

Dissertação de Mestrado

**A NOVEL WAVELET-BASED  
ANALOG-TO-DIGITAL CONVERTER**

**Isadora Freire Martins**

**Brasília, Novembro de 2017**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA**

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

Dissertação de Mestrado

**A NOVEL WAVELET-BASED  
ANALOG-TO-DIGITAL CONVERTER**

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

**Isadora Freire Martins**

Banca Examinadora

Prof. Sandro A. P. Haddad, PGEA/UnB

*Orientador*

\_\_\_\_\_

Prof. Wellington Avelino do Amaral, FGA/UnB

*Examinador externo*

\_\_\_\_\_

Prof. Raimundo Carlos S. Freire, DEE/UFCG

*Examinador externo*

\_\_\_\_\_

Prof. José Camargo da Costa, PGEA/UnB

*Suplente*

\_\_\_\_\_

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

**A NOVEL WAVELET-BASED ANALOG-TO-DIGITAL CONVERTER**

**ISADORA FREIRE MARTINS**

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:



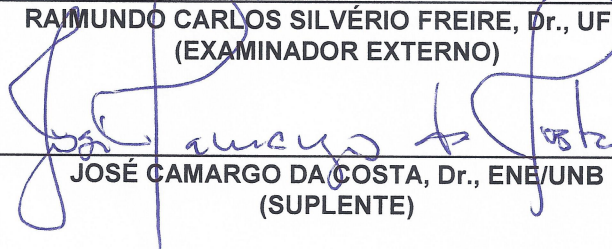
---

SANDRO AUGUSTO PAVLIN HADDAD, Dr., FGA/UNB  
(ORIENTADOR)



---

RAIMUNDO CARLOS SILVÉRIO FREIRE, Dr., UFCG  
(EXAMINADOR EXTERNO)



---

JOSÉ CAMARGO DA COSTA, Dr., ENE/UNB  
(SUPLENTE)

Brasília, 10 de novembro de 2017.

## FICHA CATALOGRÁFICA

MARTINS, ISADORA FREIRE

A NOVEL WAVELET-BASED ANALOG-TO-DIGITAL CONVERTER [Distrito Federal] 2017.  
xvi, 96 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2017).

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

Departamento de Engenharia Elétrica

1. Analog-to-digital converter

2. Wavelet transform

3. Lipschitz exponent

4. Asynchronous sampling

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

MARTINS, I.F. (2017). *A NOVEL WAVELET-BASED ANALOG-TO-DIGITAL CONVERTER*.  
Dissertação de Mestrado em Engenharia de Sistemas Eletrônicos e de Automação, Publicação  
PGEA.DM - 679/2017, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília,  
DF, 96 p.

## CESSÃO DE DIREITOS

AUTOR: Isadora Freire Martins

TÍTULO: A NOVEL WAVELET-BASED ANALOG-TO-DIGITAL CONVERTER.

GRAU: Mestre em Engenharia de Sistemas Eletrônicos e Automação ANO: 2017

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

*Isadora Freire Martins*

Isadora Freire Martins

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

## **Dedicatória**

*Dedico este trabalho à minha família; em especial, à minha mãe.*

*Isadora Freire Martins*

## Agradecimentos

*O primeiro agradecimento não poderia ser direcionado a outra pessoa que não meu orientador, Professor Sandro Haddad. Agradeço por ter me aceitado como aluna e pela orientação durante os últimos dois anos e meio; pela paciência (especialmente com meu quase não cumprimento de prazos um tanto quanto frequente na reta final); pelos conselhos; pelo respeito; e pela competência e seriedade ao exercer seu papel de professor. Nessa mesma linha, o segundo agradecimento só pode ser feito ao Professor José Edil de Medeiros, que me aceitou como aluna durante a graduação e se propôs a ser meu co-orientador durante o mestrado, de livre e espontânea vontade, e ainda à distância. Obrigada pela ajuda, pelos conselhos e pelo tempo dedicado a esta pesquisa. Tive a sorte de trabalhar com um orientador e um co-orientador igualmente competentes e engajados, mas de opiniões frequentemente divergentes, permitindo-me absorver o melhor da formalidade de um e da criatividade do outro. Acredito que são pessoas como os Professores Sandro e Edil que fazem a diferença pela qualidade da educação superior no país.*

*Também gostaria de agradecer aos membros da banca examinadora, que gentilmente aceitaram os convites de participação e dedicaram parte de seu tempo para avaliar meu trabalho, assistir à minha apresentação, dividir comigo suas opiniões, comentários e sugestões de melhorias, além de se terem posto à disposição para dúvidas e discussões. Muito obrigada, Professor José Camargo da Costa, Professor Raimundo Freire, e Professor Wellington Amaral. Também agradeço ao Professor Camargo por se colocar sempre à disposição para ajudar seus alunos a encontrarem novas oportunidades.*

*Agradeço aos colegas do LPCI e da UnB: Professor Daniel Café, que com dedicação e paciência colocou o laboratório em ordem; Yuri, que sempre está pronto para dar uma orientação e dividir ideias novas; Guilherme e José Alberto, parceiros de pesquisa; e aos colegas Reinaldo, Shimabuko, Gabriela, Iago, Piá e Dário pelos momentos de convivência e pelas agradáveis pausas para conversas.*

*Agradeço aos funcionários do SG-11, em especial Sidney e Valter; aos funcionários da FGA, em especial à Fernanda; e aos funcionários do ENE.*

*Por fim, agradeço novamente à minha família e também a meus amigos, que me ajudaram a recuperar pontos de sanidade perdidos durante a jornada (quase!) heroica em busca do grau de Mestre. Em tempo, 50 pontos ~~para a Grifinória~~ para o amigo Marcos Fonseca, que se dispôs a acordar cedo e a passar uma manhã inteira assistindo à minha defesa.*

*Agradeço à CAPES pela bolsa durante o período regular do mestrado e à FAP-DF pelo auxílio financeiro para apresentar meu artigo no ISCAS 2017.*

*Meu muito obrigada a todos.*

*Isadora Freire Martins*

---

## ABSTRACT

This manuscript presents the project of an analog-to-digital converter with a wavelet-based sampling scheme. Instead of sampling a signal with uniformly spaced samples and in a frequency limited by Nyquist's criteria, the proposed ADC represents an input signal based on its characteristics—specifically, the critical points localization and the estimation of the signal's morphology around these points. The first part of this work contains the system-level development, where the sampling algorithm is proposed as well as a polynomial reconstruction algorithm. Tests are run for different resolutions and wavelet bases and scales. The results show that the system successfully localizes the critical points and estimates the morphology of the signal, with high correlation and low RMS error values observed between the reconstructed signal and the input. The second part of this work contains the circuit-level development, where the wavelet transform is implemented with analog wavelet filters. The transfer functions of these filters are obtained by applying two different approximation methods. The results across scales show the critical points' localization.

---

## RESUMO

Nesta dissertação, é proposto um conversor analógico-digital cujo processo de amostragem é baseado em propriedades da transformada wavelet. Tais propriedades permitem identificar características de interesse do sinal—especificamente, a localização de seus pontos críticos e a descrição da morfologia nos trechos entre esses pontos—, e assim representá-lo, em vez de aplicar a amostragem uniforme e limitada pelo critério de Nyquist. A primeira parte deste trabalho apresenta a implementação do conversor em nível de sistema para diferentes resoluções e bases e escalas da transformada wavelet. Para validar o algoritmo de amostragem, é proposto também um algoritmo de reconstrução polinomial do sinal. Os resultados mostram que a identificação de pontos críticos e a estimativa da morfologia do sinal são realizadas com sucesso, tendo sido possível recuperar o sinal de entrada com alta correlação e baixo erro RMS entre os sinais original e reconstruído. A segunda parte deste texto apresenta o desenvolvimento em nível de circuito. A transformada wavelet é implementada por filtros wavelet analógicos, que são testados utilizando-se duas aproximações diferentes para sua resposta em frequência. Os resultados de simulações para variadas escalas permitem identificar os pontos críticos do sinal.



# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b> .....	<b>1</b>
1.1	CONTEXT: ANALOG-TO-DIGITAL CONVERSION .....	1
1.2	OUR PROPOSAL .....	6
1.2.1	OBJECTIVES .....	7
1.3	OUTLINE .....	8
<b>2</b>	<b>THEORETICAL BACKGROUND</b> .....	<b>9</b>
2.1	THE WAVELET TRANSFORM .....	9
2.1.1	LIPSCHITZ COEFFICIENT .....	14
2.2	ANALOG WT FILTERS .....	17
2.2.1	PADÉ APPROXIMATION .....	17
2.2.2	$L_2$ APPROXIMATION .....	19
2.2.3	GM-C FILTERS .....	20
2.3	SUMMARY OF THE CHAPTER .....	23
<b>3</b>	<b>SYSTEM-LEVEL DEVELOPMENT</b> .....	<b>25</b>
3.1	SAMPLING ALGORITHM .....	25
3.2	RECONSTRUCTION ALGORITHM .....	26
3.3	METHODOLOGY .....	27
3.3.1	THE TEST SIGNAL .....	27
3.3.2	WT PARAMETERS .....	28
3.3.3	ERROR METRICS .....	29
3.3.4	QUANTIZATION .....	30
3.4	SYSTEM RESULTS .....	32
3.4.1	COMPARISON WITH A STANDARD ADC .....	44
3.5	SUMMARY OF THE CHAPTER .....	46
<b>4</b>	<b>CIRCUIT-LEVEL DEVELOPMENT</b> .....	<b>47</b>
4.1	CIRCUIT BLOCK DIAGRAM .....	47
4.2	METHODOLOGY .....	48
4.3	RESULTS .....	49
4.3.1	ANALOG WAVELET FILTER BLOCK .....	49

4.3.2	LIPSCHITZ EXPONENT ESTIMATION.....	63
4.4	DISCUSSION.....	67
<b>5</b>	<b>CONCLUSION.....</b>	<b>71</b>
5.1	FUTURE WORKS .....	72
	<b>REFERENCES .....</b>	<b>73</b>
	<b>APPENDICES .....</b>	<b>76</b>
<b>I</b>	<b>RESUMO ESTENDIDO EM PORTUGUÊS .....</b>	<b>77</b>
<b>II</b>	<b>MATLAB SCRIPTS.....</b>	<b>83</b>
II.1	SAMPLING ALGORITHM.....	83
II.1.1	QUANTIZATION FUNCTION.....	86
II.1.2	LIPSCHITZ EXPONENT ESTIMATION FUNCTION .....	87
II.1.3	FUNCTION TO GENERATE TEST SIGNAL .....	88
II.2	RECONSTRUCTION ALGORITHM .....	88
II.3	RECONSTRUCTION TESTS .....	89
II.3.1	ERROR METRICS .....	92
II.4	GAUSSIAN FILTERS .....	94

# LIST OF FIGURES

1.1	Conventional ADC (adapted from de la Rosa [1]).	2
1.2	Anti-aliasing filter (adapted from de la Rosa [1]).	2
1.3	Ideal quantizer characteristics [2].	3
1.4	Level-crossing ADC (adapted from Tsvividis [3]).	5
1.5	Information acquired to represent a signal based on (a) its critical points and (b) its morphology.	7
2.1	The WT compares the signal to the left to versions of the wavelet basis in various scales and positions, shown to the right [4].	10
2.2	Some wavelet bases: (a) gaus1 (b) gaus2 (c) Morlet (d) Daubechies (db6). [5]	11
2.3	In small scales (left) the base is compressed and the details of the signal can be analyzed. For larger scales (right), the base is expanded and slow variations of the signal become more evident. [4]	12
2.4	The convolution smooths the original signal. The first-order transform, $W_1f(u, s)$ , is computed with $\psi = -\theta'$ , while the second-order transform, $W_2f(u, s)$ , is computed with $\psi = \theta''$ . [6]	14
2.5	Function from Eq. (2.20) for different values of $\alpha$ [5].	16
2.6	Gm-C integrator [5].	21
2.7	System block diagram for state-space representation [5].	21
2.8	Filter implementation scheme with state-space matrices [5].	22
2.9	Gm-C filter implementation of a 6 <sup>th</sup> order filter with orthonormal ladder structure (adapted from Karel et al. [7]).	23
3.1	Wavelet-based ADC sampling algorithm.	26
3.2	Identified critical points of a test signal: 'x' represents a local maximum or minimum, and '+' represents an inflection point. 'P <sub>n</sub> ' are the polynomials that reconstruct the signal piecewise.	27
3.3	Test signal.	28
3.4	Gaussian wavelet bases [8].	31
3.5	(a) Daubechies and biorthogonal bases with $n = 1$ vanishing moment. Bases with $n = 2$ vanishing moments: (b) Daubechies (c) Biorthogonal (d) Reverse biorthogonal [8].	31
3.6	Critical points detection: local maxima and minima.	32

3.7	Critical points detection: inflection points. ....	33
3.8	Coefficients lines for second order wavelet transform at scales 16, 32, 48 and 64. ....	35
3.9	Lipschitz exponent estimation. ....	36
3.10	Sampled signals. ....	37
3.11	Quantized sampled signals for 4, 8, and 12-bit resolutions. ....	38
3.12	Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for gaus1 and gaus2 wavelet bases, set of scales from 16 to 64, and ideal (not quantized) values. ....	39
3.13	Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for gaus1 and gaus2 wavelet bases, set of scales from 16 to 64, and quantized values. ....	40
3.14	Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for db1 and db2 wavelet bases, set of scales from 1 to 64, and ideal (not quantized) values. ....	44
3.15	Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for db1 and db2 wavelet bases, set of scales from 1 to 64, and quantized values. ....	45
3.16	Number of bits x RMS error x sampling rate. ....	46
4.1	Wavelet-based ADC block diagram. ....	47
4.2	First derivative of a gaussian (gaus1) for sixth order $L_2$ approximation for (a) original transfer function $H_{gaus1}(s)$ (b) approximated transfer function $H'_{gaus1}(s)$ . ....	50
4.3	Second derivative of a gaussian (gaus2) for sixth order $L_2$ approximation for (a) original transfer function $H_{gaus2}(s)$ (b) approximated transfer function $H'_{gaus2}(s)$ . ....	51
4.4	Circuit implementation of gaussian wavelet filters. ....	53
4.5	Testbench schematic for WT-1 and WT-2 gaussian filters. ....	53
4.6	Impulse responses for gaus1. ....	54
4.7	Impulse responses for gaus2. ....	55
4.8	First-order wavelet transform of the test signal using analog wavelet filters with gaussian basis. ....	56
4.9	Second-order wavelet transform of the test signal using analog wavelet filters with gaussian basis. ....	58
4.10	First (top) and second (bottom) derivatives of the gaussian function for 10-th order filter. ....	60
4.11	Impulse response for gaus1 for 10-th order filter. ....	61
4.12	Impulse response for gaus2 for 10-th order filter. ....	62
4.13	First-order wavelet transform for the test signal and gaus1 wavelet basis in selected scales, using a 10 <sup>th</sup> order Padé approximation. ....	64

4.14	Second-order wavelet transform for the test signal and gaus1 wavelet basis in selected scales, using a 10 <sup>th</sup> order Padé approximation.....	65
I.1	Informação amostrada do sinal de entrada pelo ADC proposto.....	78
I.2	Algoritmo de amostragem. ....	80
I.3	Diagrama de blocos do circuito proposto para o conversor A/D.....	81
I.4	Sinal de entrada (vermelho) e sua reconstrução (amarelo, linha tracejada).....	82

# LIST OF TABLES

2.1	Taylor coefficients for expansions in the Laplace domain ( $k = 16$ ) of gaussian functions (part 1). .....	19
2.2	Taylor coefficients for expansions in the Laplace domain ( $k = 16$ ) of gaussian functions (part 2). .....	19
2.3	Padé coefficients for $Q(s)$ ( $n = 10$ ) of gaussian functions. ....	19
2.4	Padé coefficients for $P(s)$ ( $m = 6$ ) of gaussian functions. ....	20
3.1	Chosen wavelet bases .....	28
3.2	Polynomial reconstruction parameters for gaus1, gaus2, scales from 16 to 64 (ideal case) .....	39
3.3	Correlation coefficient (%) for scales from 16 to 64 .....	41
3.4	RMS error (%) for scales from 16 to 64 .....	41
3.5	Correlation coefficient (%) for scales from 1 to 64 .....	42
3.6	RMS error (%) for scales from 1 to 64 .....	42
4.1	Extrema of the WT2 coefficients lines across scales for the analog filter with sixth order transfer function approximation. ....	66
4.2	Lipschitz exponents obtained with manual calculations for the analog filter with sixth order transfer function approximation. ....	66
4.3	Extrema of the WT2 coefficients lines across scales for the analog filter with tenth order transfer function approximation. ....	67
4.4	Lipschitz exponents obtained with manual calculations for the analog filter with tenth order transfer function approximation. ....	67

# LIST OF SYMBOLS

## Greek Symbols

$\alpha$  Lipschitz Coefficient

## Superscripts

\*

Complex conjugate

## Abbreviations

AAF	Anti-Aliasing Filter
A/D	Analog-to-Digital
ADC	A/D Converter
CS	Compressive Sensing
CTADC	Continuous-Time Analog-to-Digital Converter
DAC	Digital-to-Analog Converter
LMSE	Least-Mean-Square-Error
LSB	Least Significant Bit
MRI	Magnetic Resonance Imaging
NUS	Non-Uniform Sampler
RMPI	Random-Modulation Pre-Integrator
S/H	Sample-and-Hold
sps	Samples per second
WNN	Wavelet Neural Network
WT	Wavelet Transform

# Chapter 1

## Introduction

In this chapter we present the context and motivation for the design of a wavelet-based analog-to-digital converter, describing the main characteristics and objectives of the project. In addition, this manuscripts' outline is described.

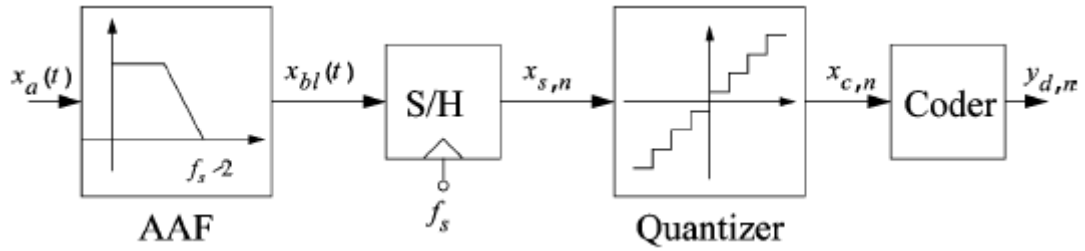
### 1.1 Context: analog-to-digital conversion

Analog-to-digital conversion is fundamental to modern signal processing, given the need to represent signals from our analog world in the digital domain. In order to complete this conversion, the analog signal needs to be sampled. One of the most important theorems that rule the way in which this sampling is done is the Nyquist-Shannon Sampling Theorem [9, 10]: to assure that a signal can be recovered after sampling, it must be sampled at a rate equal to at least twice its bandwidth. Many of the converters today, then, sample at or slightly above Nyquist's rate (the so-called Nyquist converters), or at a rate way greater than Nyquist's (a practice known as oversampling).

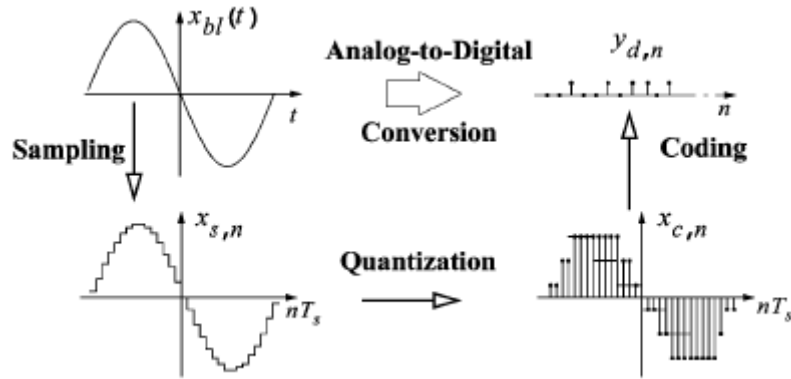
Figure 1.1(a) shows a typical analog-to-digital converter block diagram, composed of an anti-aliasing filter (AAF); a sampler, implemented with a sample-and-hold (S/H) block; a quantizer; and an encoder. The need for the AAF is one of the consequences of following Nyquist's criterion: if the sampling rate is based on the signal's bandwidth, then the input signal must be band-limited. Also, a Nyquist converter must have a very well designed AAF to account for the sharp edge required for sampling at the Nyquist rate, as is shown in Figure 1.2(a), whereas an oversampled converter can apply an AAF with a smooth edge, as shown in figure 1.2(b) [1].

The next block is the sampling block. A sampled-and-held signal is illustrated in Figure 1.1(b): the input signal is continuous in time and amplitude, while the sampled signal is discrete in time, but still continuous in amplitude. The typical ADC applies uniform sampling techniques, which means that the sampling times are uniformly-distributed according to the sampling rate.



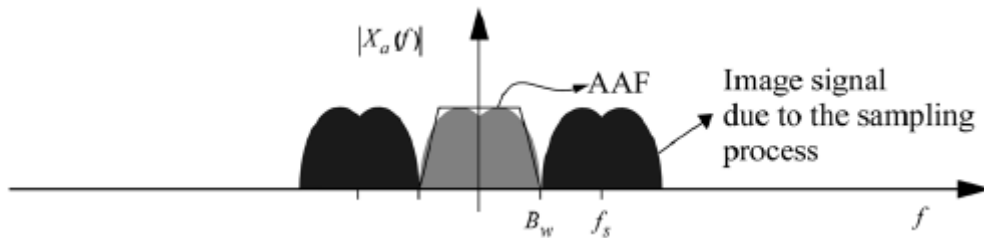


(a) Typical ADC block diagram

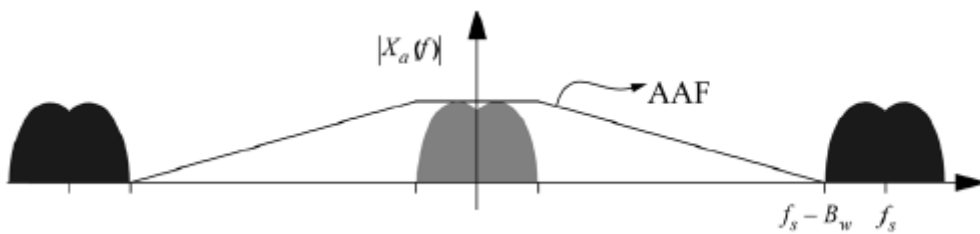


(b) Steps of analog-to-digital conversion

Figure 1.1: Conventional ADC (adapted from de la Rosa [1]).



(a) AAF for Nyquist-rate converter



(b) AAF for oversampled converter

Figure 1.2: Anti-aliasing filter (adapted from de la Rosa [1]).

