunto de escalas. Por outro lado, escolhendo-se uma base de alta correlação com o sinal, pode-se obter erros pequenos e resultados satisfatórios na reconstrução mesmo com o primeiro conjunto de escalas, o que contribui para a redução do número de pontos necessários para reconstruir o sinal original com precisão. Por exemplo, para o sinal 4, um dos resultados de menor erro foi obtido com a base biortogonal reversa, apresentando erro RMS, para o caso sem quantização, de 0.0400 com escalas de 16 a 64 e de 0.0246 com escalas de 1 a 64, quase metade do erro observado no primeiro caso. Enquanto isso, para o primeiro caso são utilizadas 3449 amostras contra 8645 no segundo caso, isto é, o número de amostras necessárias para reconstrução fica pouco acima do dobro do valor observado no primeiro caso. Nesse contexto, para aplicações de média resolução, pode compensar utilizar escalas maiores para reduzir o número de amostras, já que esse parâmetro pode elevar o consumo de potência.

Ainda no que diz respeito ao número de amostras utilizadas em cada caso, observou-se que a compressão dos dados do sinal foi maior para o sinal 1, que foi reconstruído com pequeno erro a partir das informações de 9 dos 1000 pontos originais. Essa razão aumenta para os outros sinais, mas verifica-se que, nos casos em que há maior número de amostras, esse valor pode chegar a aproximadamente metade do comprimento do sinal original, principalmente quando são utilizadas as escalas de 1 a 64, que identificam, em geral, uma quantidade de pontos muito superior ao caso com escalas de 16 a 64. Portanto, há compressão para todos os casos observados. Além disso, o fato de que o número de amostras varia de acordo com o sinal de entrada e os parâmetros escolhidos mostra que os conversores implementados não mantêm sua taxa de amostragem fixa, ao contrário dos conversores típicos.

Quanto à quantização, todas as verificações mostraram que as reconstruções com média e alta resolução, ou seja, para 8 e 12 bits, ficaram muito próximas do resultado obtido sem quantização. Já para resolução baixa, 4 bits, o resultado ficou mais próximo para o sinal 1, que é mais bem comportado, e permitiu identificar o comportamento do sinal. Isso já não ocorre para os outros sinais; especificamente, no caso dos sinais 3 e 4, de eletrocardiograma, foi observado que a reconstrução com 4 bits pode acarretar a perda completa da informação dos picos do sinal. Além disso, a diferença entre os resultados para reconstruções com 8 e 12 bits foi pequena, indicando que, dependendo da aplicação, o esforço necessário para aumentar a resolução de 8 para 12 bits talvez não compense a leve diferença nos resultados. Ainda, se for desejada uma aplicação de alta resolução, não deve ser necessário utilizar mais que 12 bits, considerando que os erros para esse caso ficaram praticamente iguais aos erros obtidos no caso sem quantização. Essa análise vale para a quantização da amplitude. Já sobre o coeficiente de Lipschitz, a quantização não tem o mesmo efeito, porque verificou-se que os resultados para 8 e 12 bits e para altíssima resolução ficaram muito próximos, indicando que, para aplicações de alta resolução, não são necessários mais que 8 bits para a representação do coeficiente de Lipschitz. Ainda, os resultados para baixa resolução não apresentaram uma diferença tão gritante com relação às outras resoluções quanto àquela observada para a quantização da amplitude. Assim, a quantização com apenas 4 bits para o coeficiente de Lipschitz pode ser suficiente para aplicações que não requeiram alta resolução nos resultados.

Um problema identificado foi a defasagem do sinal reconstruído em relação ao sinal original. Como as métricas de erro utilizadas são baseadas em comparações dos valores dos dois sinais ponto a ponto, elas não refletem diretamente essa defasagem, e são afetadas por elas. Assim, provavelmente os erros observados ainda poderão ser reduzidos se a diferença de fase entre os dois sinais for eliminada.