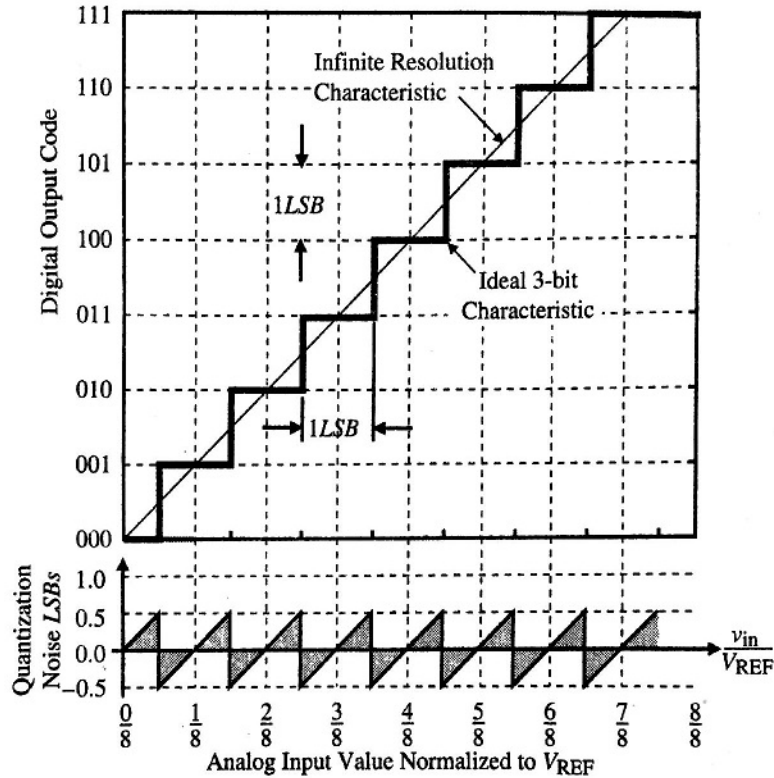


Figure 1.3: Ideal quantizer characteristics [2].

The third block, the quantizer, discretizes the amplitude of the signal. Unlike the sampling process, quantization is irreversible and always results in some permanent loss of information. Figure 1.3 shows the characteristics of an ideal uniform mid-tread 3-bit quantizer. The quantizer divides the reference tension,  $V_{REF}$ , in intervals, and allocates the samples to the closest quantization levels. A binary  $N$  bit code is set to each of the  $2^N$  quantization levels. The quantization error is also shown in Figure 1.3, and is limited to  $\pm 0.5$  LSB (Least Significant Bit). The quantized signal is then represented as a strand of bits, in the process named encoding.

The processes described above underly the functioning of almost all ADCs in consumer appliances known today. However, they do have their flaws. First of all, the design of ADCs presents trade-offs between conversion speed, conversion resolution, and power consumption, among other features. For example, flash ADCs are known to be high-speed converters, but with low resolution and also high power consumption, while oversampled ADCs, such as  $\Sigma\Delta$  ADCs, have high resolution and low conversion speed. Another possible trade-off is the one between bandwidth and resolution: in order to keep the same resolution, a Nyquist-rate converter (less bandwidth) must have more amplitude quantization levels than an oversampled one [11]. Another important point is that the Shannon-Nyquist theorem only sets a sufficient condition for reconstruction—and not a necessary condition. The choice of a rate that follows the Nyquist-Shannon criteria comes with disadvantages

other than the need to limit the bandwidth of the signal; for instance, a signal with time varying frequency has its sampling rate determined based on its highest frequency component, which implies that the lower frequency portions of the signal will be unnecessarily oversampled.

The problem of developing a more efficient sampling techniques arises. Many solutions to this problem and alternative ways to develop sampling and sensing of signals are investigated in the field of compressive sensing (CS). In an overly simplified manner, CS applies different transforms and/or bases in order to reach a sparse representation of a signal of interest. The use of a basis for which this signal is sparse and densely represented allows a sub-Nyquist sampling of the signal, along with its compression, since the useful information is comprised in a small amount of data. Undersampling can be useful when there are project constraints, such as a limited number of sensors; or when the financial cost of a measurement is very high, as in imaging processes via neutron scattering; or when the sensing process is slow, as in MRI scans [12]; or for low-power consumption applications, such as biomedical devices.

CS methods mostly apply computational and mathematical tools to the processing of discrete signals. The implementation of a CS device in integrated circuits to exceed the performance of a state-of-the-art ADC for high-bandwidth signals was the object of the “Analog-to-Information Project” [13]. Their results were two prototypes, called NUS (“Non-Uniform Sampler”) and RMPI (“Random Modulation Pre-Integrator”): RMPI is a prototype that digitizes signals at high-bandwidth at a sub-Nyquist rate; while NUS digitizes analog signals with a sparse representation in the frequency domain at Nyquist rate [12, 14, 15].

Another line of research that investigated novel sampling schemes started with Sayiner [16, 17, 11] and the level-crossing ADC, also referred to as a continuous time ADC (CTADC) [18]. Sayiner proposes a high resolution, high speed ADC architecture which outputs amplitude-time ordered pairs that represent samples that are non-uniformly spaced in time. Level-crossing sampling is an intermediate scheme between the Nyquist and the zero-crossing sampling schemes. Zero-crossing sampling schemes are the ones which represent the signal by the instants at which the signal equals a determined value. While the zero-crossing sampling represents a signal with a minimal number of values within a time interval, it also requires very high time resolution, in order to represent with high precision those zero-crossing instants [11]. The level-crossing sampling, then, consists of sampling the signal whenever it crosses a quantization level, also called a threshold level. Figure 1.4 illustrates this process. The additional information required to recover the input signal can be approximated by interpolation methods [3].

Recently, in 2016, Masry and El-Dib [18] proposed an ADC in which a signal is sampled using the level-crossing ADC approach, followed by a wavelet neural network (WNN) applied as an interpolation method, generating a high-resolution low-power converter. The main reasons to choose the wavelet networks are the known wavelet transform properties and applications, especially its strong compression, time-series prediction and data classifi-

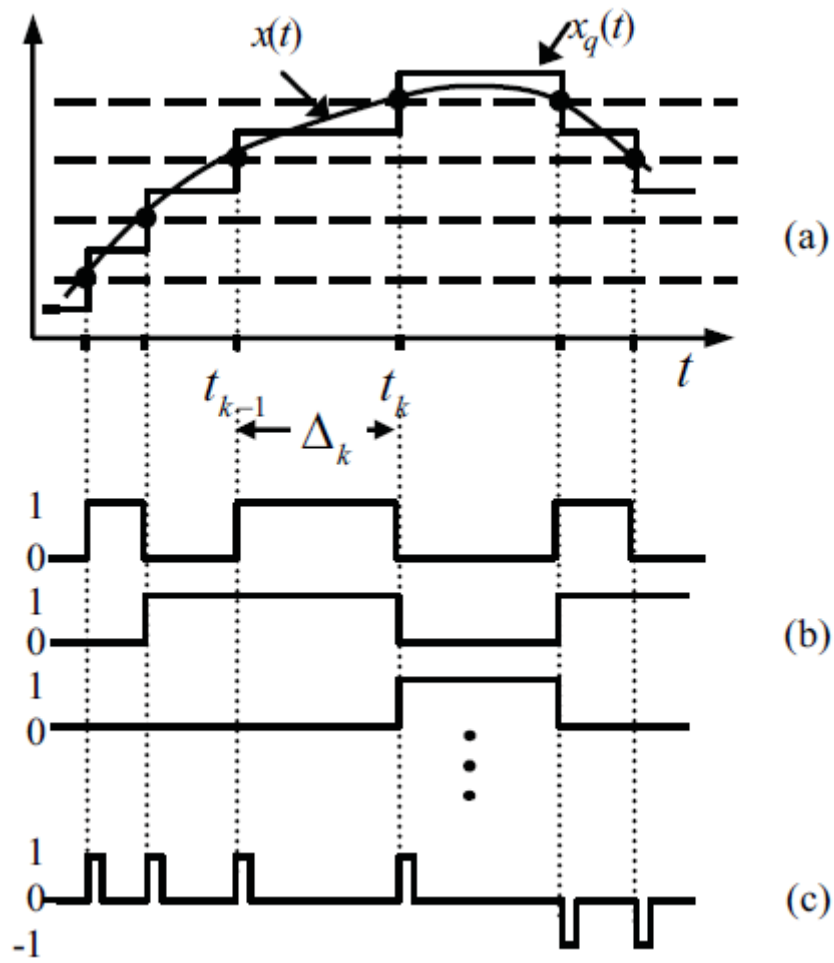


Figure 1.4: Level-crossing ADC (adapted from Tsividis [3]).

cation abilities.

Even though the project presented in this manuscript does not belong to the research fields of CS and level-crossing ADCs, it shares with them some similarities in their objectives of undersampling and signal compression (for compressive sensing) and of asynchronous sampling and low power performance (for CTADCs), as well as the use of the wavelet transform properties, in the specific case of the clockless asynchronous ADC from Masry and El-Dib [18]. This projects' proposed ADC is the object of the next section.

## 1.2 Our proposal

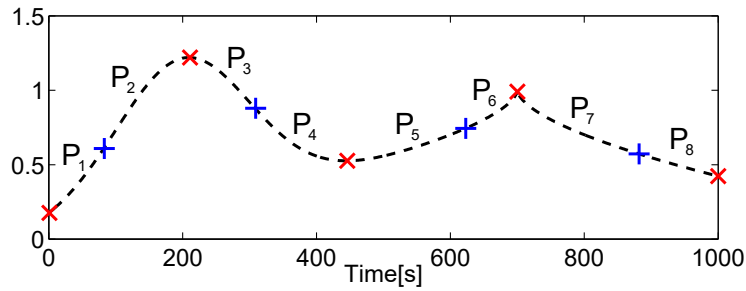
This project proposes a novel wavelet-based sampling technique that aims to reduce the power consumption of an ADC by selecting which points to sample, instead of using the traditional uniform sampling method.

The traditional sampling block of an ADC has been discussed. Another important aspect is how to recover a signal from its samples. After a signal is sampled, it is quantized and converted to digital form, with codes that represent each sample. Even if we consider that the recovery of all the samples from the digital code is lossless, we need to interpolate this data in order to reconstruct the input signal from the samples. Usually, we use a constant function to interpolate these samples. The last step is to smooth this curve, what can be achieved with a low-pass filter.

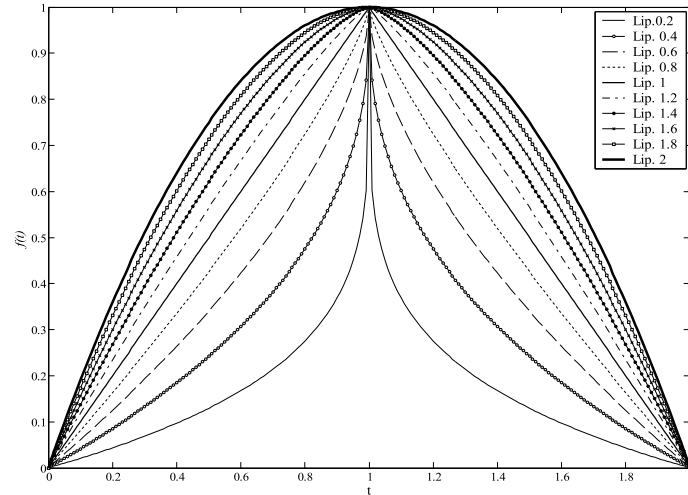
During sampling, we not only wish to convert the signal from analog to digital form, but we also choose how to represent the signal, or how to store the information it carries. In the traditional method, we choose samples in a uniform and periodic manner, regardless of the signal's characteristics or properties. CS changes the way a signal is sampled with no modifications to the way it is reconstructed, since the signal is represented in another domain, where less samples are needed to store the same information as in other domains. This projects' proposal is:

1. when sampling the signal: to choose only samples that carry relevant and specific information about the signal;
2. when recovering the signal: to use another interpolation function instead of the constant/step one traditionally applied.

One way to analyze a signal and its behaviour is to localize its critical points. This is our approach: the special points selected during sampling are the signal's local maxima, minima, and inflection points. In this way, the proposed sampling algorithm is event-based. Fig. 1.5(a) shows an example of the critical points identification in the test signal, where the 'x's correspond to the local maximum and minimum points, and the '+'s correspond to the inflection points. The interpolation function is based on the signal's morphology, which



(a) Critical points identification.



(b) Signal whose morphology varies with the Lipschitz exponent.

Figure 1.5: Information acquired to represent a signal based on (a) its critical points and (b) its morphology.

can be represented by the Lipschitz exponent extracted at a given point. A function whose concavity varies according to the Lipschitz exponent is shown in Fig. 1.5(b). This function is the basis for a polynomial reconstruction algorithm that is also proposed and allows the validation of this sampling method: the signal is reconstructed piecewise, with polynomials described by the Lipschitz coefficient and limited by two consecutive critical points. These polynomials are also indicated in Fig. 1.5(a). Both the critical points localization and the Lipschitz coefficient estimation, which are the outputs of this ADC, are realized by applying the wavelet transform (WT) to the input signal.

### 1.2.1 Objectives

The objective of this project is to evaluate and validate a novel wavelet-based sampling scheme for ADCs both in system- and circuit-level. Even though no specific application is taken into account in the scope of this project, we expect that it must be suitable for low-power applications because of its asynchronous and compressive characteristics.

The algorithm for a wavelet-based analog-to-digital converter was initially proposed in

a previous work [8]. The first specific objective of this work is to give continuity to that research, by improving possible flaws in the algorithm and expanding its scope to circuit-level analysis.

### **1.3 Outline**

Chapter 2 contains a review of the wavelet transform theory that was applied in the development of this project. Chapter 3 shows the system-level implementation of the proposed ADC and the polynomial reconstruction method. Chapter 4 contains the circuit development and a discussion of the results, and Chapter 5 concludes this manuscript.

# Chapter 2

## Theoretical Background

This chapter contains an overview of the theory that was necessary to develop the wavelet-based analog-to-digital converter project. The first section contains the theoretical framework regarding the wavelet transform and the Lipschitz exponent, and the last section explains the methods chosen to implement the wavelet transform in circuit level.

### 2.1 The Wavelet Transform

A transform can be interpreted as an operation which represents a signal from one domain in another domain. Mathematically, it is the result of the convolution of a basis with the function that represents the analyzed signal. The convolution integral of a linear, shift-invariant system (LSIS) with input signal  $x(t)$  and output signal  $y(t)$  is:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau \quad (2.1)$$

where  $h(t)$  is the system's impulse response.

Then, the Fourier transform of a function  $f(t)$  is expressed by the convolution between the Fourier basis,  $e^{j\omega t}$ , and the function  $f(t)$ :

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{j\omega t}dt \quad (2.2)$$

The wavelet transform of an input signal  $f(t)$  is the convolution of this signal with a wavelet basis  $\psi(t)$ , and is expressed by Equation 2.3, where  $a > 0$ . The superscript  $*$  indicates complex conjugate, and can be removed from the equation when the wavelet basis is real.



$$W_f(u, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(t) \psi^* \left( \frac{t-u}{s} \right) dt \quad (2.3)$$

Equation 2.3 shows that the wavelet transform has two parameters: the scale  $s$  and the position  $u$ . This means that the wavelet transform outputs coefficients that indicate the correlation between the input signal  $f(t)$  and versions of the wavelet basis  $\psi(t)$  shifted in time and also compressed or expanded according to the scale. Figure 2.1 [4] illustrates this interpretation of the wavelet transform.

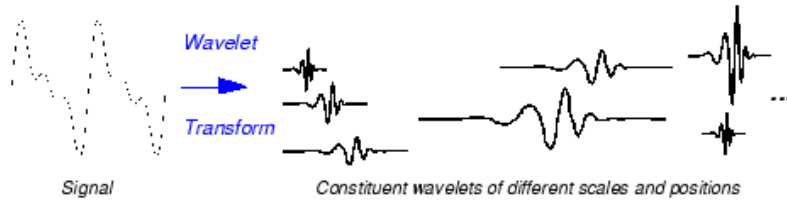


Figure 2.1: The WT compares the signal to the left to versions of the wavelet basis in various scales and positions, shown to the right [4].

A wavelet basis is a “small wave”, a finite and not necessarily symmetrical signal, hence the name “wavelet” transform. There are some requirements for a function to be a wavelet basis:

- It must have finite energy, oscillatory behaviour and zero average, according to Eq. 2.4:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (2.4)$$

- Its Fourier transform must have a zero-frequency component:

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega = C_{\Psi} < \infty \quad (2.5)$$

Eq. 2.5 implies that the wavelets can be interpreted as pass-band filters in the Fourier domain [5].

Some well-known wavelet bases are the first and second derivatives of the gaussian function (“gaus1” and “gaus2”, respectively), the Morlet wavelet and the Daubechies wavelets (“db”). Figure 2.2 [5] shows examples of such bases.

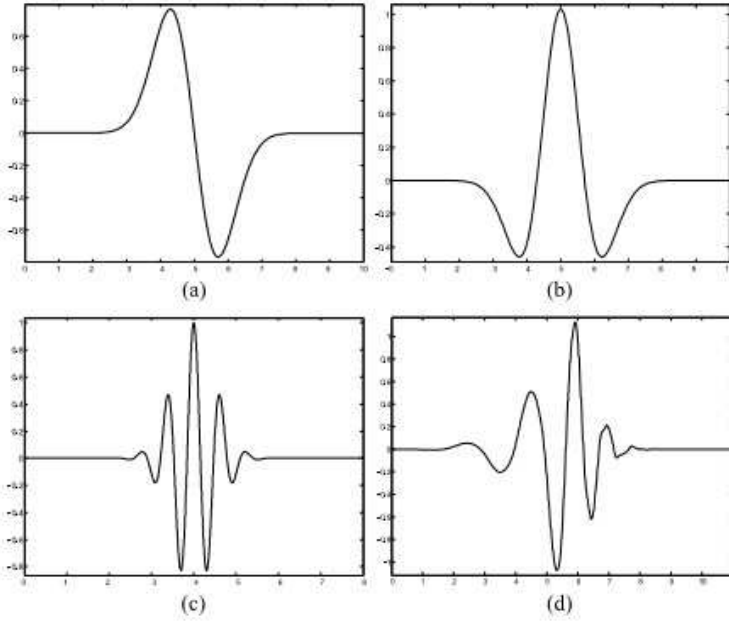


Figure 2.2: Some wavelet bases: (a) gauss1 (b) gauss2 (c) Morlet (d) Daubechies (db6). [5]

While the Fourier transform allows total frequency resolution and no time resolution, the wavelet transform can have varied degrees of time and frequency resolution according to the settings of the scale and position parameters. By altering these parameters, one can evaluate a signal for different time and frequency resolutions and thus describe it [5].

Mallat [6] shows that the time resolution,  $\Delta t$ , and the frequency resolution,  $\Delta \omega$ , are related by Heisenberg's uncertainty principle, which is expressed by Equations 2.6 to 2.8, where  $\Psi(t)$  is the wavelet basis' derivative. In this way, the higher the frequency resolution is, the smaller the time resolution will be. On the other hand, the wavelet transform allows the signal to be analyzed under different conditions, by altering the scale. A small scale corresponds to a compressed wavelet basis, then, the transform will have higher coefficients in positions corresponding to the details and sharp transitions of the signal, that is, its higher frequency components. A large scale corresponds to an expanded wavelet base, which will indicate general characteristics of the signal, slower transitions, the low-frequency components. Figure 2.3 [4] illustrates this property. This means that one way to analyze the signal behaviour in one specific point of interest is to compute the WT of this signal and gradually reduce the scale, focusing more and more on the point of interest. This property is known as the wavelet zoom property.

$$\Delta t \Delta \omega > \frac{1}{2} \quad (2.6)$$

$$\Delta t = \sqrt{\frac{\int t^2 |\psi(t)|^2 dt}{\int |\psi(t)|^2 dt}} \quad (2.7)$$

$$\Delta \omega = \sqrt{\frac{\int \omega^2 |\Psi(\omega)|^2 d\omega}{\int |\Psi(\omega)|^2 d\omega}} \quad (2.8)$$



Figure 2.3: In small scales (left) the base is compressed and the details of the signal can be analyzed. For larger scales (right), the base is expanded and slow variations of the signal become more evident. [4]

In this way, the wavelet zoom can be a powerful tool in characterizing singularities in a signal. In signal processing, singularities and discontinuities often carry relevant information, such as the edges of an image, consonantal sounds in speech, and even sharp transitions in the financial market system [19], so, it must be useful to be able to watch these important points closely.

It can be demonstrated that the signal's regularity is characterized by the reduction of the wavelet transform coefficients across scales. Therefore, the signal's singularities can be determined by the local maxima at different scales. The set of information about these maxima for different scales determines what is called the maxima line. By definition, a modulus maximum is a point  $(t_0, s_0)$  such that  $|Wf(t, s_0)|$  has a local maximum at  $t = t_0$ , which implies [6]:

$$\frac{\partial Wf(t_0, s_0)}{\partial t} = 0 \quad (2.9)$$

The maxima line shows the modulus maxima on the scale-space plane, and this indicates the wavelet transform modulus behaviour at different instants and for different scales.

Another interesting WT property useful for regularity detection are the the vanishing moments. A vanishing moment is a zero moment, and can be expressed by Equation 2.10, for a moment of  $k^{th}$  order.

$$m[k] = \int t^k \psi(t) dt = 0 \quad (2.10)$$

As long as wavelets are concerned, the number of vanishing moments corresponds to the number of zeroes at  $\pi$  for the wavelet representation in the  $Z$  domain. In this case, if a wavelet has  $n$  vanishing moments, it has  $n$  zeroes at  $\pi$  and can be rewritten as [20]:

$$H(z) = \left( \frac{1 + z^{-1}}{2} \right) Q(z) \quad (2.11)$$

Furthermore, Mallat [6] shows that a wavelet with  $n$  vanishing moments can be expressed as the  $n^{\text{th}}$  derivative of a function  $\theta(t)$ , and thus its WT functions as an  $n^{\text{th}}$  order multiscale differential operator. This is established in Theorem 1.

**Theorem 1** *A wavelet  $\psi$  with fast decay has  $n$  vanishing moments if, and only if, there is a function  $\theta$  with fast decay such that*

$$\psi(t) = (-1)^n \frac{d^n \theta(t)}{dt^n} \quad (2.12)$$

*Consequently,*

$$Wf(u, s) = s^n \frac{d^n}{du^n} (f * \bar{\theta}_s)(u) \quad (2.13)$$

$$\bar{\theta}_s = s^{-1/2} \theta(-t/s) \quad (2.14)$$

*Also,  $\psi$  has no more than  $n$  vanishing moments if, and only if,  $\int_{-\infty}^{\infty} \theta(t) dt \neq 0$ .*

Therefore, if the chosen wavelet basis has  $n = 1$  vanishing moment, the WT will function as a first order differential operator, and its modulus maxima shall identify discontinuities; also, the points for which the absolute value of the WT coefficients equal zero will correspond to the signals' maximum and minimum points. Moreover, if the chosen wavelet basis has  $n = 2$  vanishing moments, the WT functions as a second order multiscale differential operator, and its modulus maxima localize maximum curvature points, while the points for which  $|Wf(u, s)| = 0$  identify the inflection points in  $f(t)$ . Figure 2.4 illustrates these properties [6].

Vanishing moments have more properties that depend on the description of a signal's regularity with the Lipschitz coefficient, and thus will be explained in the following subsection.

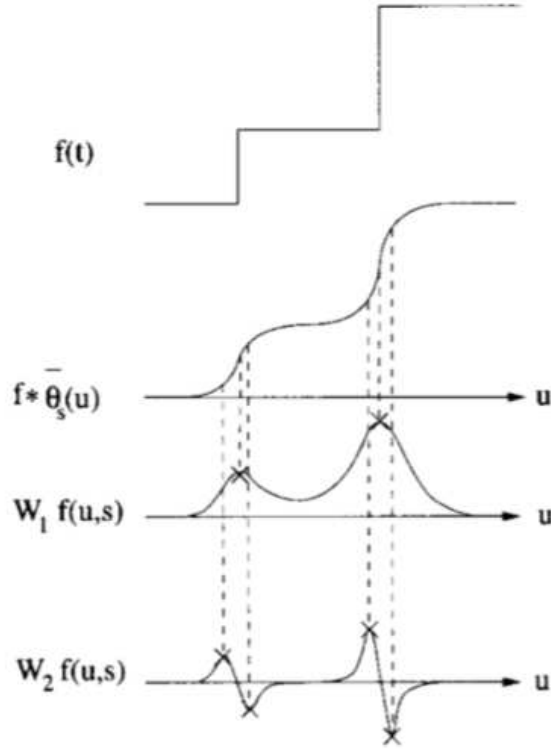


Figure 2.4: The convolution smooths the original signal. The first-order transform,  $W_1 f(u, s)$ , is computed with  $\psi = -\theta'$ , while the second-order transform,  $W_2 f(u, s)$ , is computed with  $\psi = \theta''$ . [6]

### 2.1.1 Lipschitz coefficient

The Lipschitz coefficient, also known as the Hölder coefficient in Mathematics, can characterize the local signal regularity both for a time interval and for one specific instant. If a signal contains a singularity in a point  $\nu$  i.e. is not differentiable at  $\nu$ , this singularity is characterized by the Lipschitz coefficient at  $\nu$ .

**Definition 1** A function  $f$  is Lipschitz  $\alpha \geq 0$  at  $\nu$  if there exist a constant  $K > 0$  and a polynomial  $p_\nu$  of order  $m = \lfloor \alpha \rfloor$  such that,  $\forall t \in \mathbb{R}$ ,

$$|f(t) - p_\nu(t)| \leq K |t - \nu|^\alpha \quad (2.15)$$

The function is uniformly Lipschitz  $\alpha$  over the interval  $[a, b]$  if the Equation 2.15 is satisfied for all  $\nu \in [a, b]$ , with  $K$  independent of  $\nu$ .

The Lipschitz regularity of  $f$  at  $\nu$  or over the interval  $[a, b]$  is the supremum of  $\alpha$  such that  $f$  is Lipschitz  $\alpha$ .

In the above definition, the approximation polynomial is unique for every point  $\nu$  [21]. This polynomial can be the Taylor polynomial. Equation 2.16 shows the expression for the

Taylor expansion at  $\nu$  for an  $m$  times differentiable polynomial,  $m = \lfloor \alpha \rfloor$ .

$$p_\nu = \sum_{k=0}^{m-1} \frac{f^{(k)}(\nu)}{k!} (t - \nu)^k \quad (2.16)$$

The Taylor polynomial approximation error is expressed as:

$$\epsilon_\nu(t) = f(t) - p_\nu(t) \quad (2.17)$$

and must satisfy

$$\forall t \in \mathbb{R}, \quad |\epsilon_\nu(t)| \leq \frac{|t - \nu|^m}{m!} \sup_{u \in [\nu-h, \nu+h]} |f^m(u)| \quad (2.18)$$

In this way, when  $t$  tends towards  $\nu$ ,  $m$  yields a superior bound to the Taylor polynomial approximation error with  $f$  in the vicinity of  $\nu$ . In fact, when Definition 1 is applied with  $p_\nu(t)$  being Taylor's polynomial, this superior bound is a function of the Lipschitz coefficient,  $\alpha$ , and depends on the value of the constant  $K$ :

$$|\epsilon_\nu(t)| = |f(t) - p_\nu(t)| \leq K|t - \nu|^\alpha \quad (2.19)$$

Consider the following example. Let  $f$  be a signal such that

$$f(t) = |1 - t|^\alpha \quad (2.20)$$

Figure 2.5 shows this signal's plot for values of  $\alpha$  between 0.2 e 2. This signal is regular with Lipschitz  $\alpha$  in the intervals  $0 \leq t < 1$  and  $1 < t \leq 2$ , but it contains a singularity at  $t = 1$ : to the left of this singularity,  $f$  increases; to the right of the singularity, it decreases. Then, if we know the position of a singular point, as well as the Lipschitz coefficient value and the signal morphology in each interval, we can describe the signal. In other words, the Lipschitz coefficient comprises the signal's data around a singular point [5].

The Lipschitz regularity of a signal at  $\nu$  depends on the decay at small scales of the absolute value of the wavelet transform amplitude in the neighbourhood of  $\nu$  [6]. The decay of the absolute value of the wavelet transform in the neighbourhood of  $\nu$  is determined by the decay of the modulus maxima in the cone of influence  $|u - \nu| \leq Cs$ .  $f$  is Lipschitz  $\alpha$  if, and only if, there exists a constant  $A > 0$  such that every modulus maximum in the cone of influence satisfies Equation 2.21.

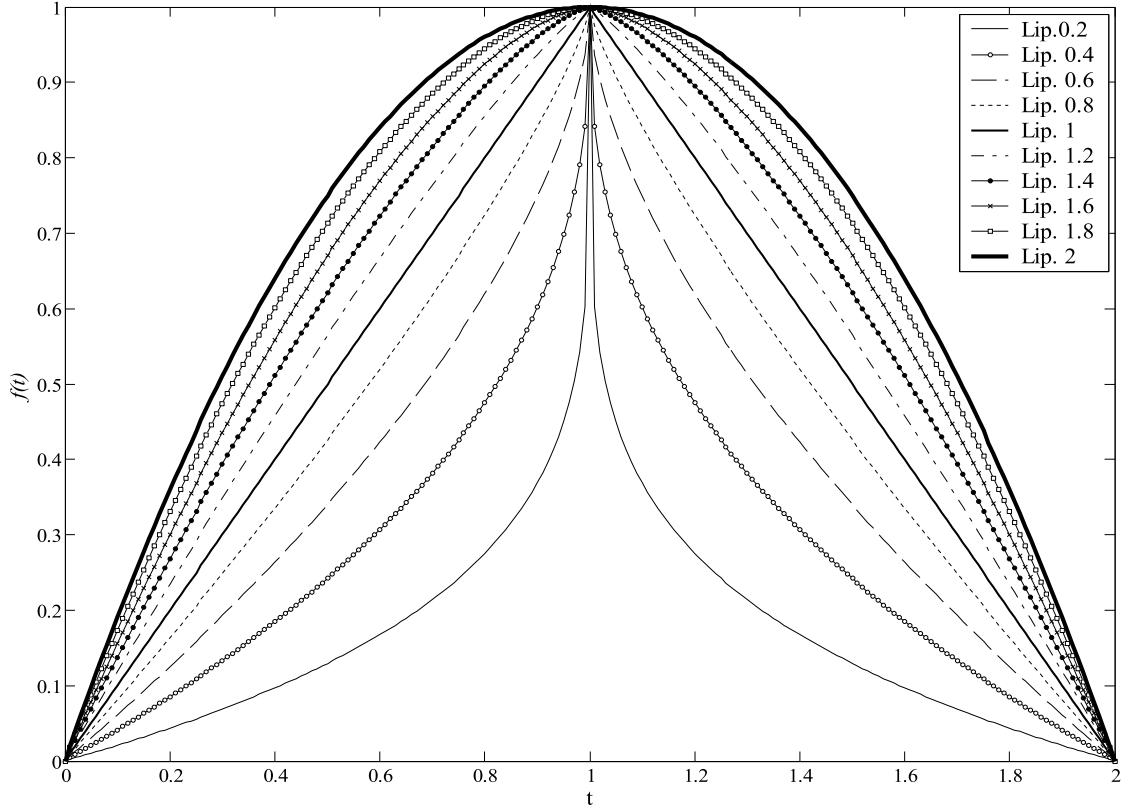


Figure 2.5: Function from Eq. (2.20) for different values of  $\alpha$  [5].

$$|Wf(u, s)| \leq As^{\alpha + \frac{1}{2}} \quad (2.21)$$

Taking the logarithm of both sides of the equation,

$$\log_2 |Wf(u, s)| \leq \log_2 A + \left( \alpha + \frac{1}{2} \right) \log_2 s \quad (2.22)$$

Equation 2.22 shows that the Lipschitz regularity at  $\nu$  corresponds to the maximum slope of  $\log_2 |Wf(u, s)|$  as a function of  $\log_2 s$  across the maxima lines that converge to  $\nu$  [5].

Also, to estimate the Lipschitz coefficient of a signal, the wavelet basis must have  $n > \alpha$  vanishing moments:

$$\int_{-\infty}^{\infty} t^k \psi(t) dt = 0 \quad , \quad 0 \leq k < n \quad (2.23)$$

The wavelet transform of the signal's polynomial approximation,  $f(t) = p_\nu(t) + \epsilon_\nu(t)$ , is:

$$Wf(u, s) = Wp_\nu(t) + W\epsilon_\nu(t) = W\epsilon_\nu(t) \quad (2.24)$$

which means that the WT estimates  $\alpha$  suppressing  $p_\nu$ , since the Lipschitz coefficient can be expressed in terms of the approximation error of the polynomial, as shown previously in Equation 2.19. This property is known as the polynomial suppression property.

## 2.2 Analog WT filters

Now that the basics of the wavelet transform theory have been presented, we move on to the theory of how to implement it, since we want to propose a sampling method that is based on it. The most common way to implement the wavelet transform is by using its digital form, applying the discrete wavelet transform. This method requires an analog-to-digital converter in order to compute the WT, which would be controversial for this project and also would require high power consumption. Fortunately, it is possible to implement the WT using analog wavelet filters: this project follows the approach presented by Haddad [5] and Karel [7]. The analog wavelet filters implemented throughout this project are based on two approximation methods ( $L_2$  and Padé) for the wavelet filter transfer function and on one design for circuit implementation, the transconductance amplifier-capacitance (Gm-C) method, and these are the objects of this subsection.

When designing an analog wavelet filter, its impulse response must correspond to the desired wavelet base, whose behavior and mathematical description are known. The approximation methods are used here to obtain a filter transfer function, whose impulse response,  $h'(t)$ , is very close to the desired impulse response corresponding to the wavelet basis,  $h(t)$ .

### 2.2.1 Padé approximation

The Padé method approximates the Laplace transform of  $h(t)$ , because it is assured to represent a transfer function of a possible filter. If the opposite was to be done, that is, the Padé approximation of  $h(t)$  was to be taken before applying the Laplace transform, the resulting transfer function would not necessarily be suitable for implementation. The Padé method approximates a function around one of its points and is based on the function's Taylor expansion coefficients, as shown in the following mathematical development.

Consider the function  $F(s)$ , whose truncated Taylor expansion is expressed in Equation 2.25, where  $\{c_0, c_1, \dots, c_k\}$  are the Taylor coefficients of  $F(s)$ .

$$F(s) = c_0 + c_1s + \dots + c_k s^k + O(s^{k+1}) \quad (2.25)$$



However, to build a filter, we need zeroes and poles, and  $F(s)$  has no poles, only zeroes. The function in Equation 2.25 can be transformed to a rational function by means of the Padé approximation:

$$\hat{F}(s) = \frac{P(s)}{Q(s)} = \frac{p_0 + p_1s + \dots + p_ms^m}{q_0 + q_1s + \dots + q_ns^n} \quad (2.26)$$

In Equation 2.26,  $\hat{F}(s)$  is the approximated  $F(s)$  function, for  $k = m + n$ . This means that the Taylor coefficients of  $\hat{F}(s)$  are known as  $\{c_0, c_1, \dots, c_{m+n}\}$ . Also, from Equation 2.26,  $\hat{F}(s) \cdot Q(s) = P(s)$  and:

$$\begin{bmatrix} c_0 & 0 & \dots & 0 \\ c_1 & c_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_k & c_{k-1} & \dots & c_{k-n} \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \quad (2.27)$$

Equation 2.27 shows that the coefficients  $\{q_0, q_1, \dots, q_n\}$  depend on the restrictions imposed over the coefficients  $\{p_0, p_1, \dots, p_k\}$ . These restrictions are:

1. The larger the order of the filter is, the better the approximation result will be. Therefore, we must maximize  $k$ ;
2.  $P(s)$  is of order  $m$ , according to Equation 2.26;
3.  $m \leq n$  for a physically realizable (causal) filter.

From restriction number 2 and Equations 2.26 and 2.27, we can conclude that  $\{p_{m+1}, \dots, p_k\} = 0$ . This leads to the following matricial representation:

$$\begin{bmatrix} p_{m+1} \\ \vdots \\ p_k \end{bmatrix} = [\hat{F}]_{m+1,k} \cdot [Q] = 0 \quad (2.28)$$

Therefore,

$$\underline{q} \in \text{Nullspace} \begin{bmatrix} c_{m+1} & \dots & c_0 & 0 \\ c_{m+2} & \dots & c_1 & c_0 \\ \vdots & \ddots & \vdots & \vdots \\ c_{m+n} & \dots & c_{m-1} & c_m \end{bmatrix} \quad (2.29)$$

In Equation 2.29,  $q_n = 1$ . The coefficients  $\{p_0, \dots, p_m\}$  are defined as:

$$\begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_m \end{bmatrix} = \begin{bmatrix} c_0 & 0 & \dots & 0 \\ c_1 & c_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & c_0 \\ c_m & c_{m-1} & \dots & c_{m-n} \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{bmatrix} \quad (2.30)$$

If the function that results from the approximation has a numerator of order  $m$  and a denominator of order  $n$ , the original function can be approximated up to order  $k = m + n$ . Therefore, the Padé approximation is defined by the values of  $m$  and  $n$ , and is referred to as a  $[m/n]$  Padé approximation.

Tables 2.1, 2.2, 2.3 and 2.4 show the  $[6/10]$  Padé and Taylor approximations coefficients for the first and second derivatives of the gaussian function, taken from an example in Haddad [5]. The first derivative of the gaussian function, hereafter referred to as gaus1, is expressed as  $\psi(t) = -2(t - 3)e^{-(t-3)^2}$ ; and the second derivative of the gaussian function, hereafter referred to as gaus2, is expressed as  $\psi(t) = (-2 + 4(t - 3)^2)e^{-(t-3)^2}$ .

Table 2.1: Taylor coefficients for expansions in the Laplace domain ( $k = 16$ ) of gaussian functions (part 1).

	$c_0$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$
gaus1	0	1.77	-5.31	8.41	-9.3	8.03	-5.74	3.54	-1.92
gaus2	0	0	1.77	-5.31	8.41	-9.3	8.03	-5.74	3.54

Table 2.2: Taylor coefficients for expansions in the Laplace domain ( $k = 16$ ) of gaussian functions (part 2).

	$c_9$	$c_{10}$	$c_{11}$	$c_{12}$	$c_{13}$	$c_{14}$	$c_{15}$	$c_{16}$
gaus1	0.94	-0.42	0.17	-0.066	0.023	-0.008	0.002	-0.007
gaus2	-1.92	0.94	-0.42	0.17	-0.066	0.023	-0.008	0.002

Table 2.3: Padé coefficients for  $Q(s)$  ( $n = 10$ ) of gaussian functions.

	$q_0$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$	$q_7$	$q_8$	$q_9$	$q_{10}$
gaus1	3.86E4	1.036E5	1.305E5	1.022E5	5.53E4	2.17E4	6.3E3	1.35E3	205.6	20.27	1
gaus2	3.78E4	1.007E5	1.264E5	9.86E4	5.33E4	2.09E4	6.1E3	1.3E3	199.7	19.91	1

## 2.2.2 $L_2$ approximation

The  $L_2$  method's approach is to minimize an error metric, specifically, the least-mean-square-error (LMSE) metric. Unlike Padé approximation, it is global, and not based around

Table 2.4: Padé coefficients for  $P(s)$  ( $m = 6$ ) of gaussian functions.

	$p_0$	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$
gaus1	-4.77	6.85E4	-2.2E4	6.1E3	-576.95	44.67	5.81
gaus2	-4.67	-13.21	1.11E4	-3.7E3	1.08E3	-131.28	13.54

one point of the function. Also, the  $L_2$  approximation is performed in the time domain, starting with the impulse response  $h(t)$  of a given filter. The next step is to minimize the LMSE with respect to the poles and zeroes of the filter. The LMSE integral is defined in Equation 2.31, where  $f(t)$  is the original function and  $\hat{f}(t)$  is the approximation.

$$\epsilon_{L_2} = \int_0^{\infty} [f(t) - \hat{f}(t)]^2 dt \quad (2.31)$$

Since, in this work, we want to approximate a wavelet basis,  $f(t)$  is a wavelet, and its approximation  $\hat{f}(t)$  is the transfer function of the filter in the time domain. Then, we know that  $f(t) = \psi(t)$  and  $\hat{f}(t) = h(t)$  have explicit expressions, and the error  $\epsilon_{L_2}$  can be minimized using standard numerical and/or computational methods.

Also, Equation 2.31 indicates that the  $L_2$  approximation, which is the minimal value for  $\epsilon_{L_2}$ , corresponds to the energy of the difference between the function and its approximations. This means that this method can be applied both in time and frequency domains, which is another advantage over the Padé approximation method: the  $L_2$  approximation method allows us to focus on specific points in the Laplace domain, for example, and start the approximation based on them. One disadvantage of this method, however, is that different starting points can yield different results, that is, we cannot assure, in general, the optimality of any approximation obtained with this method. Therefore, it also requires more computational effort than the Padé method.

### 2.2.3 Gm-C filters

Now that we know how to obtain the transfer function for an analog wavelet filter, we need to define the filter's topology. A good option for low-power applications is to design a Gm-C (transconductor-capacitor) filter. This filter implements a transfer function  $G/sC$ , where  $G$  is implemented by a tranconductor, and  $1/sC$ , by a capacitor, hence its name. This transfer function is characteristic of an integrator. A filter of  $n^{\text{th}}$  order is described by an  $n^{\text{th}}$  order differential equation, and can be implemented with  $n$  integrators. Indeed, Figure 2.6 [5] shows a single-ended Gm-C integrator. A Gm-C filter, composed of Gm-C integrators, can achieve transconductance values in the order of  $nA/V$ , what allows the use of capacitors with not so large capacitances. This relation can be deduced by evaluating Figure 2.6. From it, we see that the capacitor voltage is expressed as:

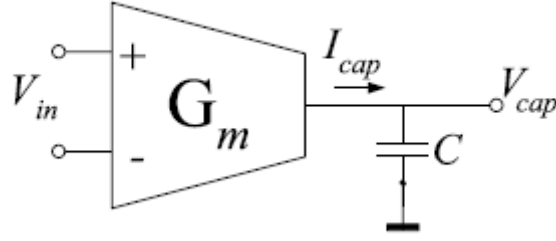


Figure 2.6: Gm-C integrator [5].

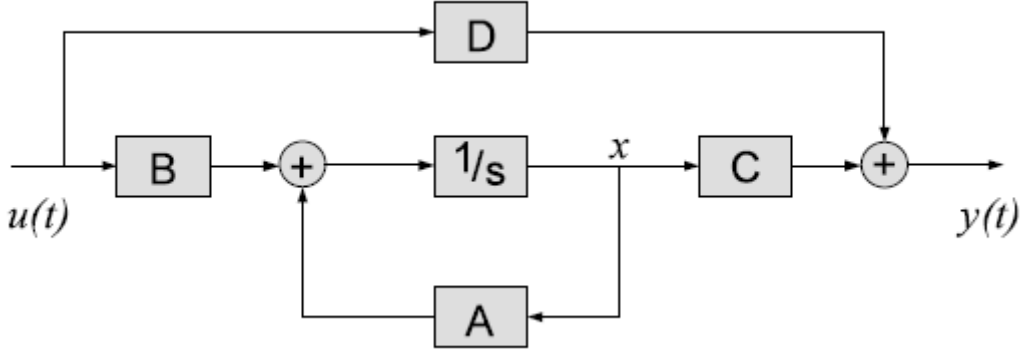


Figure 2.7: System block diagram for state-space representation [5].

$$V_{cap} = \frac{I_{cap}}{sC} = \frac{G_m V_{in}}{sC} \quad (2.32)$$

and the time constant  $\tau$ , as:

$$\tau = \frac{C}{G_m} \quad (2.33)$$

From control theory, it is known that an  $n^{\text{th}}$  order differential equation can be transformed to a set of first order differential equations using a state-space representation. The typical system of state-space differential equations is expressed in Equation 2.34, where  $u(t)$  is the system's input;  $y(t)$  is the system's output; and  $x(t)$  is the state-space variable. Figure 2.7 [5] shows the block diagram for the system represented in the state-space form.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (2.34)$$

To describe the system in its state-space representation, we need to obtain the elements of the matrices  $A$ ,  $B$ ,  $C$  and  $D$ . Knowing that the system's transfer function expressed in terms of the state-space matrices is as described in Equation 2.35, where  $I$  is the identity matrix, and knowing the desired transfer function of the filter, the state-space matrices'

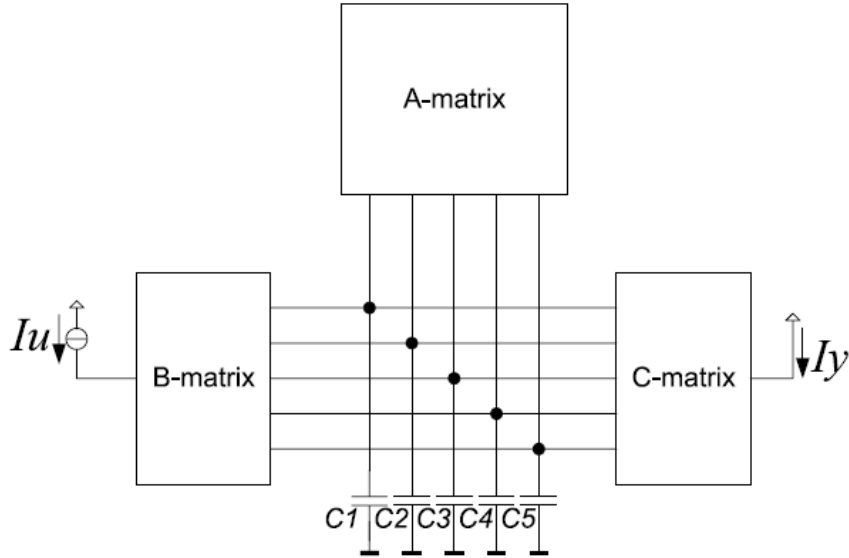


Figure 2.8: Filter implementation scheme with state-space matrices [5].