Outra característica observada é que a identificação dos pontos de inflexão dependeu muito das aproximações feitas no algoritmo para os sinais menos bem comportados, como foi mostrado no caso do sinal 2 para bases gaus1 e gaus2 e escalas de 1 a 64. A escolha da linha de coeficientes de maior escala para identificação dos pontos de inflexão foi heurística e baseada nos conceitos teóricos e, apesar de as aproximações do algoritmo terem permitido obter resultados aparentemente bons, novas estratégias de identificação dos pontos de inflexão pelas linhas de coeficientes da transformada wavelet podem ser objeto de trabalhos futuros. Algumas possibilidades são utilizar uma linha de coeficientes de menor escala ou utilizar as informações de mais de uma dessas linhas.

Também nota-se que os erros apresentam picos nos trechos inicial e final dos sinais reconstruídos. Isso pode ocorrer porque nas bordas, isto é, nos instantes inicial e final, o cálculo do Lipschitz é feito com menos informação de sua vizinhança, o que afeta a caracterização do sinal em torno desses pontos. Esse comportamento pode caracterizar uma dificuldade de implementação do circuito analógico, pois indica que a estimativa do Lipschitz é não-causal.

Capítulo 5

Conclusão

Neste trabalho foi avaliado e testado o conversor analógico-digital baseado em transformada wavelet desenvolvido no trabalho de conclusão de curso do aluno José Alberto Alves de Andrade, da Faculdade UnB—Gama [1] e, nesse processo, foram feitas algumas modificações em seu algoritmo. A primeira foi na estratégia de usar valores de limiar para gerar os sinais que identificam os pontos críticos, que foi substituída por um comparador que identifica os instantes em que as linhas de coeficientes da transformada wavelet têm valor zero; para detectar singularidades, foi utilizada a linha de coeficientes de menor escala e, para pontos de inflexão, a linha de coeficientes da maior escala aplicada. A segunda modificação foi passar a amostrar a amplitude do sinal de entrada também nos pontos de inflexão detectados, e não só nos pontos de singularidade; ainda, foram incluídos como máximos ou mínimos locais os pontos inicial e final do sinal de entrada do conversor A/D, para que sua reconstrução pudesse ser realizada de acordo com o algoritmo desenvolvido para o DAC. O conversor D/A desenvolvido reconstrói o sinal deslocando no tempo e na amplitude a função apresentada na Eq. 2.22 e na Figura 2.10, cujo comportamento é alterado de acordo com o expoente α , ou seja, o coeficiente de Lipschitz.

O DAC permite a avaliação do processo de conversão A/D. Na validação, foram aplicados diferentes sinais de entrada, bases e escalas; além disso, foi verificado o efeito da quantização, que ainda não havia sido levado em conta no algoritmo do ADC. Foi visto que a escolha das bases e escalas pode afetar consideravelmente o resultado da reconstrução, portanto, esses parâmetros devem ser definidos de acordo com a aplicação, lembrando que os melhores resultados são obtidos para bases de maior correlação com o sinal avaliado e para escalas pequenas, com a ressalva de que o número de pontos identificados pelo ADC e utilizados na reconstrução é sensível a essas mudanças. Dessa forma, a escolha das bases e escalas na transformada wavelet permitem selecionar a resolução e a velocidade de conversão de acordo com a aplicação desejada. Quanto ao efeito da quantização, para as amplitudes, os resultados para resoluções de 8 e 12 bits (média e alta resolução) foram próximos ao resultado verificado antes da inclusão do quantizador, enquanto a resolução de 4 bits acarretou aumento considerável do erro de reconstrução em todos os casos observados. Para os coeficientes de Lipschitz, a quantização com 8 bits mostrou-se suficiente para aplicações de alta resolução; ainda, a resolução de 4 bits forneceu resultados próximos àqueles verificados para 8 e 12 bits, o que indica que pode ser utilizada para aplicações que não requeiram alta resolução.

Outra característica importante do ADC é que o número de pontos amostrados variou de acordo com o sinal aplicado e os parâmetros utilizados. O ADC é orientado a eventos e, dessa forma, não é necessário manter sua taxa de amostragem constante. Ainda, para o sinal 1, verificou-se a maior compressão da informação, enquanto para os sinais 2, 3 e 4, foram atingidos resultados de alta resolução com número de pontos menor ou igual à metade do comprimento do respectivo sinal original em MATLAB.

Com base no exposto, espera-se que o consumo de potência devido à amostragem seja reduzido com relação aos ADCs típicos conhecidos, já que a taxa de amostragem é alterada de acordo com o sinal de entrada. Quanto ao consumo de potência devido às operações realizadas no algoritmo do ADC para detecção de pontos e estimativa do coeficiente de Lipschitz, espera-se que ele não seja elevado: o coeficiente de Lipschitz é calculado usando logaritmos e operações matemáticas básicas; os sinais de detecção de pontos podem ser obtidos com um circuito simples com comparador; e a transformada wavelet pode ser realizada com bancos de filtros. Dependendo da aplicação, pode ser utilizado o quantizador com resolução de 8 bits, que apresentou baixos erros na reconstrução dos sinais e consome menos potência que o de 12 bits.

Portanto, os resultados indicam que o ADC proposto com as modificações sugeridas fornece informações suficientes para a reconstrução do sinal de entrada para aplicações de média e alta resolução, com a expectativa de redução da potência consumida em relação aos conversores A/D típicos conhecidos.

5.1 Trabalhos futuros

Com base na discussão presente no texto, possíveis desenvolvimentos futuros para este trabalho incluem:

- Comparar o ADC e o DAC propostos com conversores triviais;
- Avaliar os sinais de entrada no domínio da frequência para obter a quantidade de amostras necessárias para reconstrução de acordo com o Teorema de Nyquist, e comparar esse número com a quantidade de amostras utilizadas pelo algoritmo proposto na reconstrução do sinal, visando a quantificar a compressão dos dados dos sinais;
- Investigar o motivo da defasagem do sinal reconstruído em relação ao sinal original, como foi observado para o sinal 2, e verificar os valores dos erros após essa correção;
- Alterar o algoritmo do DAC para que se possa reconstruir um sinal com um ou mais pontos de inflexão entre um máximo e um mínimo locais;
- Avaliar outras estratégias de reconstrução do sinal e comparar com os resultados obtidos para o DAC proposto neste trabalho;
- Verificar novas alternativas ao algoritmo de detecção de pontos de inflexão, visando a diminuir ou eliminar a necessidade de aproximações;

- Projetar e implementar o circuito correspondente aos conversores A/D e D/A, com a finalidade de verificar o consumo de potência no processo.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ANDRADE, J. A. A. de. Projeto de um conversor a/d de baixo consumo para aplicações de alta resolução. TCC 1. Faculdade UnB Gama, Universidade de Brasília. 2014.
- [2] BAKER, R. J. *CMOS Circuit Design, Layout and Simulation*. 3rd. ed. Hoboken, NJ: Wiley-IEEE Press, 2010.
- [3] ALLEN, P. E.; HOLBERG, D. R. *CMOS Analog Circuit Design*. 2nd. ed. New York, NY: Oxford University Press, 2002.
- [4] MATHWORKS Documentation: Continuous Wavelet Transform. <http://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform.html>. Acessado em 10/11/2014.
- [5] HADDAD, S. A.; SERDIJN, W. A. *Ultra-low Power Biomedical Signal Processing: An Analog Wavelet Filter Approach For Pacemakers*. New York, NY: Springer, 2009.
- [6] MALLAT, S. *A Wavelet Tour of Signal Processing*. 2nd. ed. San Diego, CA: Academic Press, 1999.
- [7] ECG Processing — R-peaks Detection. <http://www.librow.com/cases/case-2>. Acessado em 26/11/2014.
- [8] RAZAVI, B. *Design of Analog CMOS Integrated Circuits*. International. Singapore: McGraw-Hill, 2001.
- [9] MATHWORKS Documentation: Wavelet Toolbox. <http://www.mathworks.com/help/wavelet/index.html>. Acessado em 10/11/2014.
- [10] STRANG, G.; NGUYEN, T. *Wavelets and Filter Banks*. Revised edition. Wellesley, MA: Wellesley-Cambridge, 1997.
- [11] DAUBECHIES, I. *Ten Lectures on Wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992. (CBMS-NSF Regional Conference Series in Applied Mathematics).
- [12] COHEN, A.; DAUBECHIES, I.; FEAUVEAU, J.-C. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, v. 45, n. 5, p. 485–560, 1992.
- [13] UNSER, M. Ten good reasons for using spline wavelets. In: *Wavelets Applications in Signal and Image Processing V*. San Diego, CA: SPIE Proceedings, 1997. v. 3169, p. 422–431.