elements can be determined. For more information on state-space representations or how these equations are derived, please refer to a control engineering theory book, such as Ogata [22].

$$H(s) = C \cdot (sI - A)^{-1} \cdot B + D \quad (2.35)$$

Once the state-space matrices are obtained, they must be implemented in order to compose the filter, in the manner shown in Figure 2.8 [5]. However, it would be advantageous if these matrices were in their sparse representations, so that there are more elements that equal zero, and, consequently, less coefficients to be implemented in the circuit. There are many ways to obtain an orthogonal representation of the state-space matrix [5], but the chosen strategy for this project is to transform the matrices to their corresponding orthonormal ladder structure. The orthonormal ladder structure is unique for a given transfer function and its general configuration is:

$$A = \begin{bmatrix} 0 & a_1 & 0 & 0 & \cdots & 0 & 0 \\ -a_1 & 0 & a_2 & 0 & \cdots & 0 & 0 \\ 0 & -a_2 & 0 & a_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & -a_{n-1} & a_n \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & b_n \end{bmatrix}$$

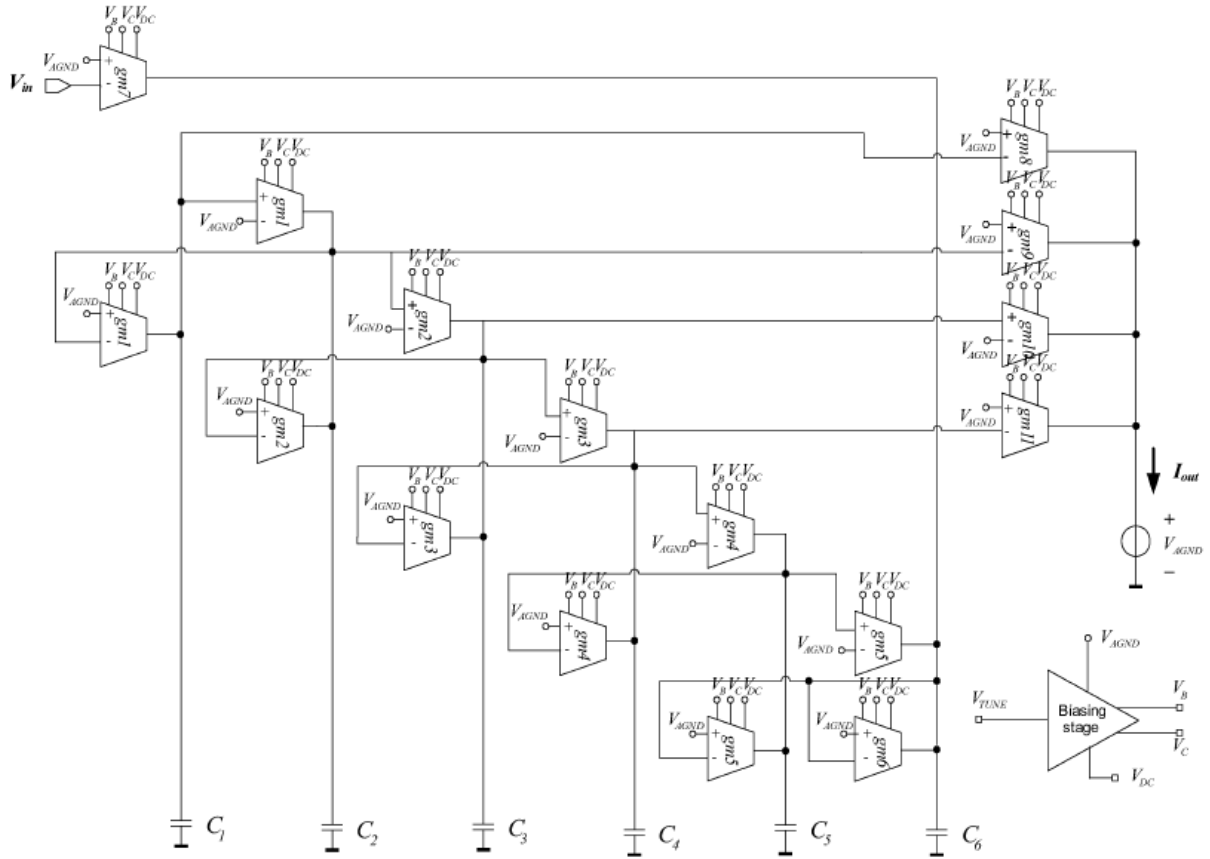


Figure 2.9: Gm-C filter implementation of a 6<sup>th</sup> order filter with orthonormal ladder structure (adapted from Karel et al. [7]).

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \cdots & c_{n-1} & c_n \end{bmatrix}$$

$$D = [0]$$

Finally, putting together the schematics from Figures 2.6 to 2.8, we can obtain the Gm-C filter topology, shown in Figure 2.9 for a sixth-order filter, adapted from Karel et al. [7].

## 2.3 Summary of the Chapter

This chapter presented a review of the theoretical framework that is necessary to the understanding of the development in the next chapters. First, the continuous wavelet transform was presented, along with its main properties, followed by the Lipschitz exponent definition and how it can be estimated with the wavelet transform. The last section shows one way to design and implement analog wavelet filters, which will be used in this project to compute the wavelet transform.

The chapters that follow contain the project's development *per se*, starting with the system-level design in Chapter 3.

# Chapter 3

## System-Level Development

In this chapter, the project is studied at a system-level. The proposed sampling method is presented, as well as the polynomial reconstruction algorithm that enables its verification. The theoretical portion is followed by the methodology adopted for the system simulations and their results.

### 3.1 Sampling algorithm

The flowchart for the sampling algorithm is shown in Figure 3.1. The input signal is processed and its critical points must be identified, as well as the Lipschitz exponent at local maxima and minima. After choosing the WT parameters (bases and scales), the local maxima and minima are identified by computing the first-order WT of the input signal. Because of the wavelet zoom property, the coefficients line for the smallest scale is used for this purpose, and the maxima and minima points are localized by identifying this coefficients line's zero-crossings. Similarly, to detect the inflection points, the second-order WT is computed, but, for this purpose, the coefficients line for a large scale is selected, and its zero-crossings are detected. At this point, some approximations may be necessary:

- The maxima and minima identification prevails over the inflection points' localization, that is, if the algorithm detects a local extremum at the same point as an inflection, this point is considered to be an extremum;
- This algorithm considers that there is only one inflection point between a local maximum and a local minimum. If more than one inflection point is detected, its location is approximated by taking the mean of the positions of all the points detected; if no inflection is detected, its location is approximated as the midrange of the maximum and the minimum points' positions.

The second-order WT is also used to estimate the Lipschitz exponent. This is done by applying the definition in Equation 2.22, and selecting the instants which correspond to the



local maxima and minima, identified in the previous steps.

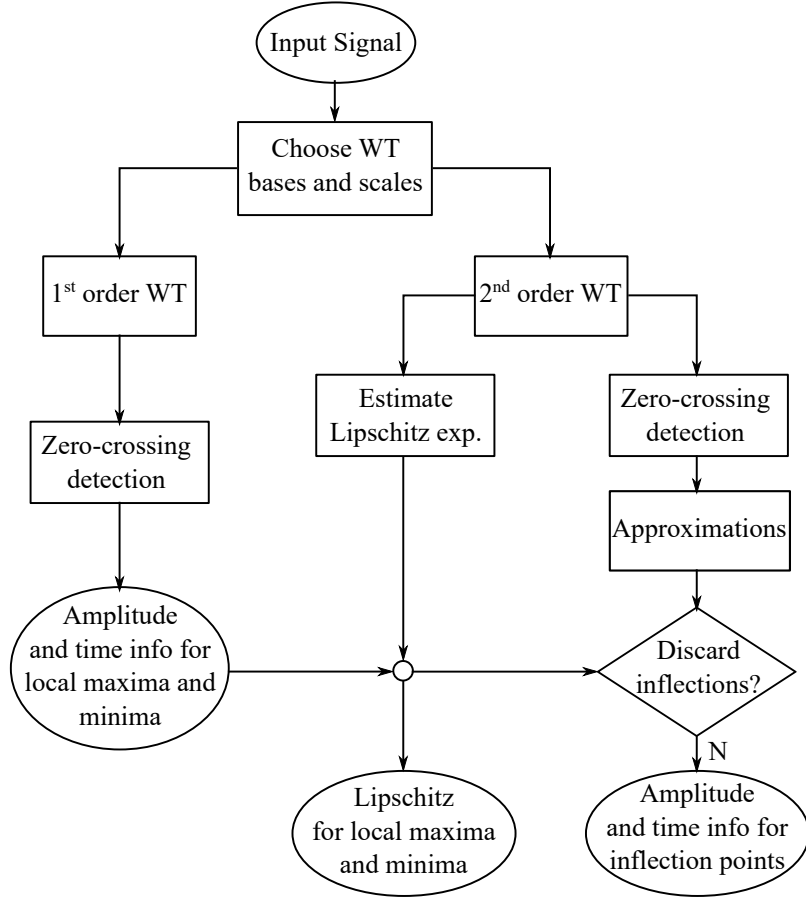


Figure 3.1: Wavelet-based ADC sampling algorithm.

### 3.2 Reconstruction algorithm

In order to verify that the sampled information allows recovery of the input signal, a reconstruction algorithm is also proposed, as illustrated in Fig. 3.2.

Since the critical points have been sampled, we choose to reconstruct the signal piecewise, with each piece determined by one maximum or minimum point and one inflection point. The morphology of the signal between the critical points is determined by the Lipschitz exponent, and each piece is expressed as a polynomial. This polynomial is based on the function in Equation 2.20, modified to allow time and amplitude shifting. The result is:

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

where

- $A_i$  and  $A_m$  are the sampled amplitudes for, respectively, the inflection points and the

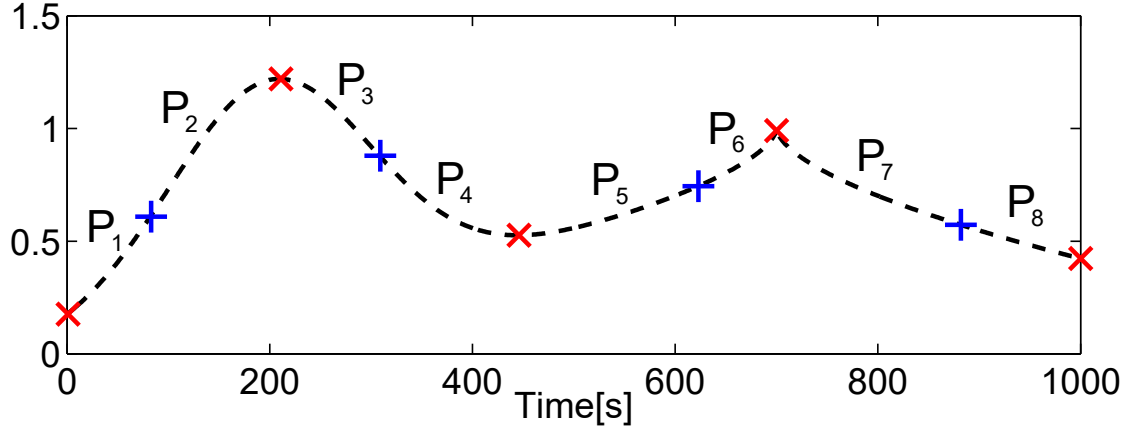


Figure 3.2: Identified critical points of a test signal: ‘x’ represents a local maximum or minimum, and ‘+’ represents an inflection point. ‘P<sub>n</sub>’ are the polynomials that reconstruct the signal piecewise.

local maxima or minima;

- $\alpha_m$  is the estimated Lipschitz exponent at the local maximum or minimum;
- $t_0$  and  $t_f$  are the initial and final points of the polynomial  $P_n$ , that is, one of them must be an inflection while the other must be either a local maximum or a local minimum;
- $\mu$  is the time interval for  $P_n$  i.e.  $\mu = t_f - t_0$ ;
- $\tau$  is the position of the local maximum or minimum.

The variables  $\tau$  and  $\mu$  allow shifting in time. The shift in amplitude is achieved by multiplying the expression by  $(A_m - A_i)$ , and then adding  $A_i$ . The expression  $(A_m - A_i)$  also indicates the concavity of the signal: if  $A_m > A_i$ , then  $(A_m - A_i) > 0$  and the extremum is a local maximum; if  $A_m < A_i$ , then  $(A_m - A_i) < 0$  and the extremum must be a local minimum. The addition of  $A_i$  to the expression guarantees the continuity of the signal when all pieces are put together.

### 3.3 Methodology

The algorithms were evaluated by simulation in the softwares MATLAB and Simulink. All the scripts necessary to run the tests are available in Appendix II. The parameters for the measurements are detailed in the following subsections.

#### 3.3.1 The test signal

A test signal, shown in Figure 3.3 and expressed by Equation 3.2, was chosen to evaluate these algorithms. This function was chosen as the primary test signal because it is well-behaved, making it easier to observe the differences between the reconstructed signal and

the input signal; and also because of its singular point, which should require a high sampling rate in trivial ADCs to achieve a good representation of the signal.

$$f(t) = \exp \left[ -\frac{(t - 0.2)^2}{2 \cdot 0.1^2} \right] + \frac{0.75^{0.6} - |t - 0.7|^{0.6}}{0.75^{0.6}} \quad (3.2)$$

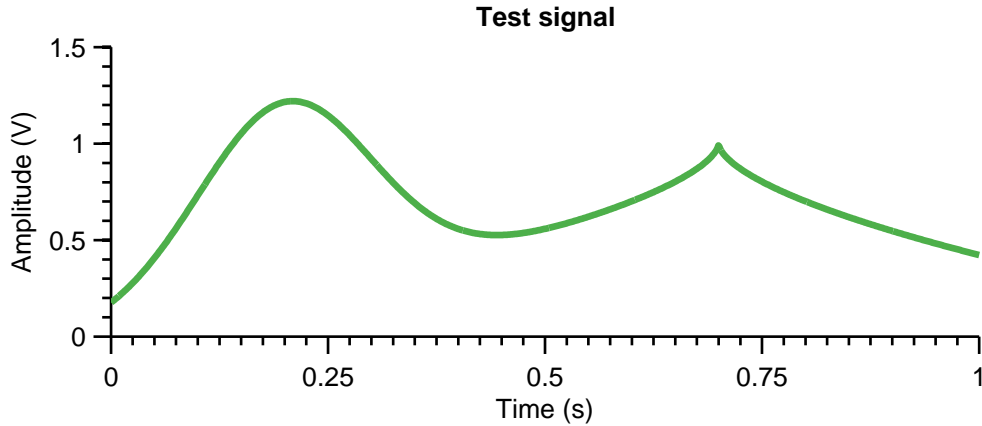


Figure 3.3: Test signal.

### 3.3.2 WT parameters

The parameters for the wavelet transform are its bases and scales. Since the algorithm applies first- and second-order wavelet bases, we chose wavelet bases from the same families in these two versions. The bases available in MATLAB that corresponded to this criterion are the ones indicated in Table 3.1. The tests were run with two sets of scales: the first one with the smallest scale being  $a = 1$  and every integer scale until the largest one,  $a = 64$ ; the second set consists of four scales:  $a = 16$ ,  $a = 32$ ,  $a = 48$  and  $a = 64$ .

Table 3.1: Chosen wavelet bases

<b>Family</b>	<b>1st order</b>	<b>2nd order</b>
Gaussian	gaus1	gaus2, mexh
Daubechies	db1	db2
Biorthogonal	bior1	bior2
Reverse biorthogonal	rbio1	rbio2

The gaussian wavelet bases are the derivatives of  $p$ -th order of the gaussian function, described in Equation 3.3. The gaussian wavelet bases family is normally represented in literature as  $\text{gaus}N$ , where  $N$  is the number of vanishing moments of the base. Therefore, the wavelet bases  $\text{gaus1}$  and  $\text{gaus2}$  are, respectively, the first and second order derivatives of the gaussian function. The mexican hat basis,  $\text{mexh}$ , is proportional to  $\text{gaus2}$ , having two

vanishing moments as well. Figure 3.4 [8] shows the plots of these three gaussian wavelet bases.

$$\begin{aligned} f(x) &= C_p e^{-x^2} \\ \|f^{(p)}\|^2 &= 1 \end{aligned} \tag{3.3}$$

The Daubechies bases were described by Ingrid Daubechies [23]. They are orthonormal wavelets of compact support, with the generic expression of Equation 3.4 [24], where  $\psi$  is the wavelet,  $\phi$  is the scaling function, and  $h_n$  is the expression for both the associated reconstruction and decomposition filters. The Daubechies wavelet bases are referred to as  $dbN$ , where  $N \in \mathbb{Z}_+^*$  is the number of vanishing moments.

$$\psi(x) = \sqrt{2} \sum_n (-1)^n h_{-n+1} \phi(2x - n) \tag{3.4}$$

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n) \tag{3.5}$$

The biorthogonal bases are biorthogonal splines, characterized by the general expression of Equation 3.6 [24, 25]. Their associated reconstruction and decomposition filters,  $h_n$  and  $g_n$ , are not necessarily identical, and thus confer both the wavelet function  $\psi$  and the scaling function  $\phi$  one synthesis and one analysis form. In this project, the analysis form of the wavelet function is applied. The reverse biorthogonal basis is derived from the biorthogonal by inverting the analysis and synthesis functions. The biorthogonal bases are represented as  $biorNr.Nd$ , and the reverse biorthogonal bases, as  $rbioNr.Nd$ , where  $Nr$  and  $Nd$  are the number of vanishing moments of the synthesis and analysis functions, respectively.

$$\psi(x) = \sqrt{2} \sum_n g_n \phi(2x - n) \tag{3.6}$$

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n) \tag{3.7}$$

Figure 3.5 [8] shows the wavelet bases from the Daubechies and biorthogonal families. For  $n = 1$ , they are all equal to the Haar wavelet, in Figure 3.5(a). The others are the bases for  $n = 2$  vanishing moments: (b) for  $db2$ , (c) for  $bior2.2$ , and (d) for  $rbio2.2$ .

### 3.3.3 Error metrics

In order to measure the quality of the signal reconstruction, two metrics are used in this project: the RMS error metric and the correlation coefficient between the original and the recovered signals.

The RMS error is the root-mean-squared error, expressed by Equation 3.8, where  $l$  is the length of the original signal  $y$  and of the recovered signal  $\hat{y}$ . This error is represented by a scalar value, and allows us to compare the similarity between the two signals globally.

$$e_{RMS} = \sqrt{\frac{\sum_{k=1}^l (\hat{y}_k - y_k)^2}{l}} \quad (3.8)$$

The Pearson correlation coefficient  $\rho$  between the discrete random variables  $X$  and  $Y$  is expressed by Equation 3.9, where:  $E(X)$  is the expected value of the variable  $X$ ;  $\sigma(X)$  is the standard deviation of the variable  $X$ ; and  $cov(X, Y)$  is the covariance of the variables  $X$  and  $Y$ . The correlation coefficient is a scalar of absolute value between 0 and 1. The covariance of  $X$  and  $Y$  is the first central moment of the variables. If  $cov(X, Y) = 0$ , then  $X$  and  $Y$  are completely independent and  $\rho = 0$ , which also means that  $X$  and  $Y$  are uncorrelated [26]. The correlation coefficient between the original and the recovered signals, then, will express how much their morphologies are alike, or correlated.

$$\rho = \frac{E[XY] - E[X] \cdot E[Y]}{\sigma_X \cdot \sigma_Y} = \frac{cov(X, Y)}{\sigma_X \cdot \sigma_Y} \quad (3.9)$$

### 3.3.4 Quantization

To analyze the effect of quantization of the sampled values (amplitudes and Lipschitz exponents) in the signal recovery, a uniform quantizer was implemented, with resolutions of 4, 8 and 12 bits, representing low, medium, and high resolution.

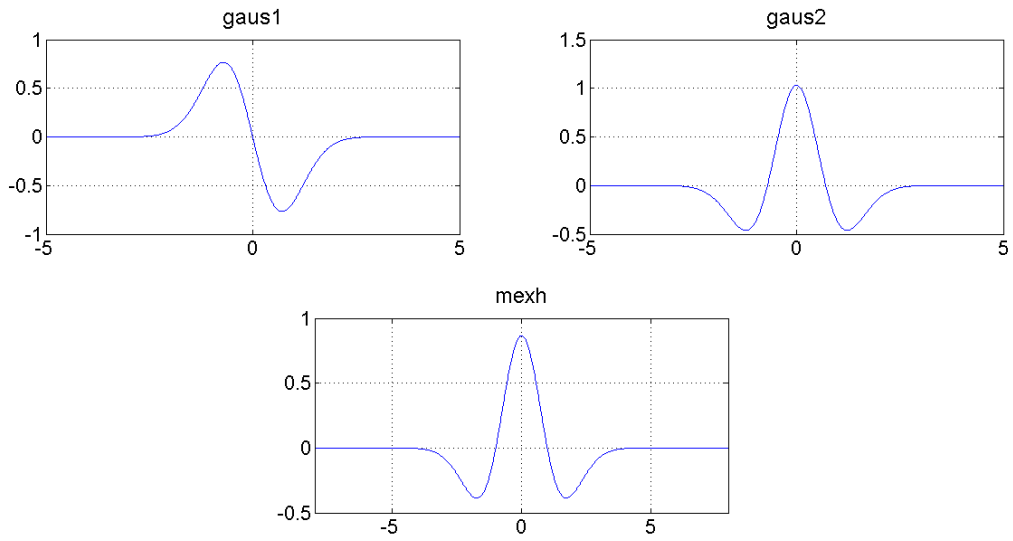


Figure 3.4: Gaussian wavelet bases [8].