ANEXOS

I. *SCRIPTS* DE MATLAB

I.1 Funções utilizadas

Esta seção mostra os *scripts* de todas as funções utilizadas nos algoritmos do ADC e do DAC.

I.1.1 Cálculo do coeficiente de Lipschitz.

A função de estimativa do coeficiente de Lipschitz originalmente proposta em [1] foi modificada para que a transformada wavelet pudesse ser computada da linha de comando do MATLAB.

I.1.1.1 Função original [1]

```
1 function [lpz] = fz_lpz_calc(coef)
2
3 dim = size(coef.coefs)
4 coef.coefs = abs(coef.coefs);
5
6     a = log2(coef.coefs);
7     for n=1:dim(2)
8         alpha(:,n) = diff(a(:,n)') ./ diff(log2(coef.scales));
9     end
10
11 lpz = sum(alpha) / (dim(1)-1);
```

I.1.1.2 Função modificada

```
1 function [lpz] = fz_lpz_calc_modif(coef,scales)
2
3 dim = size(coef);
4
5 coef = abs(coef);
6
7 a = log2(coef);
8
9 for n=1:dim(2)
10     alpha(:,n) = diff(a(:,n)') ./ diff(log2(scales));
11 end
12
13 lpz = sum(alpha) / (dim(1)-1);
```

I.1.2 Quantizador

```
1 function [ output_code ] = quant(input, N)
2 % Converte um valor analógico em inteiro
3 %   input = input value
4 %   N = resolution
5
6 LSB = 1/(2^N);
7 code = zeros(1,length(input));
8
9 % Quantizer process
10 for j = 1:length(input)
11     for k = 1:(2^N)-1
12         if(input(j) > k*LSB)
13             code(j) = k*LSB;
14         else
15             break;
16         end
17     end
18 end
19
20 output_code = (code);
21 end
```

I.1.3 Reconstrução do sinal

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % UnB - FT - EnE
3 % Isadora F. Martins - 10/0013147
4 %
5 % Função de reconstrução polinomial
6 %   Entradas: parâmetros do polinômio
7 %   Saída: resultado da reconstrução
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 function [result] = reconstruct(num_pol,A_i,A_s,am_lpz,tempo,delta,tau)
11
12 tempo = [];
13 pol = [];
14 result = [];
15
16 for k = 1:num_pol
17     tempo = tempo(k):delta:tempo(k+1);
18
19     nu = tempo(k+1) - tempo(k); % suporte no tempo
20
21     seli = ceil(k/2);           % seleciona info relacionada a inflexão
```

```

22     sels = floor(k/2 + 1);    % seleciona info relacionada a singularidade
23
24     pol = A_i(seli) + (A_s(sels)-A_i(seli))*...
25           (1-(abs((tau(sels) - tempo)./nu).^am_lpz(sels)));
26
27     % Montagem do sinal
28     if k==1
29         result = pol;
30     else
31         result = [result pol];
32         result = result(1,1:length(result)-1);
33     end
34 end

```

I.1.4 Cálculo de erros

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % UnB - FT - EnE
3  % Isadora F. Martins - 10/0013147
4  %
5  % Função de cálculo do erro de aproximação
6  %     Entradas: sinais original e reconstruído
7  %     Saídas: erros absoluto e relativo
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 function [erro_abs,erro_rel,rms] = erros_calc(result,sig)
11
12 erro_abs = abs(result-sig);           % erro absoluto
13
14 erro_rel = 100*erro_abs./sig;        % erro relativo percentual
15 %erro_rel_max = max(erro_rel)        % valor máximo do erro relativo (%)
16
17 rms = sqrt(sum((sig(:)-result(:)).^2)/numel(sig));    % erro RMS

```

I.2 Sinais de entrada

Os *scripts* a seguir foram desenvolvidos para gerar os diferentes sinais de entrada.

I.2.1 Sinal 1 [1]

```

1 clear;
2 clc;
3
4 t = linspace(0,1,1000);

```

```
5 sig = gaussmf(t, [0.1 0.2]) + fz_p_sandro(t, 1, 0.75, 0.7, 0.6);
```

```
1 function [f] = fz_p_sandro(t,A,v,tal,lipsc)
2
3 k = 1/(v)^lipsc;
4 f = A*k*(v^lipsc-abs(t-tal).^lipsc);
5 for n=1:length(t)
6
7     if(f(n)<0)
8         f(n)=0;
9     end
10 end
11 }
```

I.2.2 Sinal 2

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % UnB - FT - EnE
3 % Isadora F. Martins - 10/0013147
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5
6 clear;
7 clc;
8
9 i = imread('monarch.bmp');
10 i = double(i);
11 sig = [i(:,1,1) ; i(:,2,1)];
12 sig = sig';
13 t = linspace(0,1,length(sig));
```

I.2.3 Sinal 3

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % UnB - FT - EnE
3 % Isadora F. Martins - 10/0013147
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5
6 clear;
7 clc;
8
9 load ecgdemodata1
10 sig = ecg;
11 t = linspace(0,1,length(sig));
```

I.2.4 Sinal 4

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % UnB - FT - EnE
3 % Isadora F. Martins - 10/0013147
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5
6 clear;
7 clc;
8
9 load ecgdemodata2
10 sig = ecg;
11 t = linspace(0,1,length(sig));
```

I.3 ADC

I.3.1 Amostragem

```
1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 % UnB - FT - EnE
4 % Isadora F. Martins - 10/0013147
5 %
6 % Rotina genérica que implementa o ADC.
7 %     Faz a média dos pontos de inflexão quando há mais de um.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10
11 %% Faz a CWT pela linha de comando:
12
13 % Selecionar escalas:
14
15 %escalas = [16:16:64];
16 escalas = [1:1:64];
17
18 % Selecionar base:
19
20 %base1 = 'gaus1';
21 %base1 = 'db1';
22 %base1 = 'bior1.1';
23 base1 = 'rbior1.1';
24
25 %base2 = 'gaus2';
26 %base2 = 'mexh';
27 %base2 = 'db2';
28 %base2 = 'bior2.2';
```