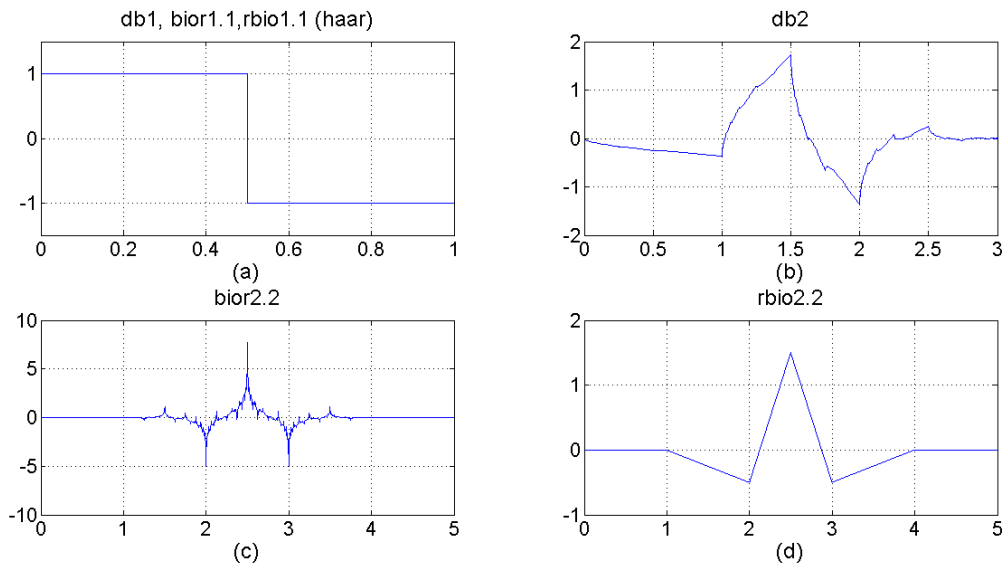
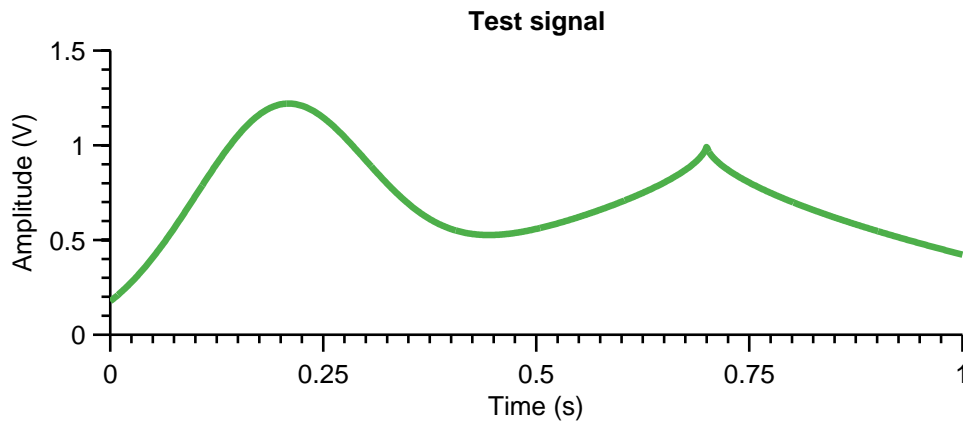


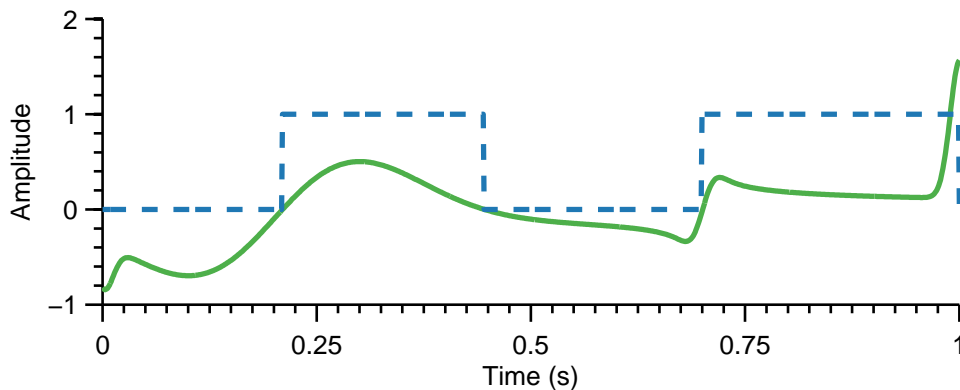
Figure 3.5: (a) Daubechies and biorthogonal bases with  $n = 1$  vanishing moment. Bases with  $n = 2$  vanishing moments: (b) Daubechies (c) Biorthogonal (d) Reverse biorthogonal [8].

### 3.4 System Results

According to the flowchart in Figure 3.1, the first step in the algorithm is to determine the WT bases and scales, in order to identify the signal's critical points. Let us evaluate the algorithm step-by-step with one of the test configurations: the pair of gaussian wavelet bases, gaus1 and gaus2, and the second set of scales (from  $a = 16$  to  $a = 64$ ). With the first-order WT results, the local maxima and minima points can be detected, as Figure 3.6 shows. Note that the local maxima and minima are correctly identified with the first-order wavelet transform processing. Figure 3.6(b) shows the coefficients line for  $a = 16$  and the zero-crossing detection output. The second-order WT is used to detect the inflection points positions, which is shown in Figure 3.7. Figure 3.7(b) shows the coefficients line for the largest scale,  $a = 64$ , and the zero-crossing detection result. In this case, an approximation was needed, because the coefficients line crosses zero thrice in the last portion of the signal, meaning that two inflection points are detected initially. Apart from this approximation, the inflection points appear to have been detected with accuracy by applying the coefficients line for the largest scale.

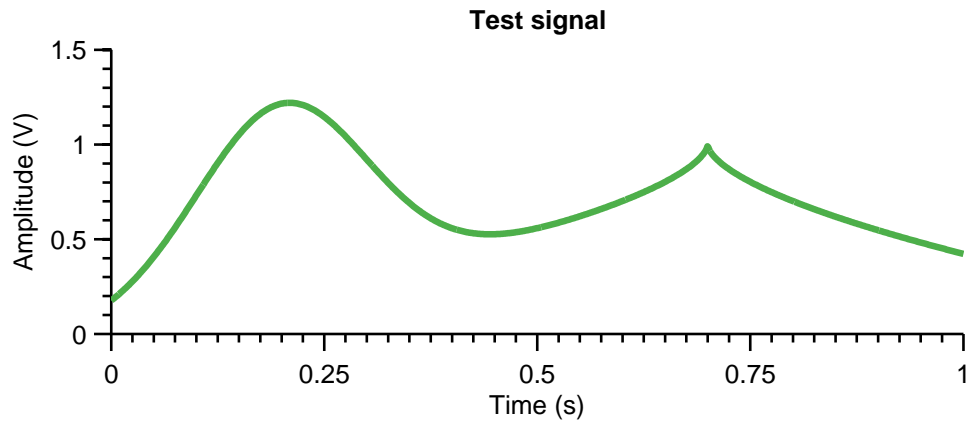


(a) Input signal.

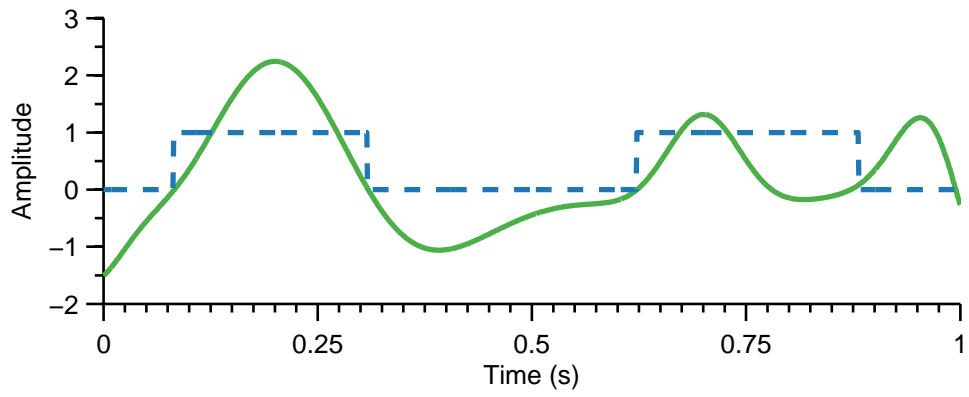


(b) 1st order WT coefficients line for scale  $a = 16$  (green) and zero-crossing detection output (dashed, blue).

Figure 3.6: Critical points detection: local maxima and minima.



(a) Input signal.



(b) 2nd order WT coefficients line for scale  $a = 64$  (green) and zero-crossing detection output (dashed, blue).

Figure 3.7: Critical points detection: inflection points.



The second-order WT results are also used to estimate the Lipschitz exponent at every point of the signal. The `gaus2` coefficients lines for the four scales are shown in Figure 3.8, and Figure 3.9 displays the result of the Lipschitz exponent estimation. The peaks of the coefficients line in Figure 3.8 all happen at the same position when evaluating them for different scales. The Lipschitz exponent is estimated based on the decay of these maxima across scales at the identified critical points positions. As an exercise, let us estimate the Lipschitz coefficient, manually, by evaluating the figures and applying the definition, for the first local maximum point of the input signal. The first maximum occurs at approximately 0.2s, so the first step is to determine the values of the coefficients lines for all scales at that point of interest:

$$\begin{aligned}
 a = 16 & \rightarrow \text{amplitude} \approx 0.1 \\
 a = 32 & \rightarrow \text{amplitude} \approx 0.5 \\
 a = 48 & \rightarrow \text{amplitude} \approx 1.3 \\
 a = 64 & \rightarrow \text{amplitude} \approx 2.2
 \end{aligned}$$

Then, applying the definition of the Lipschitz coefficient ( $\alpha$ ) estimation, we must calculate the decay across scales. This is achieved by evaluating the decay in amplitude at every two consecutive scales, and then taking the mean value of those results:

$$\begin{aligned}
 \alpha &= \frac{\alpha_1 + \alpha_2 + \alpha_3}{3} - 0.5 \\
 \alpha_1 &= \frac{\log|2.2| - \log|1.3|}{\log|64| - \log|48|} = 1.83 \\
 \alpha_2 &= \frac{\log|1.3| - \log|0.5|}{\log|48| - \log|32|} = 2.36 \\
 \alpha_3 &= \frac{\log|0.5| - \log|0.1|}{\log|32| - \log|16|} = 2.32
 \end{aligned}$$

$$\therefore \alpha = 2.17 - 0.5 = 1.67$$

In Figure 3.9, the Lipschitz coefficient at  $t = 0.2$  s appears to be  $\alpha \approx 1.7$ . Actually, its precise value is 1.7654, which is fairly close to the result obtained analytically, above, with approximated values from the plots in Figure 3.8.

With the localization of the critical points, the input and Lipschitz signals can be sampled. Figure 3.10(a) shows the amplitude sampling, which happens at all identified critical points. The Lipschitz coefficient is sampled only at the detected local maxima and minima, as shown in Figure 3.10(b). It is worth noting that the estimated Lipschitz exponent at the critical points cannot be larger than 2, because this is the number of vanishing moments

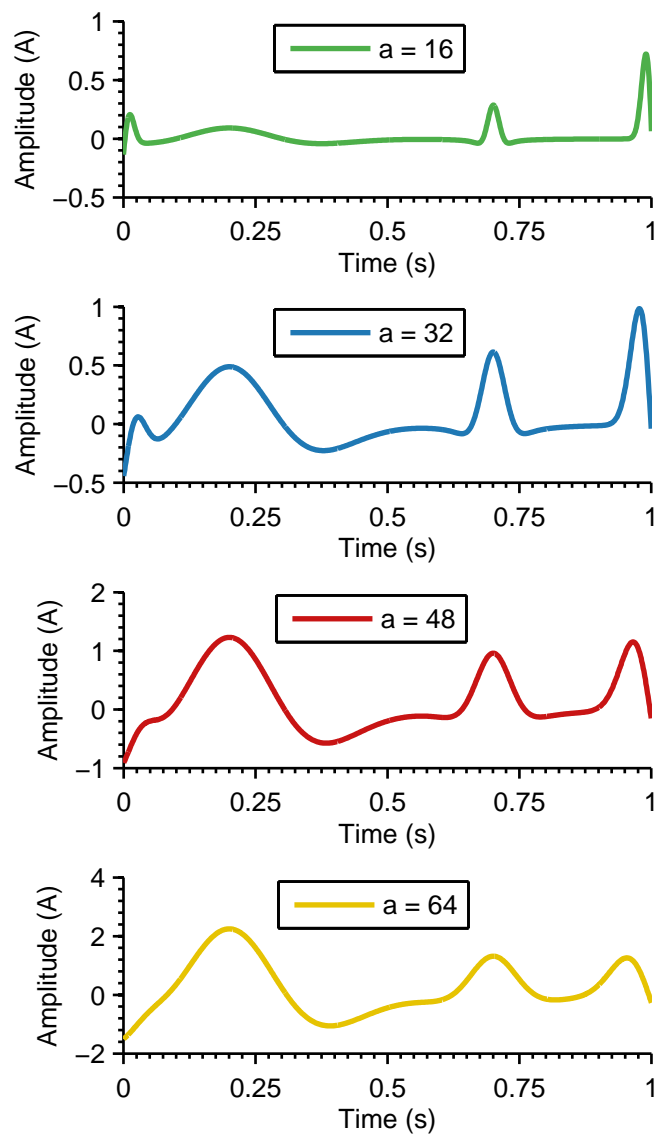
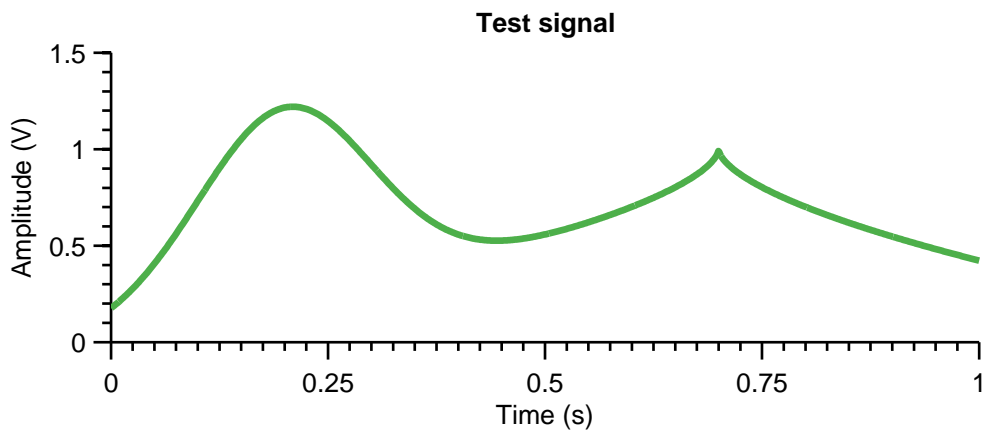
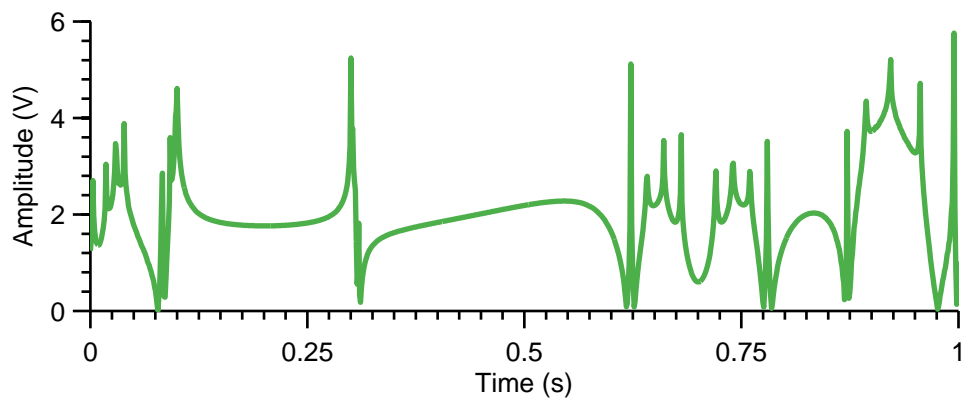


Figure 3.8: Coefficients lines for second order wavelet transform at scales 16, 32, 48 and 64.



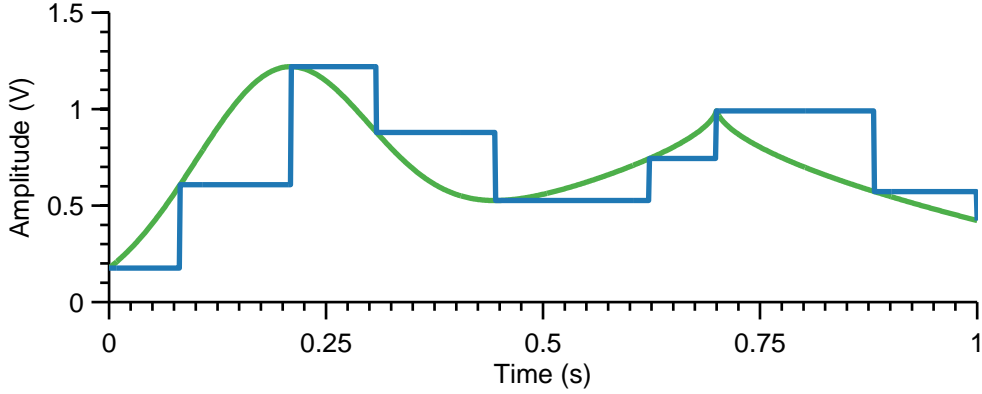
(a) Input signal.



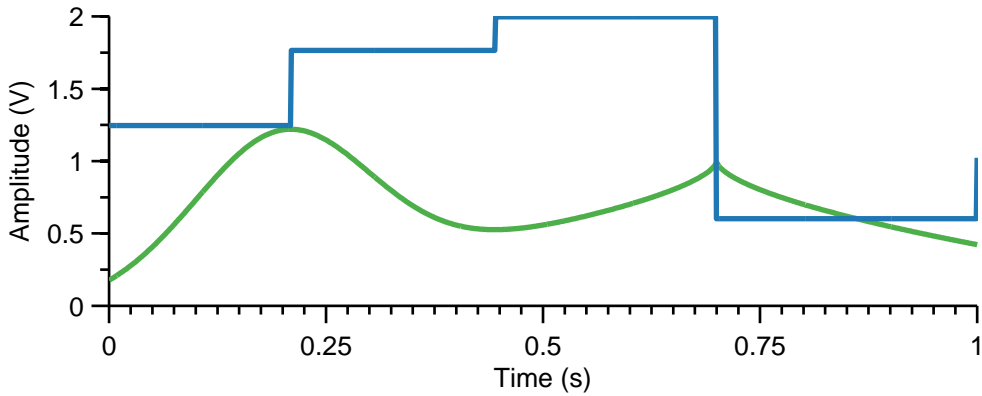
(b) Lipschitz exponent estimated at all points of the input signal.

Figure 3.9: Lipschitz exponent estimation.

of the wavelet basis that is used to estimate  $\alpha$ , in this case, `gaus2`. Figure 3.11 shows the amplitude and Lipschitz coefficient sampling signals, but quantized for three different resolutions: 4, 8, and 12 bits. Note that there is a considerable difference between the values for 4 bits resolution and the values for 8 and 12 bits resolutions. Also, the latter ones are practically indistinguishable.



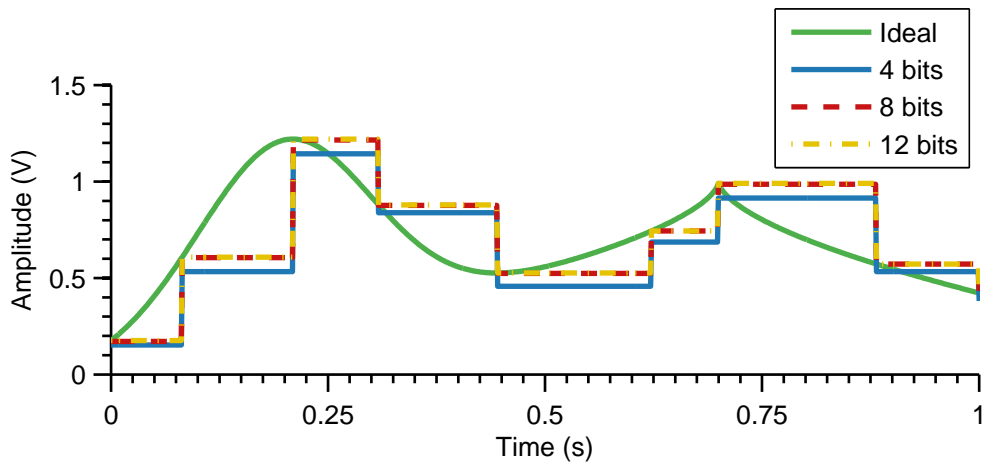
(a) Amplitude sampling.



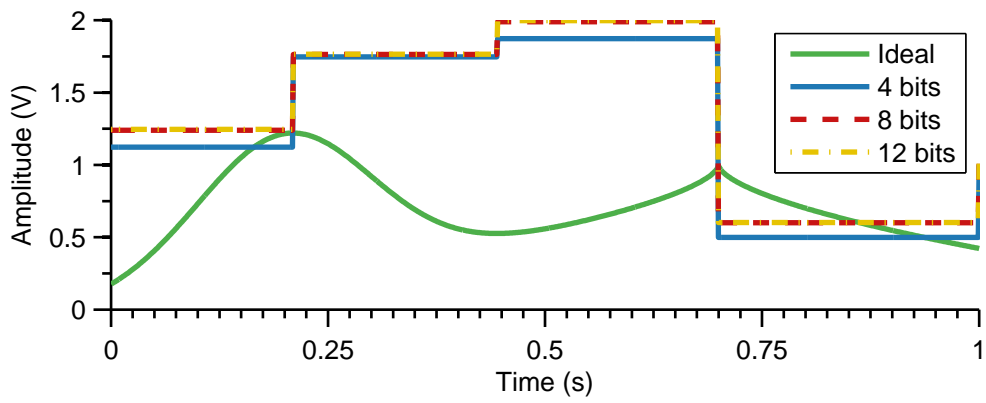
(b) Lipschitz coefficient sampling.

Figure 3.10: Sampled signals.

In order to test this algorithm, the signal must be recovered from these outputs. The polynomial reconstruction algorithm for the ideal case yields the result shown in Figure 3.12. The signal reconstruction is successful, with a high correlation coefficient between the recovered signal and the input signal of  $\rho = 0.98853$ , and an RMS error of  $e_{RMS} = 0.008547$ . Table 3.2 contains the polynomial parameters used to reconstruct the signal piece by piece for this case, in the manner illustrated earlier by Figure 3.2. Table 3.2 is in terms of the parameters in Equation 3.1, where  $A_i$  and  $A_m$  are the amplitudes of the critical points,  $\alpha_m$  is the Lipschitz exponent, measured at the local maxima and minima, and the others are time parameters. The Lipschitz exponent indicates the morphology of the signal around a local maximum or minimum, which is why it is only sampled at these points; also, this means that its value is used for two polynomials, for example,  $P_2$  and  $P_3$ , that reconstruct the signal around the first identified maximum, localized at 0.2110 s and with an amplitude



(a) Amplitude sampling.



(b) Lipschitz coefficient sampling.

Figure 3.11: Quantized sampled signals for 4, 8, and 12-bit resolutions.

of 1.2204 V.  $P_2$  starts with an inflection point, identified at 0.0830 s and with an amplitude of 0.6087 V, and  $P_3$  ends with an inflection, which occurs at 0.3090 s with an amplitude of 0.8790 V.