```

29 base2 = 'rbio2.2';
30
31 coefic = cwt(sig,escalas,base1);
32 coefic2 = cwt(sig,escalas,base2);
33
34 %% Escolher linhas de coeficientes
35
36 % Linha de coeficientes (menor escala) da transformada de primeira ordem:
37 %linha_coef = coefic.coefs(1,:);
38 linha_coef = coefic(1,:);
39
40 % Linha de coeficientes (maior escala) da transformada de segunda ordem:
41 %s2 = size(coefic2.coefs);
42 %linha_coef2 = coefic2.coefs(s2(1),:);
43 s2 = size(coefic2);
44 linha_coef2 = coefic2(s2(1),:);
45
46 % Identifica pontos especiais:
47 sing = +(linha_coef >= 0);
48 inflexao = +(linha_coef2 >= 0);
49
50 %% Coeficientes de Lipschitz calculados pela função do José
51 %lpz = abs(fz_lpz_calc(coefic2) - 0.5);
52
53 %modificada para cwt pela linha de comando:
54 lpz = abs(fz_lpz_calc_modif(coefic2,escalas) - 0.5);
55 %% Identificar e eliminar pontos de inflexão a mais
56
57 % Máscara para pontos de singularidade + inicial e final
58 sing2 = [sing(1:length(sing)-1) 2];
59 mask_sing = +(abs(diff([2 sing2])) > 0);
60
61 % Máscara para pontos de inflexão
62 mask_infl = +(abs(diff([0 inflexao])) > 0);
63
64 %% Verificar se nenhum ponto de inflexão foi identificado na mesma posição
65 %% de um ponto de singularidade:
66
67 verif = mask_sing + mask_infl;
68
69 for k = 1:length(sig)
70     if verif(k)==2
71         mask_infl(k) = 0;
72     end
73 end
74
75 %% Identificar amostras com as máscaras
76
77 p = 1:1:length(t);           % vetor de posições
78
79 pos_sing = p.*mask_sing;     % vetor com posições de singularidades ~= 0
80

```

```

81 pos_infl = p.*mask_infl;    % vetor com posições de inflexão ~= 0
82
83 % Retira zeros de pos_sing:
84
85 pos_sing2 = [];           % vetor apenas com posições de sing. + i/f
86
87 for k=1:length(sig)
88     if pos_sing(k)~=0
89         pos_sing2 = [pos_sing2 pos_sing(k)];
90     end
91 end
92
93 % Identifica e elimina pontos de inflexão a mais, substituindo todos no
94 %trecho pela média das posições dos pontos identificados:
95
96 pos_infl2 = [];         % vetor apenas com posições de inflexão
97
98 for k = 2:length(pos_sing2)
99
100     c1 = pos_sing2(k-1);
101     c2 = pos_sing2(k);
102
103     x = sum(mask_infl(1,c1:c2));
104     y = sum(pos_infl(1,c1:c2));
105
106     if x == 0
107         x=1;
108         y=round((c1+c2)/2);
109     end
110
111     pos_infl2 = [pos_infl2 round(y/x)];
112
113 end
114
115
116 %% Parâmetros do polinômio
117
118 %% Número de polinômios
119 num_pol = length(pos_sing2) + length(pos_infl2) -1;
120
121 %% Amplitudes nos pontos de inflexão
122 A_i = sig(pos_infl2);
123
124 %% Amplitudes nos pontos de singularidade
125 A_s = sig(pos_sing2);
126
127 %% Coeficientes de lipschitz nas sing. + pontos inicial/final
128 am_lpz = lpz(pos_sing2);
129
130 %% Valores no tempo
131
132 amostras = sort([pos_sing2 pos_infl2]);    % ordena posições de interesse

```

```

133
134 tempos = t(amostras);           % instantes de interesse
135
136 delta = (t(length(t))-t(1))/length(t);   % passo de tempo
137
138 % Instantes dos pontos de singularidade
139 tau = t(pos_sing2);
140
141 %% Informações amostradas
142
143 am_amp_sing = mask_sing.*sig;   % amplitudes de pontos de singularidade
144 am_amp_infl = mask_infl.*sig;   % amplitudes de pontos de inflexão antes
145 am_amp_infl2 = zeros(size(sig));   % amplitudes para
146 am_amp_infl2(pos_infl2) = sig(pos_infl2);   % pontos de inflexão depois
147 am_lpz2 = zeros(size(sig));   % coeficiente de lipschitz
148 am_lpz2(pos_sing2) = am_lpz;   % nos pontos selecionados
149
150 %% Gráficos do ADC
151
152 figure
153 subplot(3,1,1)
154 plot(t,linha_coef,t,sing)
155 title('Sinais de detecção de máximos, mínimos e singularidades','FontSize',18)
156 legend('Linha de coeficientes','Saída do comparador')
157 xlabel('Tempo(s)', 'FontSize',18)
158 set(gca,'FontSize',16)
159 grid on
160
161 subplot(3,1,2)
162 plot(t,am_amp_sing)
163 title('Amplitudes nos pontos de máximo/mínimo/singularidade','FontSize',18)
164 xlabel('Tempo(s)', 'FontSize',18)
165 set(gca,'FontSize',16)
166 grid on
167
168 subplot(3,1,3)
169 plot(t,am_lpz2)
170 title('Coeficientes de lipschitz amostrados','FontSize',18)
171 xlabel('Tempo(s)', 'FontSize',18)
172 set(gca,'FontSize',16)
173 grid on
174
175 figure
176
177 subplot(2,1,1)
178 plot(t,linha_coef2,t,inflexao)
179 title('Sinais de detecção de pontos de inflexão(sem aproximações)','FontSize',18)
180 legend('Linha de coeficientes','Saída do comparador')
181 xlabel('Tempo(s)', 'FontSize',18)
182 set(gca,'FontSize',16)
183 grid on
184

```

```

185 subplot(2,1,2)
186 plot(t,am_amp_infl,t,am_amp_infl2)
187 legend('Original','Com aproximação')
188 title('Amplitude nos pontos de inflexão','FontSize',18)
189 xlabel('Tempo(s)','FontSize',18)
190 set(gca,'FontSize',16)
191 grid on

```

I.3.2 Quantização

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % UnB - FT - EnE
3 % Isadora F. Martins - 10/0013147
4 %
5 % Rotina de quantização
6 %     Quantiza amplitudes e coef. de Lipschitz para 4, 8 e 12 bits
7 %     Usa a função quant
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 %%% Amplitudes nos pontos de máximos/mínimos/singularidades
11 n = max(A_s);
12 A_s_norm = A_s/n;
13 A_s4 = n*quant(A_s_norm,4);
14 A_s8 = n*quant(A_s_norm,8);
15 A_s12 = n*quant(A_s_norm,12);
16
17 %%% Amplitudes nos pontos de inflexão
18 n = max(A_i);
19 A_i_norm = A_i/n;
20 A_i4 = n*quant(A_i_norm,4);
21 A_i8 = n*quant(A_i_norm,8);
22 A_i12 = n*quant(A_i_norm,12);
23
24 %%% Coeficientes de Lipschitz
25 n = max(am_lpz);
26 am_lpz_norm = am_lpz/n;
27 am_lpz4 = n*quant(am_lpz_norm,4);
28 am_lpz8 = n*quant(am_lpz_norm,8);
29 am_lpz12 = n*quant(am_lpz_norm,12);

```

I.4 DAC

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % UnB - FT - EnE
3 % Isadora F. Martins - 10/0013147

```

```

4 %
5 % Finalização do DAC.
6 %     Usa as funções de reconstrução e cálculo de erro.
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8
9
10
11 %% Quantizando amplitudes e lipschitz:
12
13 %%% Sem quantização
14 result = reconstruct(num_pol,A_i,A_s,am_lpz,tempo,delta,tau);
15 [erro_abs,erro_rel,rms] = erros_calc(result,sig);
16
17 %%% Quantização com 4 bits
18 resultq4 = reconstruct(num_pol,A_i4,A_s4,am_lpz4,tempo,delta,tau);
19 [erro_abs4,erro_rel4,rms4] = erros_calc(resultq4,sig);
20
21 %%% Quantização com 8 bits
22 resultq8 = reconstruct(num_pol,A_i8,A_s8,am_lpz8,tempo,delta,tau);
23 [erro_abs8,erro_rel8,rms8] = erros_calc(resultq8,sig);
24
25 %%% Quantização com 12 bits
26 resultq12 = reconstruct(num_pol,A_i12,A_s12,am_lpz12,tempo,delta,tau);
27 [erro_abs12,erro_rel12,rms12] = erros_calc(resultq12,sig);
28
29
30
31 %% Quantizando só amplitudes:
32
33 %%% Sem quantização
34 result = reconstruct(num_pol,A_i,A_s,am_lpz,tempo,delta,tau);
35 [erro_abs,erro_rel,rms] = erros_calc(result,sig);
36
37 %%% Quantização com 4 bits
38 resultq4 = reconstruct(num_pol,A_i4,A_s4,am_lpz,tempo,delta,tau);
39 [erro_abs4,erro_rel4,rms4] = erros_calc(resultq4,sig);
40
41 %%% Quantização com 8 bits
42 resultq8 = reconstruct(num_pol,A_i8,A_s8,am_lpz,tempo,delta,tau);
43 [erro_abs8,erro_rel8,rms8] = erros_calc(resultq8,sig);
44
45 %%% Quantização com 12 bits
46 resultq12 = reconstruct(num_pol,A_i12,A_s12,am_lpz,tempo,delta,tau);
47 [erro_abs12,erro_rel12,rms12] = erros_calc(resultq12,sig);
48
49
50
51 %% Quantizando só lipschitz:
52
53 %%% Sem quantização
54 result = reconstruct(num_pol,A_i,A_s,am_lpz,tempo,delta,tau);
55 [erro_abs,erro_rel,rms] = erros_calc(result,sig);

```