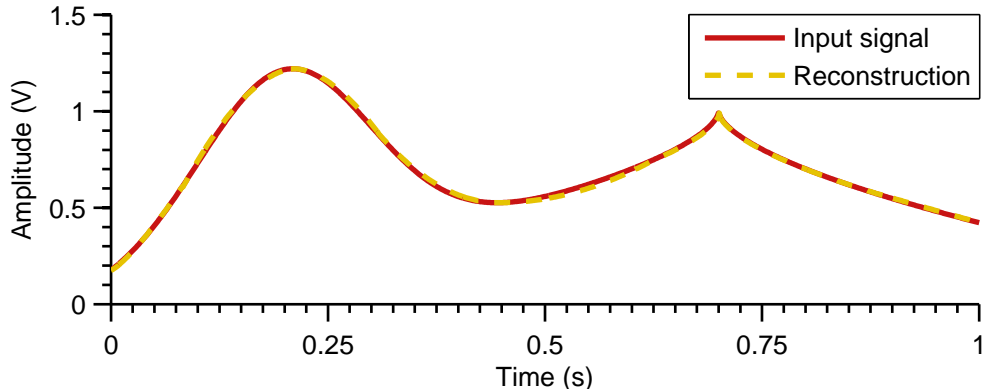


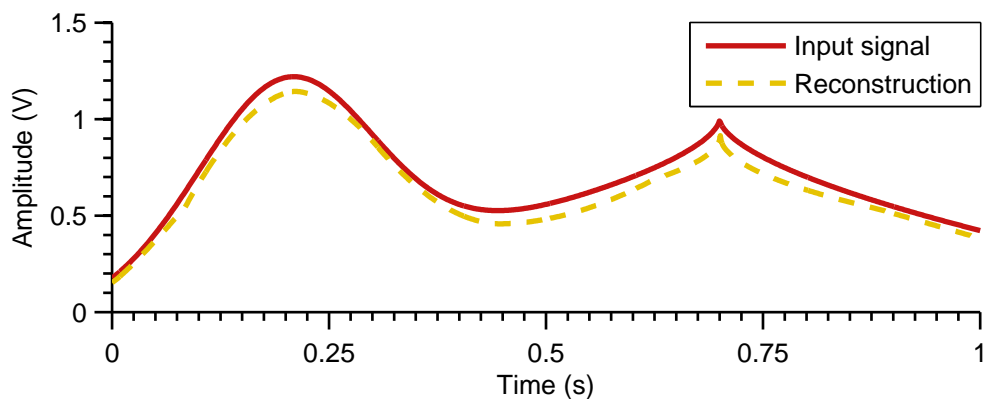
Figure 3.12: Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for gaus1 and gaus2 wavelet bases, set of scales from 16 to 64, and ideal (not quantized) values.

Table 3.2: Polynomial reconstruction parameters for gaus1, gaus2, scales from 16 to 64 (ideal case)

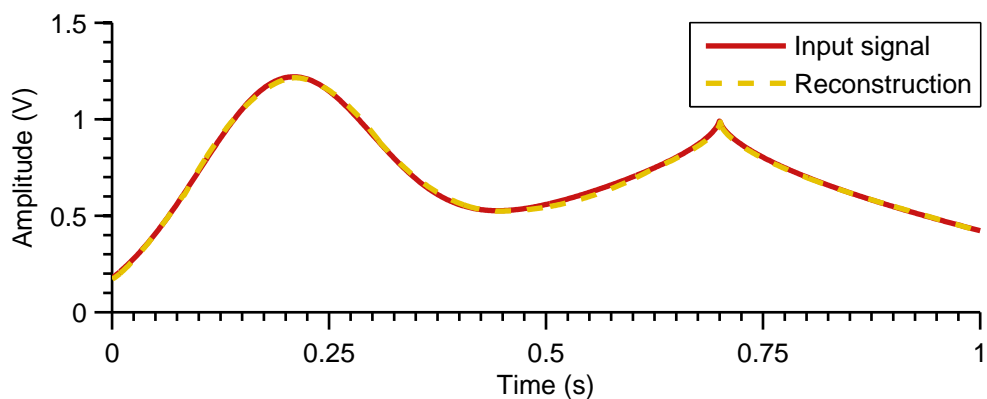
$n$	$A_i$	$A_m$	$\tau$	$\mu$	$\alpha_s$	$t_0$ (s)	$t_f$ (s)
1	0.6087	0.1759	0.0010	0.0820	1.2464	0.0010	0.0830
2	0.6087	1.2204	0.2110	0.1280	1.7654	0.0830	0.2110
3	0.8790	1.2204	0.2110	0.0980	1.7654	0.2110	0.3090
4	0.8790	0.5263	0.4460	0.1370	1.9970	0.3090	0.4460
5	0.7442	0.5263	0.4460	0.1770	1.9970	0.4460	0.6230
6	0.7442	0.9909	0.7000	0.0770	0.6024	0.6230	0.7000
7	0.5726	0.9909	0.7000	0.1820	0.6024	0.7000	0.8820
8	0.5726	0.4229	1.0000	0.1180	1.0232	0.8820	1.0000

Figures 3.13(a), (b) and (c) show the results for the three quantization resolutions, respectively 4, 8 and 12 bits. As noticed before, the quality of the reconstruction with 4-bit resolution is clearly below the quality of the ones with 8 and 12 bits, which come very close to the ideal case of Figure 3.12. The correlation coefficients and RMS errors for each of these cases, as well as the results for other tests, for scales from 16 to 64 are displayed in Tables 3.3 and 3.4. In these tables, the column labeled ‘Ideal’ contains the metrics for the case when the signal is recovered with the values before they pass through the quantizer, and this serves as a comparison parameter for the other results. This is the case illustrated in Figure 3.12, for the bases gaus1 and gaus2. The column labeled ‘Amplitude’ contains the results for when only the amplitudes are quantized, and the Lipschitz coefficient values are the ideal ones. Similarly, the column labeled ‘ $\alpha$ ’ evaluates the effects of quantizing only the Lipschitz exponent in the signal reconstruction. The column labeled ‘Amplitude +  $\alpha$

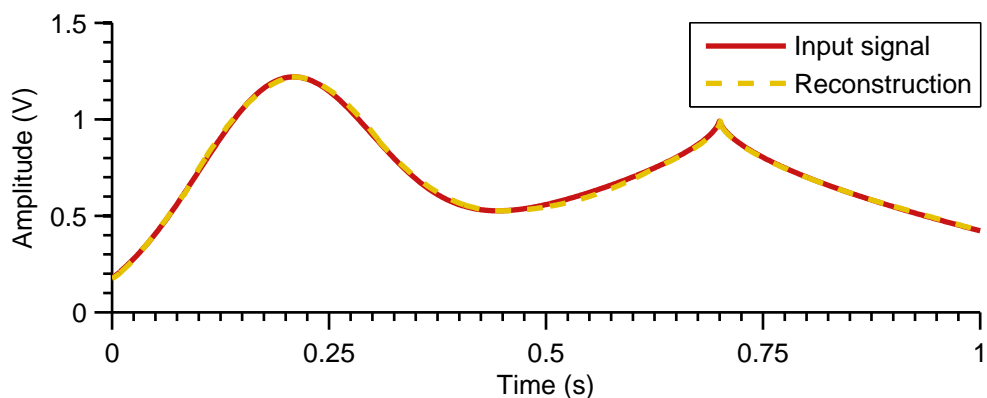
shows the results when the amplitudes and the Lipschitz coefficients are quantized at the same resolution, which are the cases shown in Figure 3.13. Following this same pattern, the metrics obtained from the tests for the other set of scales, ranging from 1 to 64, are displayed in Tables 3.5 and Tables 3.6.



(a) 4-bit resolution



(b) 8-bit resolution



(c) 12-bit resolution

Figure 3.13: Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for gaus1 and gaus2 wavelet bases, set of scales from 16 to 64, and quantized values.

The tables allow us to compare the results according to the chosen bases, the set of

Table 3.3: Correlation coefficient (%) for scales from 16 to 64

Bases	Quantization (bits)									
	Ideal	Amplitude			$\alpha$			Amplitude + $\alpha$		
		12	8	4	12	8	4	12	8	4
gaus1, gaus2	98.9	98.9	98.9	96.0	98.9	98.9	97.0	98.9	98.9	94.7
gaus1, mexh	99.0	99.0	99.0	96.5	99.0	99.0	94.2	99.0	98.8	90.0
db1, db2	73.9	73.9	74.7	63.1	73.8	73.8	75.3	73.9	74.6	64.2
bior1, bior2	88.4	88.5	88.8	79.5	88.4	88.5	89.3	88.5	88.9	82.1
rbio1, rbio2	96.0	96.0	95.7	94.7	96.0	96.0	94.9	96.0	95.7	94.7

Table 3.4: RMS error (%) for scales from 16 to 64

Bases	Quantization (bits)									
	Ideal	Amplitude			$\alpha$			Amplitude + $\alpha$		
		12	8	4	12	8	4	12	8	4
gaus1, gaus2	0.855	0.855	0.877	6.119	0.855	0.852	1.379	0.855	0.874	6.381
gaus1, mexh	0.869	0.874	1.032	6.291	0.870	0.885	2.155	0.875	1.052	7.239
db1, db2	4.510	4.498	4.341	6.318	4.510	4.516	4.487	4.450	4.348	6.308
bior1, bior2	2.935	2.925	2.809	5.082	2.934	2.917	2.634	2.924	2.792	5.173
rbio1, rbio2	2.068	2.071	2.194	6.358	2.067	2.057	2.186	2.070	2.185	6.497



Table 3.5: Correlation coefficient (%) for scales from 1 to 64

Bases	Quantization (bits)									
	Ideal	Amplitude			$\alpha$			Amplitude + $\alpha$		
		12	8	4	12	8	4	12	8	4
gaus1, gaus2	98.3	98.3	98.4	94.9	98.3	98.3	98.2	98.3	98.4	94.6
gaus1, mexh	99.2	99.2	99.1	96.7	99.2	99.1	98.9	99.2	99.1	96.3
db1, db2	68.5	68.6	69.4	61.3	68.5	68.4	66.8	68.6	69.3	58.9
bior1, bior2	86.9	86.9	87.3	73.8	86.9	87.2	90.9	86.9	87.5	78.6
rbio1, rbio2	98.3	98.3	98.3	92.7	98.3	98.3	98.3	98.3	98.3	93.5

Table 3.6: RMS error (%) for scales from 1 to 64

Bases	Quantization (bits)									
	Ideal	Amplitude			$\alpha$			Amplitude + $\alpha$		
		12	8	4	12	8	4	12	8	4
gaus1, gaus2	1.064	1.057	0.996	5.859	1.063	1.057	1.157	1.056	0.992	5.773
gaus1, mexh	0.726	0.725	0.783	6.001	0.727	0.733	0.909	0.726	0.816	6.109
db1, db2	5.147	5.134	4.969	5.951	5.147	5.154	5.325	5.134	4.977	6.121
bior1, bior2	3.314	3.308	3.159	4.702	3.312	3.284	2.815	3.306	3.129	4.488
rbio1, rbio2	1.075	1.071	1.039	5.576	1.075	1.062	1.049	1.070	1.025	5.560

scales, and the quantization resolution. The correlation coefficient indicates how much of the original signal's morphology is preserved, which is primarily due to the Lipschitz coefficient estimation, and the RMS error reflects mostly the accuracy in determining the critical points' positions and amplitudes. It is clear from both metrics, though, that the best results occur for the gaussian pairs of wavelet bases, and the worst results are observed for the Daubechies pair. This is expected, since the gaussian bases hold a higher correlation with the test signal than the Daubechies and biorthogonal wavelet bases, and the wavelet transform is also a measure of correlation between the analyzed signal and the wavelet basis function. A consequence of this is that the Daubechies and biorthogonal wavelets would probably yield better results if the input signal had sharp transitions, for example, as an ECG signal. This is shown in a previous work [8]. It is worth noting that the results obtained with the reverse biorthogonal bases present higher correlation coefficients and smaller RMS error than the tests for the Daubechies and biorthogonal bases, which was not expected.

When comparing the characteristics of the RMS error tables and the correlation coefficient tables, it is noticeable that the difference between results for the error are greater than for the correlation coefficient. This indicates that there is little change in the Lipschitz coefficient estimation error for all cases, because it shows that, even if there is error in the amplitudes of the critical points, the morphology of the signal is well described. This is evidenced by evaluating the quantized  $\alpha$  column in Tables 3.4 and 3.6: for 12- and 8-bit resolutions, the results are practically the same, or very close. For a resolution of 4 bits, the error increases, but not so much as it does for the cases when the amplitude is quantized. In other words, the Lipschitz coefficient can be quantized with only 4 bits and yield medium to high quality signal recovery.

Regarding the choice of scales, the expected result is that the smaller the scale, the better the local maxima and minima detection will be, i.e. there should be smaller error for the set of scales from 1 to 64. Also, in this set, there are more scales than in the other one (from 16 to 64), what would give more information to estimate the Lipschitz exponent and result in higher correlation coefficients. However, the results vary for both metrics, and this difference is not always significant. This is not the expected result, but it can be a sign that, for this signal and these bases, the critical points localization for scale 64 is already good enough, with no gain when we try to increase the precision in this process. However, the increase in the RMS error metric when we zoom in on a smaller scale is not expected.

According to the data in the tables, the worst result occurs when the pair of Daubechies bases are selected with the second set of scales (from 1 to 64). The reconstruction results for this configuration are shown in Figures 3.14 and 3.15, respectively for the ideal and the quantized cases. The analysis of these figures shows that the high RMS error and low correlation coefficients are due to errors in the first and last portions of the signal. This can happen because, for these portions, the Lipschitz exponent is estimated with only one half of the information, since the local minima are the first and last points of the signal. In this

case, the error in the Lipschitz coefficient estimation resulted in not only a not so precise approximation for the polynomial, but also in a change of concavity. It is expected that in a real signal this “edge effect” might show up again, but not affect the following portions of the signal, as occurs with the middle portions of the signal in Figures 3.14 and 3.15.

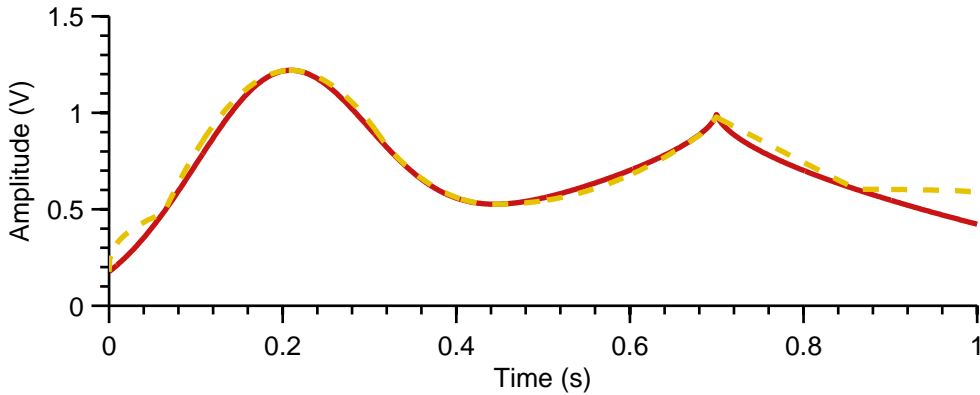
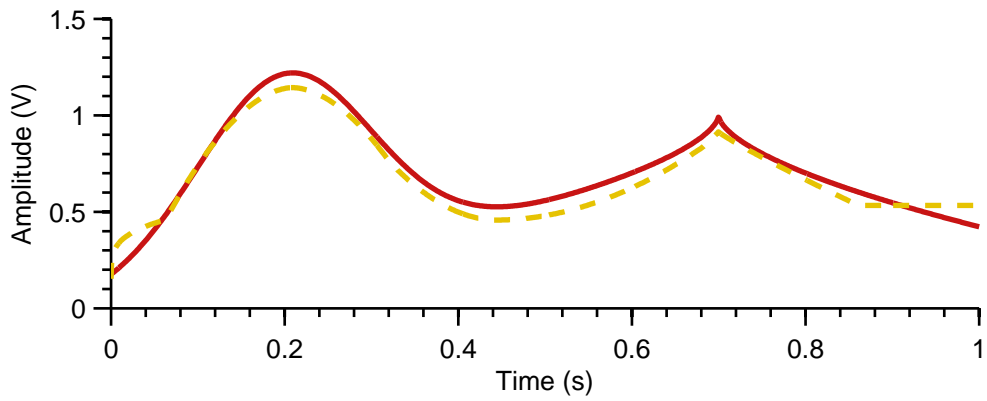


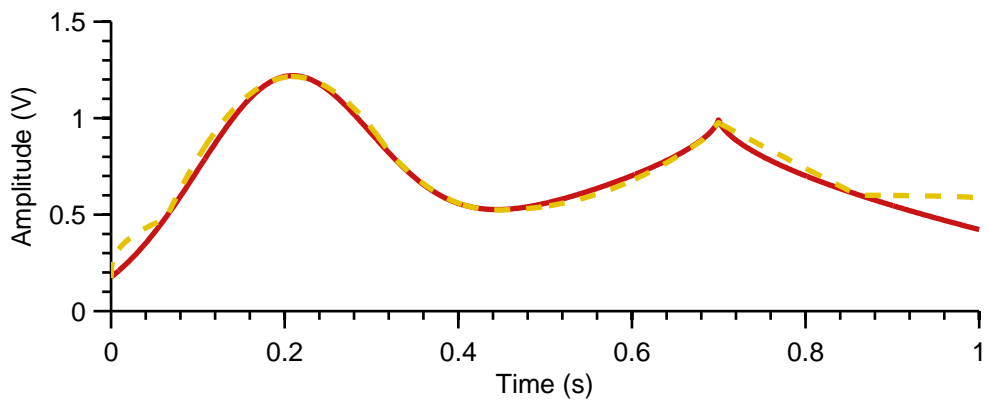
Figure 3.14: Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for db1 and db2 wavelet bases, set of scales from 1 to 64, and ideal (not quantized) values.

### 3.4.1 Comparison with a standard ADC

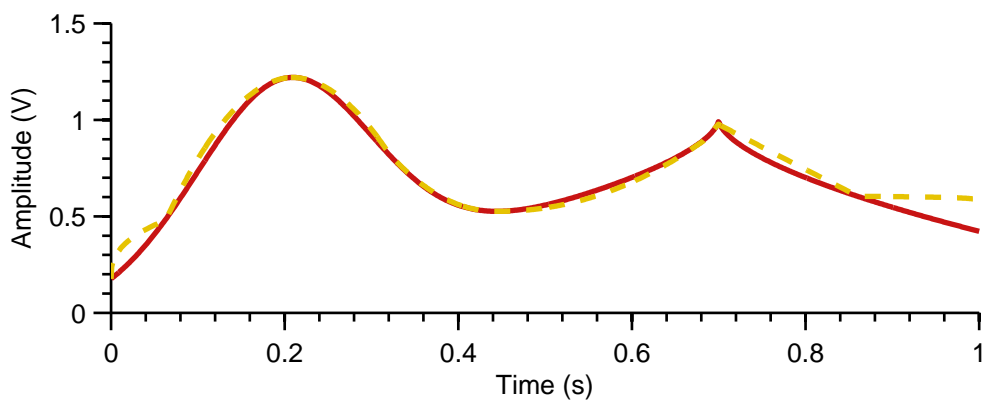
In order to compare the results with those of a standard ADC, a simple Simulink model was used. The standard ADC was modeled by passing the input signal through an ideal quantizer block and altering its parameters (quantization interval and sample time), and then measuring the RMS error of the output. Figure 3.16 shows the results in a graph that compares the number of bits,  $n$ , the RMS reconstruction error, and the sampling rate (in samples per second - sps) of the standard ADC and one example of the proposed wavelet-based ADC (WT ADC) for the test signal. From the previous analysis, the test signal was sampled at nine identified critical points, and the Lipschitz exponent, at five maxima or minima, resulting in 14 samples. Since the signal’s duration is of 1 second, this yields 14 sps. For the set that was evaluated first, that is, gaussian bases and scales from 16 to 64, with both the amplitude and the Lipschitz exponent quantized with 12 bits, the RMS error is  $e_{RMS} = 0.00855$ . If, however, we choose to quantize the Lipschitz exponent with 4 bits and the amplitude with 8, the RMS error would increase, but the number of bits/sample would decrease significantly as well.



(a) 4-bit resolution



(b) 8-bit resolution



(c) 12-bit resolution

Figure 3.15: Input signal (red, solid line) and reconstructed signal (yellow, dashed line) for db1 and db2 wavelet bases, set of scales from 1 to 64, and quantized values.

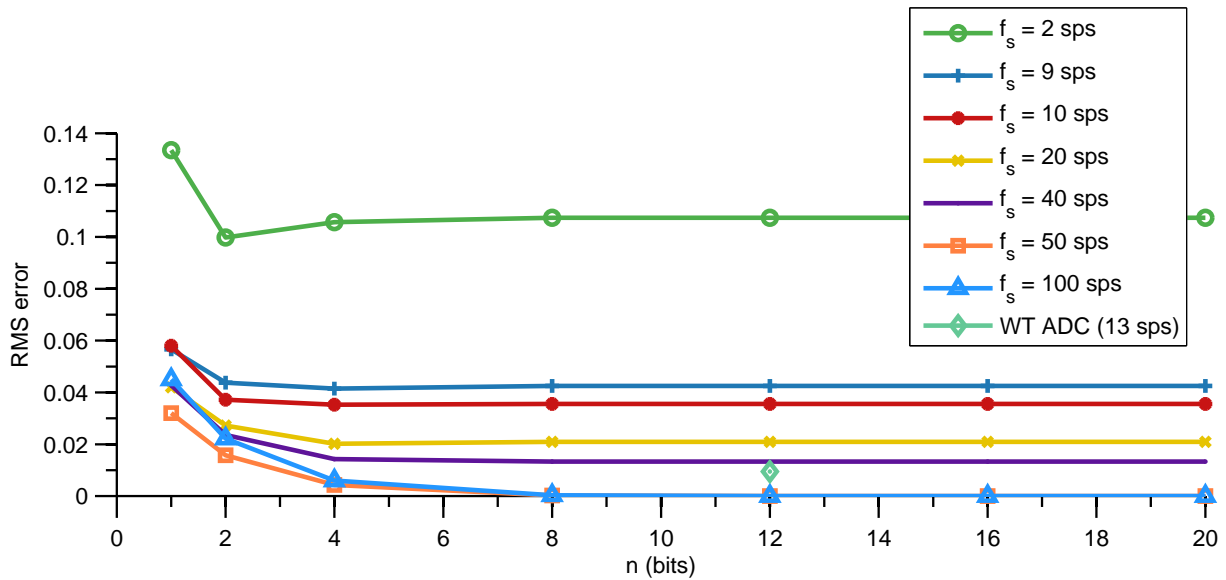


Figure 3.16: Number of bits x RMS error x sampling rate

### 3.5 Summary of the chapter

This chapter presented the system-level development of the wavelet-based analog-to-digital converter. The first section showed the proposed sampling and reconstruction algorithms. Then, the experimental tests and results were presented, which showed that the input signal could be recovered with little loss of information with only 13 samples of the input signal. Also, a high resolution can be achieved with not so high resolution quantization of the amplitude and Lipschitz exponent parameters. The next step is to implement this ADC in circuit-level, which is the object of the next chapter.

# Chapter 4

## Circuit-Level Development

In this chapter, the project is studied at circuit-level, starting with the proposed circuit block diagram and the theory of how each block can be implemented. The following sections describe the tests and their results. The last section contains a discussion of the results obtained in the project, both at system- and circuit-level.

### 4.1 Circuit block diagram

Based on the algorithm in Fig. 3.1, a circuit block diagram is proposed, as shown in Fig. 4.1.