```

56
57 %% Quantização com 4 bits
58 resultq4 = reconstruct(num_pol,A_i,A_s,am_lpz4,tempos,delta,tau);
59 [erro_abs4,erro_rel4,rms4] = erros_calc(resultq4,sig);
60
61 %% Quantização com 8 bits
62 resultq8 = reconstruct(num_pol,A_i,A_s,am_lpz8,tempos,delta,tau);
63 [erro_abs8,erro_rel8,rms8] = erros_calc(resultq8,sig);
64
65 %% Quantização com 12 bits
66 resultq12 = reconstruct(num_pol,A_i,A_s,am_lpz12,tempos,delta,tau);
67 [erro_abs12,erro_rel12,rms12] = erros_calc(resultq12,sig);
68
69
70
71 %% Gráficos do DAC
72
73 % Resultado da reconstrução nos quatro casos
74 figure
75
76 subplot(2,2,1)
77 plot(t,sig,'b-',t,result,'r--')
78 title('Reconstrução sem quantização','FontSize',18)
79 legend('Sinal original','Sinal reconstruído')
80 xlabel('Tempo(s)','FontSize',18)
81 set(gca,'FontSize',16)
82 grid on
83
84 subplot(2,2,2)
85 plot(t,sig,'b-',t,resultq4,'r--')
86 title('Reconstrução com quantização de 4 bits','FontSize',18)
87 legend('Sinal original','Sinal reconstruído')
88 xlabel('Tempo(s)','FontSize',18)
89 set(gca,'FontSize',16)
90 grid on
91
92 subplot(2,2,3)
93 plot(t,sig,'b-',t,resultq8,'r--')
94 title('Reconstrução com quantização de 8 bits','FontSize',18)
95 legend('Sinal original','Sinal reconstruído')
96 xlabel('Tempo(s)','FontSize',18)
97 set(gca,'FontSize',16)
98 grid on
99
100 subplot(2,2,4)
101 plot(t,sig,'b-',t,resultq12,'r--')
102 title('Reconstrução com quantização de 12 bits','FontSize',18)
103 legend('Sinal original','Sinal reconstruído')
104 xlabel('Tempo(s)','FontSize',18)
105 set(gca,'FontSize',16)
106 grid on
107

```

```
108
109 figure
110
111 plot(t,erro_rel,'b-',t,erro_rel4,'c--',t,erro_rel8,'m-.',t,erro_rel12,'r:')
112 title('Erro relativo de reconstrução (%)','FontSize',18)
113 legend('reconstrução sem quantização','quantizado com 4 bits',...
114        'quantizado com 8 bits','quantizado com 12 bits')
115 xlabel('Tempo(s)','FontSize',18)
116 set(gca,'FontSize',16)
117 grid on
```