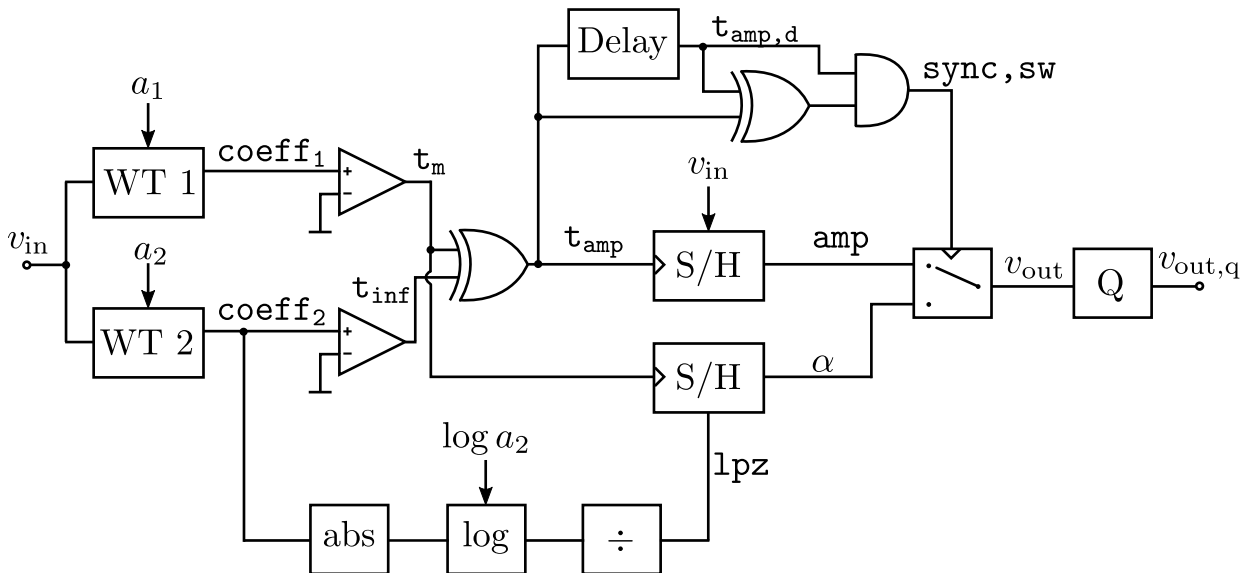


Figure 4.1: Wavelet-based ADC block diagram.

The WT of first and second orders of the input signal  $v_{in}$  can be implemented by analog wavelet filters with the method in Chapter 2. To localize the critical points, the coefficients line  $coeff_1$  and  $coeff_2$ , which are the wavelet coefficients line for scales  $a_1$  and  $a_2$ , go

through a comparator, implementing the zero-crossing detection. This outputs the signals  $\mathbf{t}_m$  and  $\mathbf{t}_{inf}$ , which have transitions at the instants when local maxima or minima occur, and when inflection points are detected. The amplitudes will be sampled at all these transitions, so the amplitude sampling is represented by a sample-and-hold block controlled by  $\mathbf{t}_{amp} = \mathbf{t}_m \oplus \mathbf{t}_{inf}$ .

The Lipschitz coefficient ( $\alpha$ ) estimation is based on Equation 2.22, and is implemented by the absolute value block ('abs') followed by the logarithm block ('log'), and then the division block. The output of this chain is the signal  $\mathbf{lpz}$ , which contains the Lipschitz exponent estimated at all points of  $v_{in}$ . Since the Lipschitz exponent is only sampled at local maxima and minima,  $\alpha$  is obtained by the sample-and-hold block controlled by  $\mathbf{t}_m$ . The mathematical operations of absolute value and division can be implemented with translinear circuits working in current mode, the logarithm can be obtained by using MOS transistors operating in weak inversion [5].

This ends the sampling algorithm implementation in circuit-level. Since the next step in an ADC is quantization, and in this project we would like to minimize power consumption, we made a choice to propose a circuit block diagram using only one quantizer, represented by the block Q. In order to do this, the signals for sampled amplitudes ( $\mathbf{amp}$ ) and for sampled Lipschitz coefficients ( $\alpha$ ) must be comprised together, without loss of information. One solution for this is to apply a switch controlled by a synchronization signal,  $\mathbf{sync}, \mathbf{sw}$ . At local maxima and minima points, both the amplitude and the Lipschitz exponent are sampled. The signal  $\mathbf{sync}, \mathbf{sw}$  contains pulses at these points, when  $\alpha$  should be selected by the switch. After the pulse, the switch goes back to its prior position, selecting  $\mathbf{amp}$  and collecting the amplitude at the maximum or minimum point and at the subsequent inflection point. The synchronization signal  $\mathbf{sync}, \mathbf{sw}$  is generated based on  $\mathbf{t}_{amp}$  and its delayed version  $\mathbf{t}_{amp,d}$ :

$$\mathbf{sync}, \mathbf{sw} = (\mathbf{t}_{amp} \oplus \mathbf{t}_{amp,d}) \cdot \mathbf{t}_{amp,d} \quad (4.1)$$

This process outputs a single signal ( $v_{out}$ ), which is then quantized ( $v_{out,q}$ ) and must be encoded. There are many suitable options for encoding, including the use of binary coding, and these are not thoroughly discussed in this project.

## 4.2 Methodology

The tests for the circuit were implemented in CADENCE VIRTUOSO ADE, with XFAB XC018 technology. The verification methodology for the proposed circuit was to test each block individually, starting with ideal blocks. After each block is completed, the results are collected and are processed offline in MATLAB. These results are then compared to the system results. In this recursive method, some circuit results evidenced the need for

modifications to the proposed system algorithm, as will be shown in the following section. Because of this and of other issues and project constraints, the circuit analysis could not be completed in due time and some parts are left as future work.

## 4.3 Results

Analyzing the circuit block diagram in Fig. 4.1, we see that the blocks that implement the wavelet transform and the ones concerning the estimation of the Lipschitz exponent are the critical components of the circuit. The first blocks to be implemented are the WT ones.

### 4.3.1 Analog wavelet filter block

The WT is implemented with an analog filter. In order to do so, we must define the orthonormal state-space representation of the basis we are interested in, which is obtained by its transfer function. The 6<sup>th</sup> order transfer function for the first derivative of the gaussian function, obtained with  $L_2$  approximation, is available in the paper from Karel [7], and is expressed as:

$$H_{gaus1}(s) = \frac{-0.08946s^5 - 0.1683s^4 - 8.326s^3 + 6.642s^2 - 139.6s}{s^6 + 5.927s^5 + 30.52s^4 + 83.11s^3 + 163.6s^2 + 176.6s + 93.29} \quad (4.2)$$

One way to obtain the second derivative of a gaussian would be to multiply the transfer function in Equation 4.2 by  $s$ , yielding:

$$H_{gaus2}(s) = \frac{-0.08946s^6 - 0.1683s^5 - 8.326s^4 + 6.642s^3 - 139.6s^2}{s^6 + 5.927s^5 + 30.52s^4 + 83.11s^3 + 163.6s^2 + 176.6s + 93.29} \quad (4.3)$$

However, since the numerator in Equation 4.2 is of 5<sup>th</sup> order, the transfer function for the second derivative of the gaussian function, shown in Equation 4.3, is a transfer function with the same number of zeroes and poles. Even though this would still be a proper function, the numerator coefficient of highest order is significantly smaller than the others, and so the transfer function can be approximated to:

$$H'_{gaus1}(s) = \frac{-0.1683s^4 - 8.326s^3 + 6.642s^2 - 139.6s}{s^6 + 5.927s^5 + 30.52s^4 + 83.11s^3 + 163.6s^2 + 176.6s + 93.29} \quad (4.4)$$

Hence, the transfer function for the second derivative of the gaussian function is expressed as is in Equation 4.5.

$$H'_{gaus2}(s) = \frac{-0.1683s^5 - 8.326s^4 + 6.642s^3 - 139.6s^2}{s^6 + 5.927s^5 + 30.52s^4 + 83.11s^3 + 163.6s^2 + 176.6s + 93.29} \quad (4.5)$$



The impulse responses for the transfer functions in Equations 4.2 and 4.4 are shown in figure 4.2. The impulse response is very close to the theoretical response expected for the first derivative of the gaussian function. The effect of the approximation in the numerator is observed as a slightly more oscillatory behaviour in the start of the signal in Figure 4.2 (b) when compared to the signal in Figure 4.2(a).

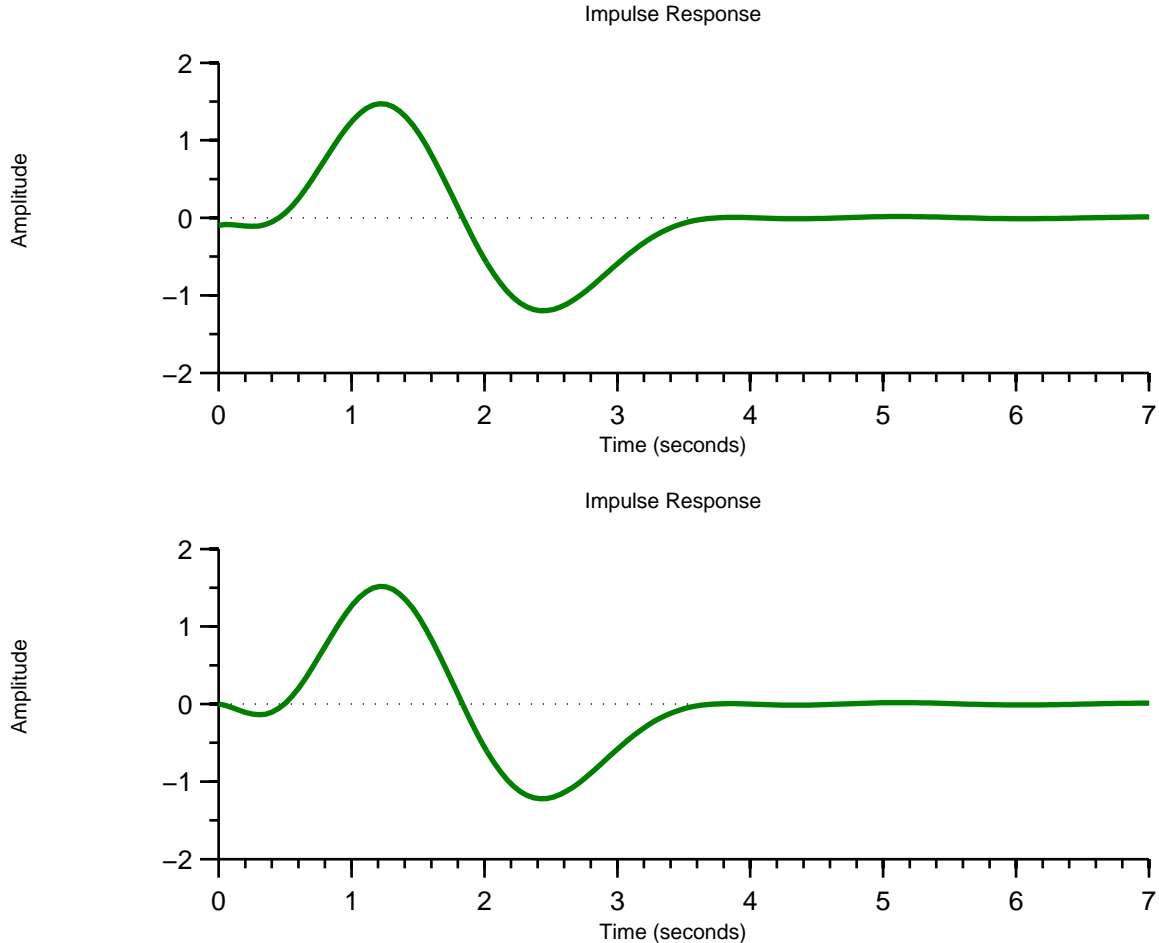


Figure 4.2: First derivative of a gaussian (gaus1) for sixth order  $L_2$  approximation for (a) original transfer function  $H_{gaus1}(s)$  (b) approximated transfer function  $H'_{gaus1}(s)$ .

The impulse responses for the second derivative of the gaussian function, the transfer functions in Equations 4.3 and 4.5, are shown in Figure 4.3. The impulse responses resemble the original gaus2 function, however, there is an asymmetry in the amplitude of both peaks of the impulse response, in both cases of Figures 4.3(a) and (b). Then, this asymmetry is either a consequence of the  $L_2$  approximation or of the procedure of obtaining the second derivative by multiplying the first derivative transfer function by  $s$ . As observed for the cases of gaus1 in Figure 4.2, the approximation in the order of the numerator in the transfer function adds a small error in the start of the signal and a slightly more oscillatory behaviour to the transfer function impulse response.

Due to the fact that the differences between the original transfer functions,  $H_{gaus1}(s)$

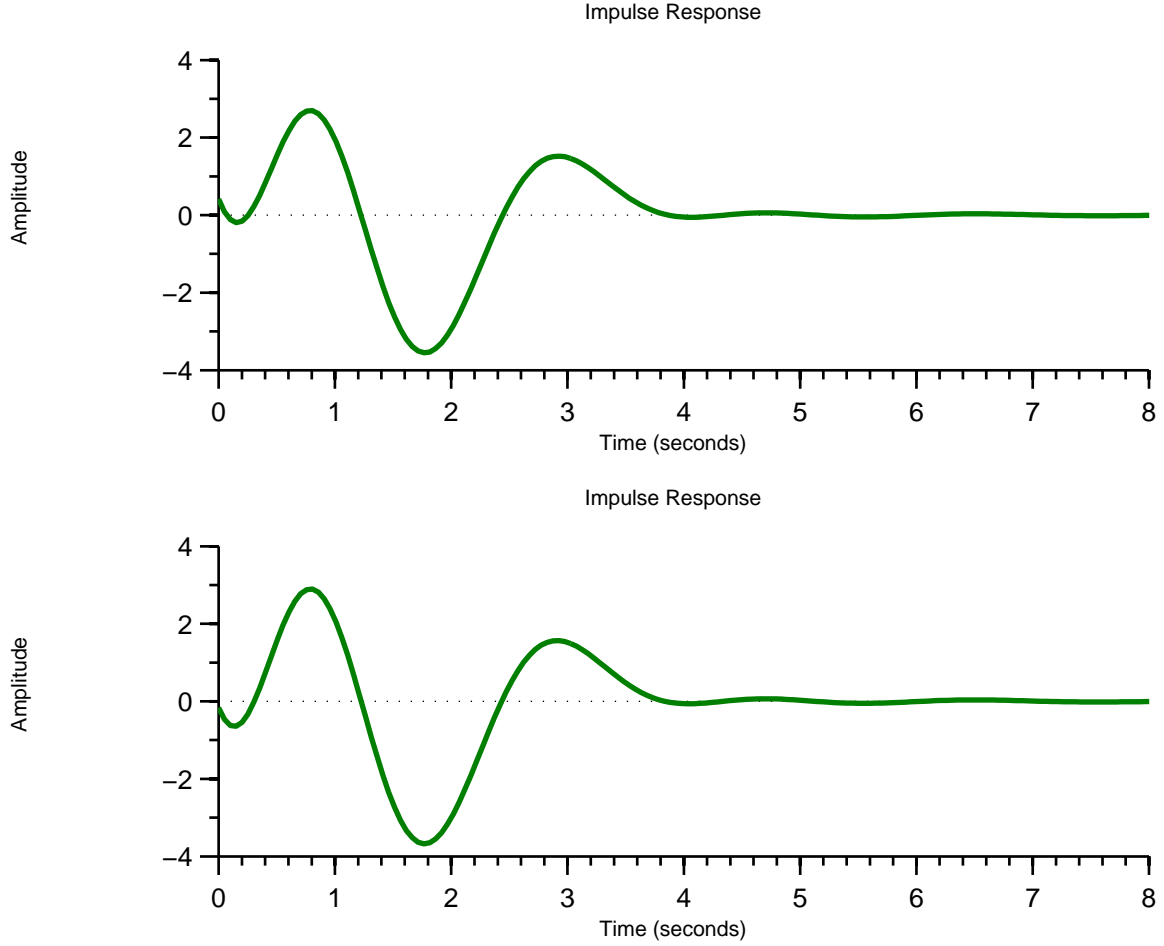


Figure 4.3: Second derivative of a gaussian (gaus2) for sixth order  $L_2$  approximation for (a) original transfer function  $H_{gaus2}(s)$  (b) approximated transfer function  $H'_{gaus2}(s)$ .

and  $H_{gaus2}(s)$ , and their approximations,  $H'_{gaus1}(s)$  and  $H'_{gaus2}(s)$ , are very subtle, we chose to implement the filters with the approximated transfer functions. Having decided which transfer functions to apply, the next step is to obtain an orthonormal state-space representation of the transfer functions, and thus obtain the coefficients to implement the Gm-C filter. The orthonormal state-space representation for  $H'_{gaus1}(s)$  in Equation 4.4 is obtained with a MATLAB function (available in the Appendix II). The resulting matrices are:

$$A = \begin{bmatrix} 0 & 1.185 & 0 & 0 & 0 & 0 \\ -1.185 & 0 & 1.637 & 0 & 0 & 0 \\ 0 & -1.637 & 0 & 2.007 & 0 & 0 \\ 0 & 0 & -2.007 & 0 & 2.431 & 0 \\ 0 & 0 & 0 & -2.431 & 0 & 4.062 \\ 0 & 0 & 0 & 0 & -4.062 & -5.927 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1.374 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.1948 & -2.424 & 0.2941 & -0.5214 & -0.03016 & -0.06514 \end{bmatrix}$$

$$D = [0]$$

The orthonormal state-space representation for the transfer function  $H'_{\text{gaus2}}(s)$ , in Equation 4.5, is obtained by the same procedure as:

$$A_{\text{gaus2}} = A = \begin{bmatrix} 0 & 1.185 & 0 & 0 & 0 & 0 \\ -1.185 & 0 & 1.637 & 0 & 0 & 0 \\ 0 & -1.637 & 0 & 2.007 & 0 & 0 \\ 0 & 0 & -2.007 & 0 & 2.431 & 0 \\ 0 & 0 & 0 & -2.431 & 0 & 4.062 \\ 0 & 0 & 0 & 0 & -4.062 & -5.927 \end{bmatrix} \quad B_{\text{gaus2}} = B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1.374 \end{bmatrix}$$

$$C_{\text{gaus2}} = \begin{bmatrix} -4.617 & -0.7125 & 7.609 & -0.6636 & -1.492 & -0.1225 \end{bmatrix}$$

$$D_{\text{gaus2}} = D = [0]$$

As expected, matrices A and B in the state-space representation for the second derivative of the gaussian function are the same as the ones in the representation for the first derivative, matrix D remains null, and only matrix C has different coefficients.

These matrices coefficients are the values of the transconductances of the Gm-C elements in the circuit schematic. Both WT-1 and WT-2 filters can be implemented in the same schematic by placing the matrices  $C_{\text{gaus1}}$  and  $C_{\text{gaus2}}$  in parallel, as shown in the schematic in Figure 4.4. The first schematic simulations were run with ideal elements available in the software library. The ideal transconductor block was implemented using a voltage-controlled current source (vccs block). The order of the transconductances is adjusted according to the capacitances and the operation frequency. The initial configuration was of  $C = 2pF$  and transconductances in the order of nA/V.

From this point on, altering the capacitances alters the operation bandwidth, which corresponds to altering the WT scale. The capacitance values that were chosen for the simulations were:  $C = 2, 8, 16, 32, 64$  and  $128pF$ . The testbench schematic for these simulations is the one shown in Figure 4.5, and consists in cascading the WT analog filters for each scale.

The system's impulse response is shown in Figure 4.6(a) (transient simulation result) and (b) (AC simulation result), for gaus1. For gaus2, the transient simulation results are shown in Figure 4.7(a) and the AC simulation results are shown in Figure 4.7(b). The impulse responses are obtained when a pulse of high amplitude and short duration is applied as an input signal. The expected results are versions of the waveforms shown in Figures 4.2(b) and 4.3(b), expanded (for larger scales) or compressed (for smaller scales). The frequency responses show how larger scales correspond to lower frequencies, and how smaller scales

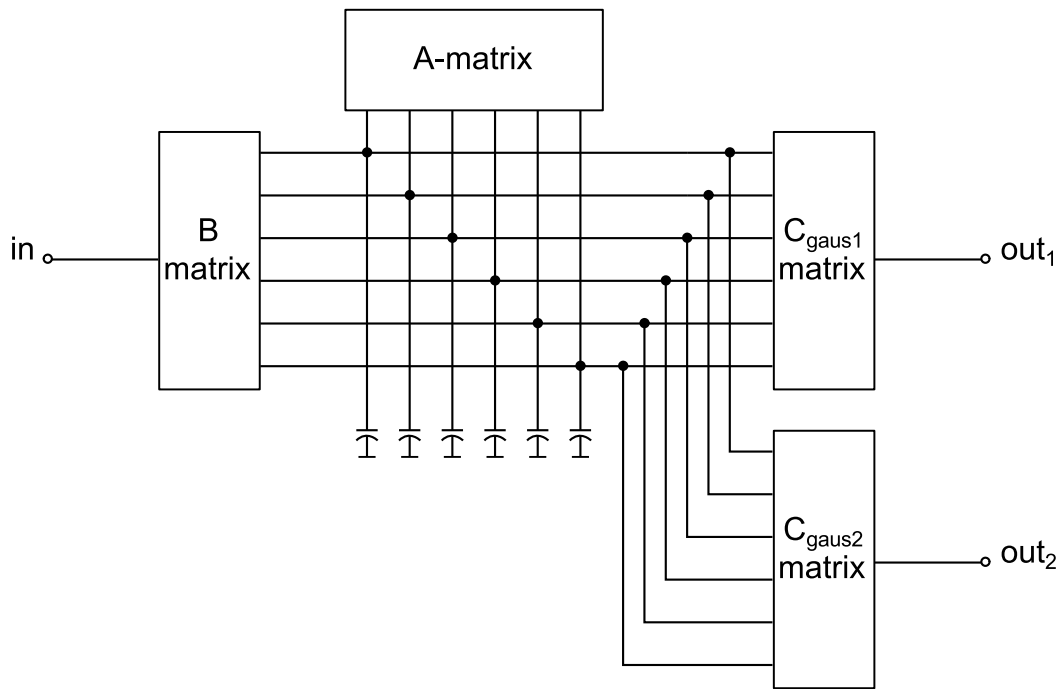


Figure 4.4: Circuit implementation of gaussian wavelet filters.

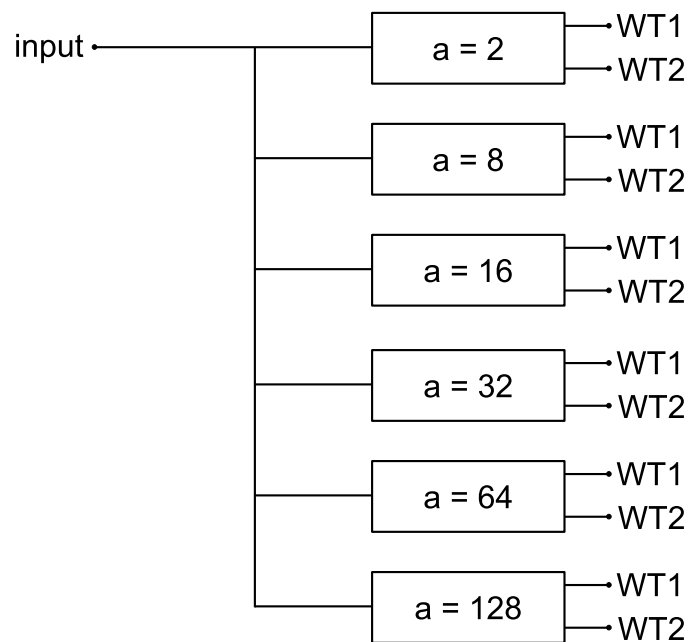
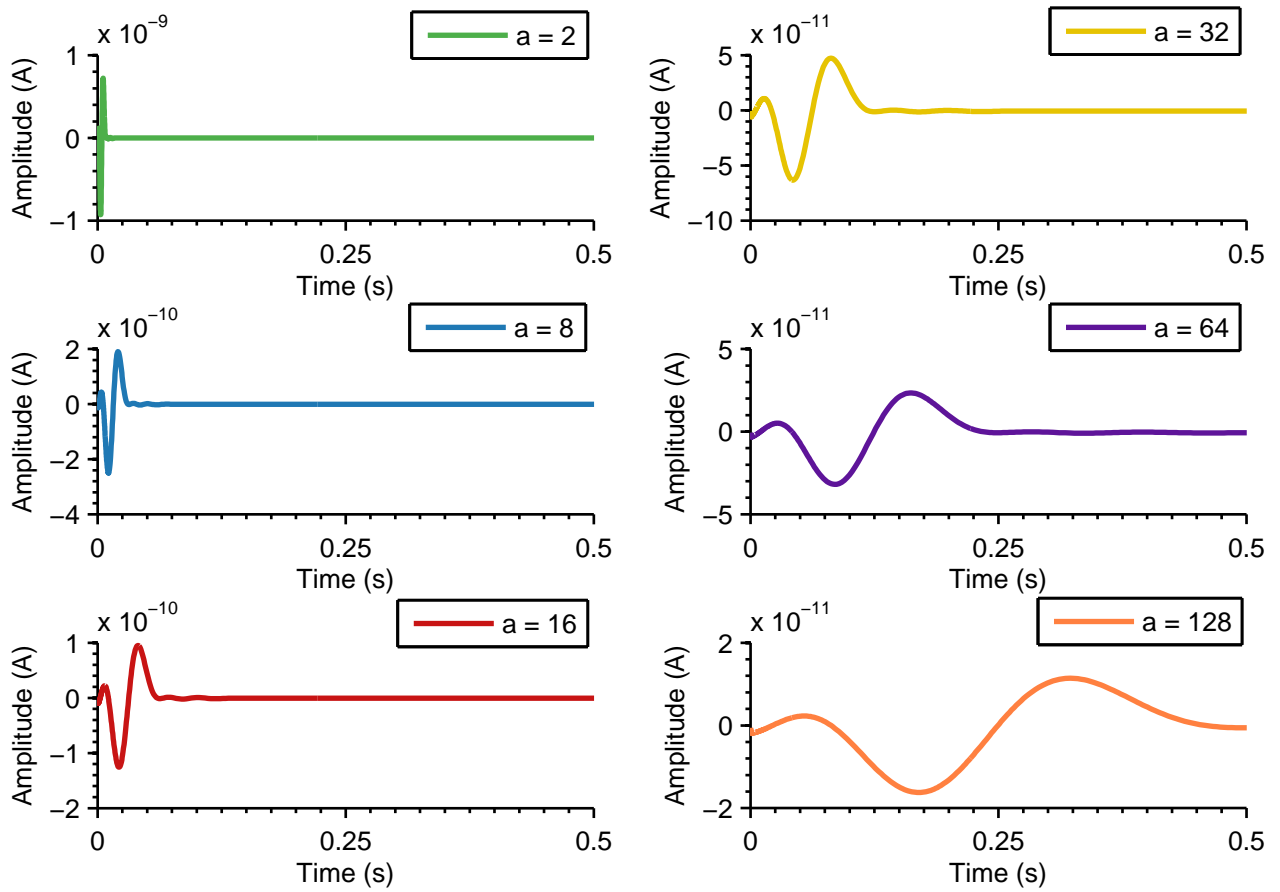


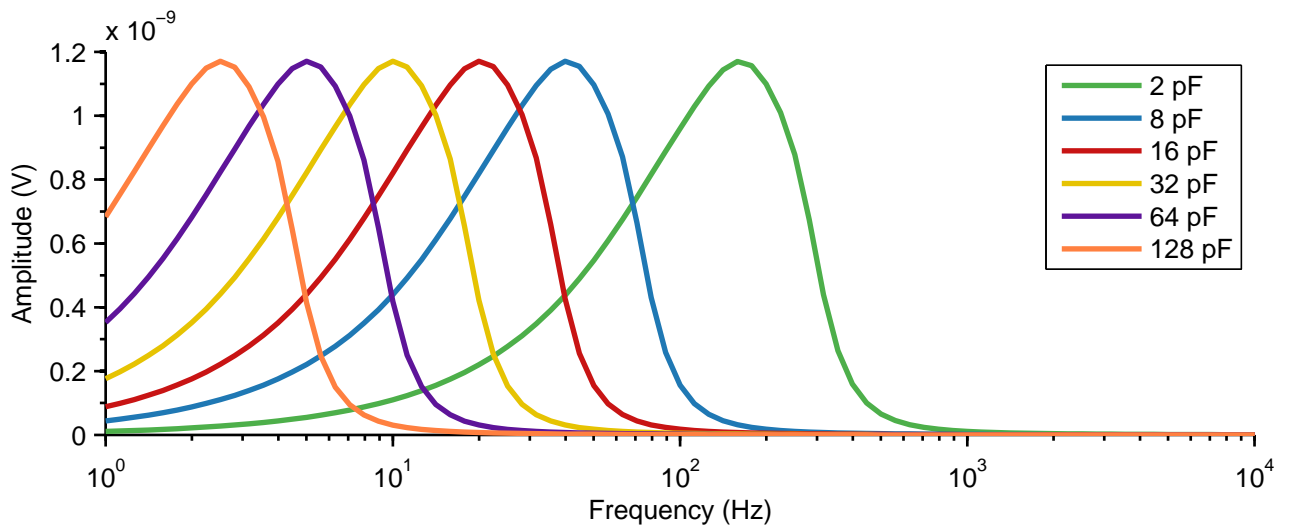
Figure 4.5: Testbench schematic for WT-1 and WT-2 gaussian filters.

correspond to higher frequencies. The impulse responses present the expected behaviour, with the symmetry error in the second derivative of the gaussian function being observed again.

Changing the input signal from an impulse to the test signal, the testbench outputs are the first- and second-orders wavelet transform coefficients lines for each scale they represent. Since there could not be established any direct mathematical relation to identify which



(a) Transient simulation results.

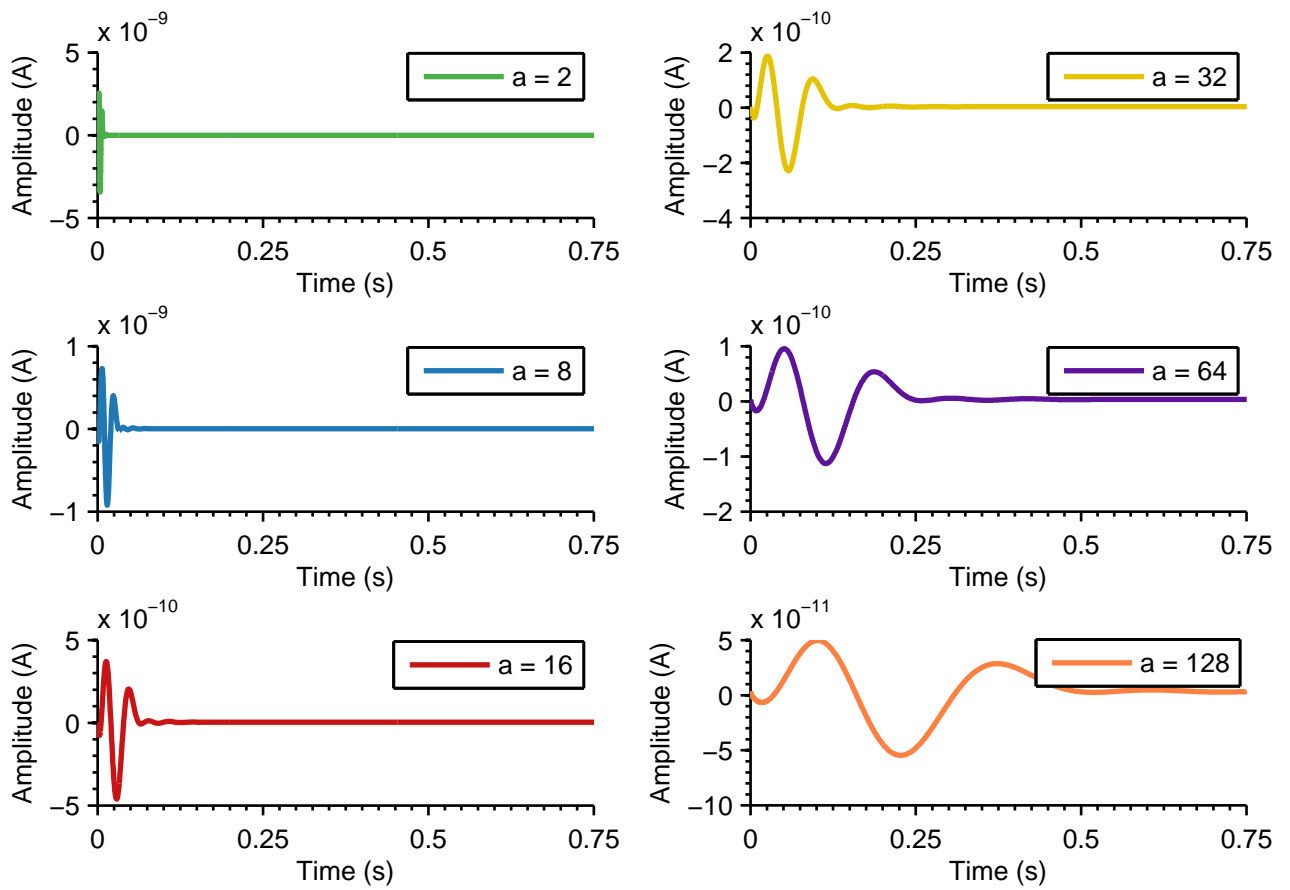


(b) AC simulation results.

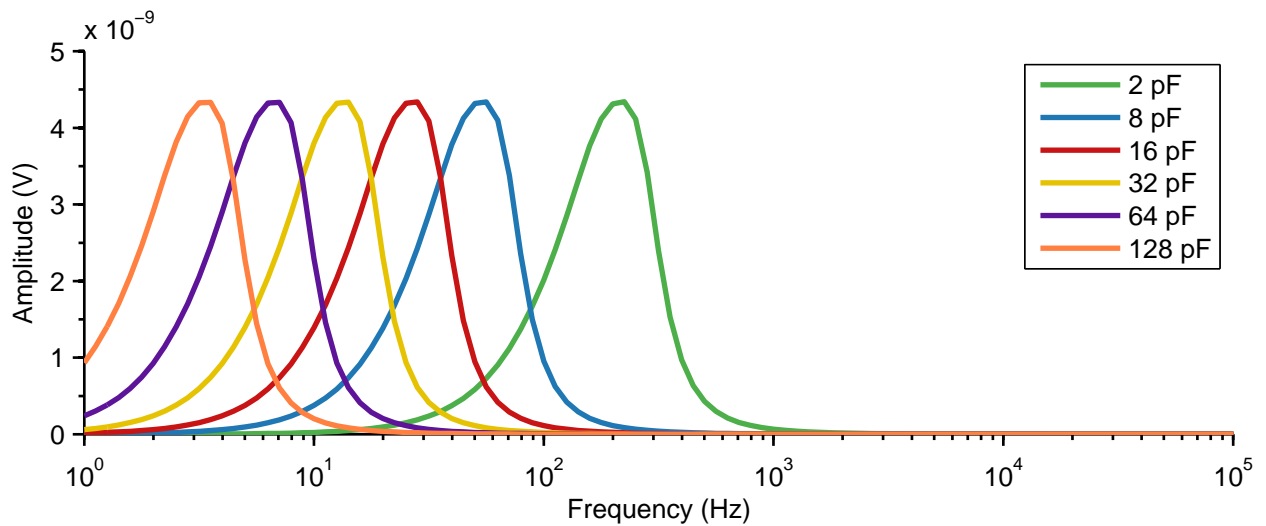
Figure 4.6: Impulse responses for gaus1.

frequencies correspond to which values of the scales used during the system simulation, the scales for the circuit results are identified by the values of the capacitances in pF.

Figure 4.8(a) shows the coefficients line for each scale/capacitance, while Figure 4.8(b)



(a) Transient simulation results.

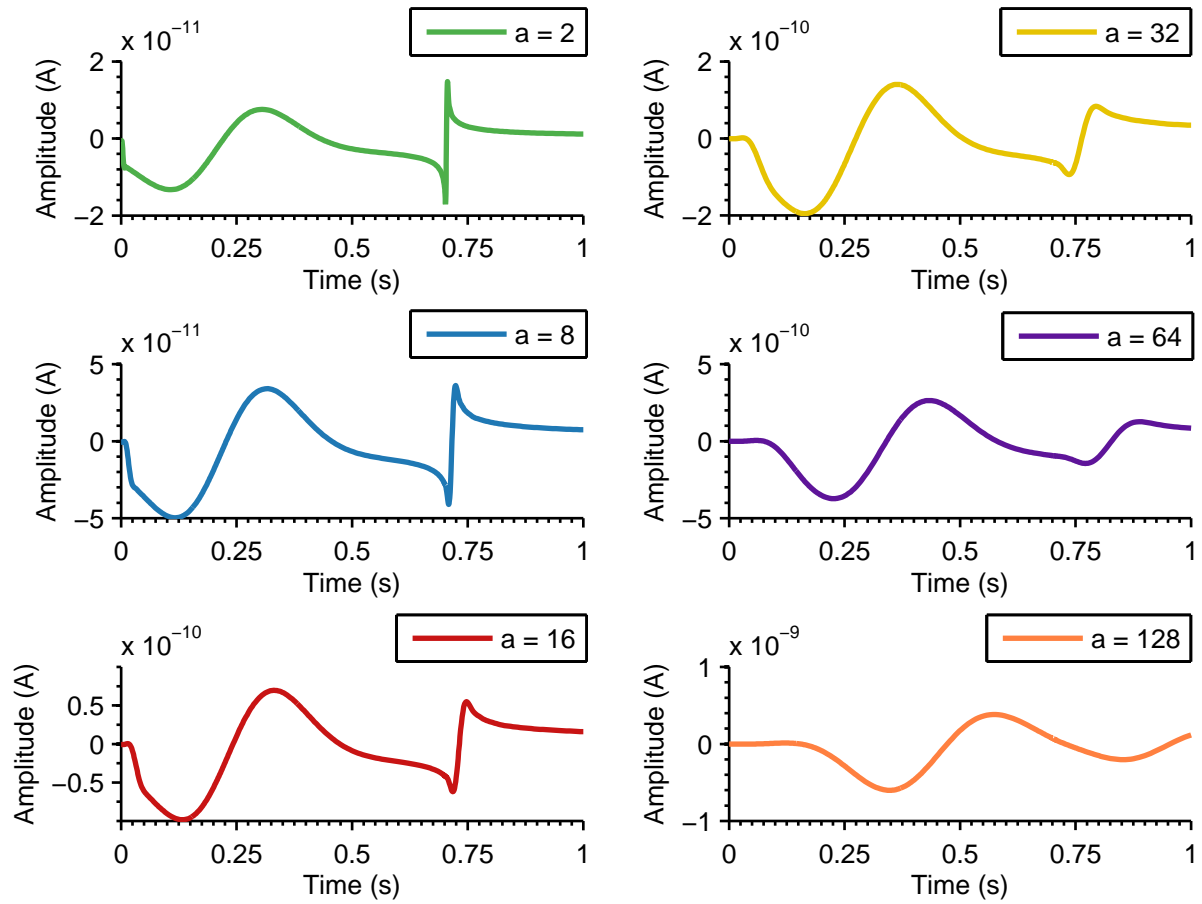


(b) AC simulation results.

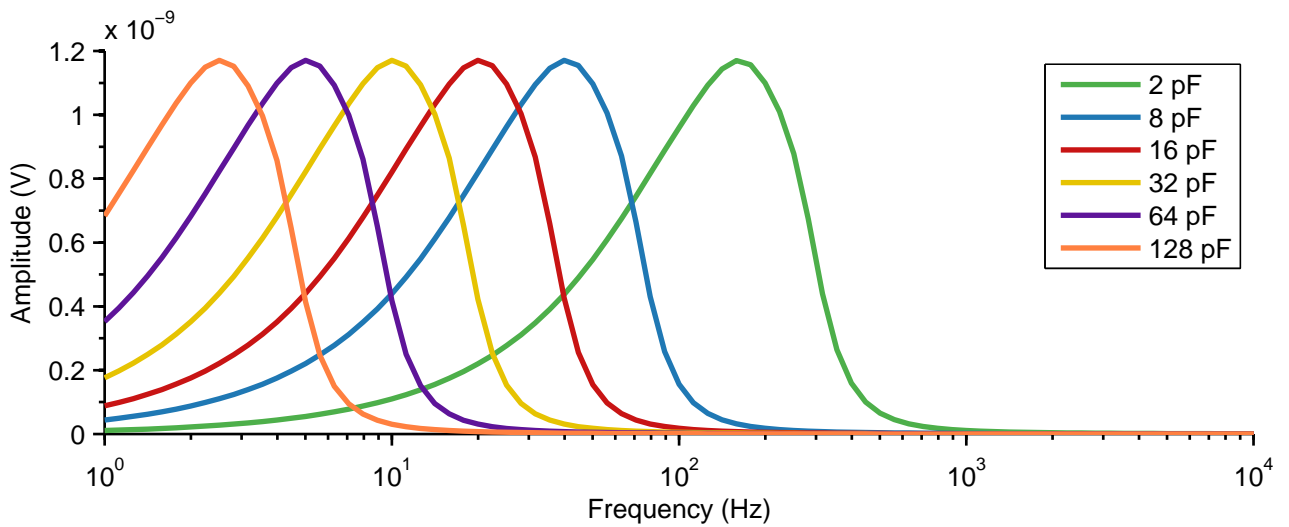
Figure 4.7: Impulse responses for gaus2.

shows the frequency response in each case for gaus1. It is worth noting that the zero-crossing of the coefficients line for the smallest scale,  $a = 2$  or  $C = 2\text{pF}$ , identifies the maxima and minima of the test signal approximately at the instants  $t = 0.2\text{s}$ ,  $t = 0.5\text{s}$ , and

$t = 0.7s$ . This is in conformity with the real values obtained from the system simulations, in Table 3.2:  $t = 0.211s$ ,  $t = 0.446s$ , and  $t = 0.700s$ .



(a) Coefficients lines for all scales.



(b) AC simulation results.

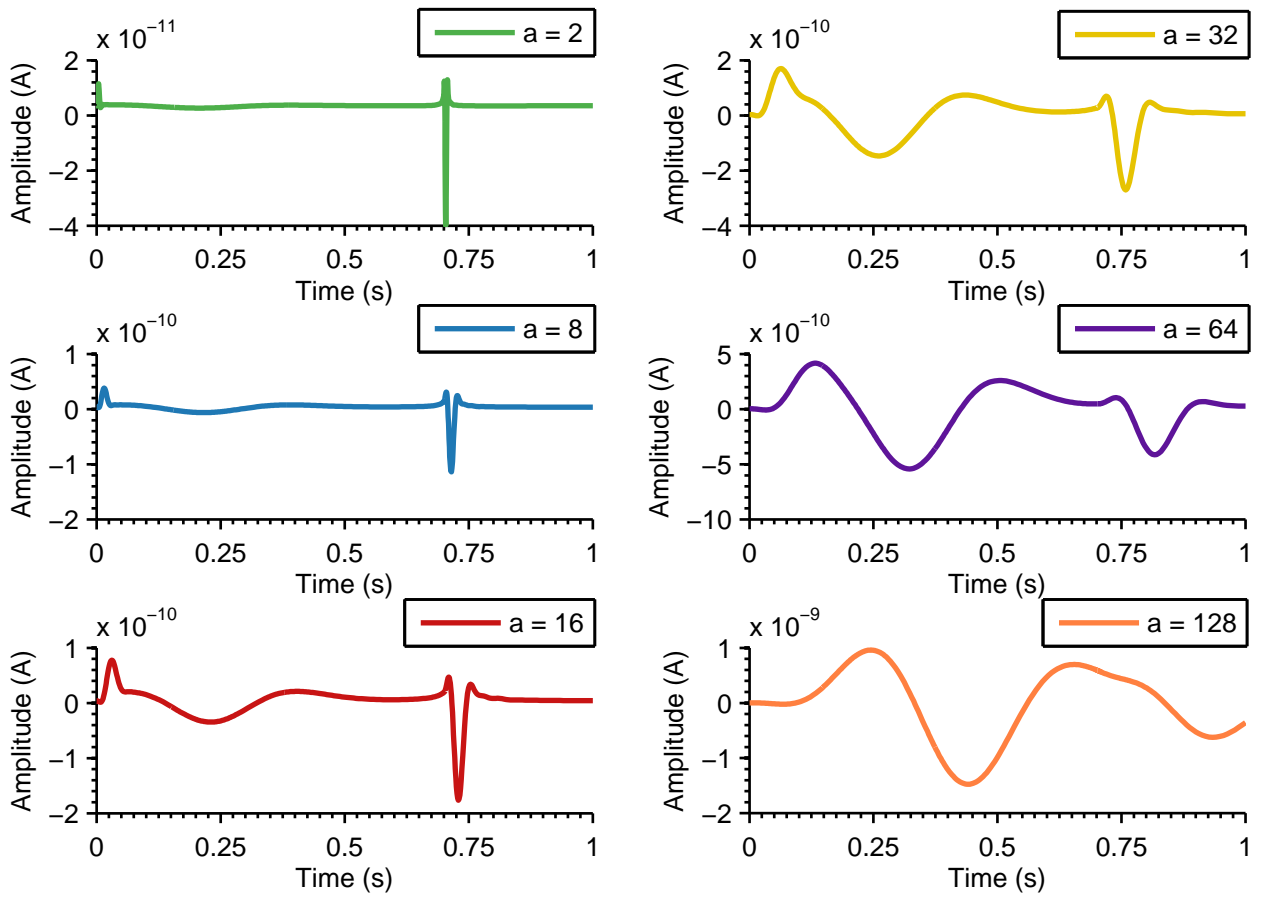
Figure 4.8: First-order wavelet transform of the test signal using analog wavelet filters with gaussian basis.

The coefficients lines for gaus2 are shown in Figure 4.9(a), and the frequency response is

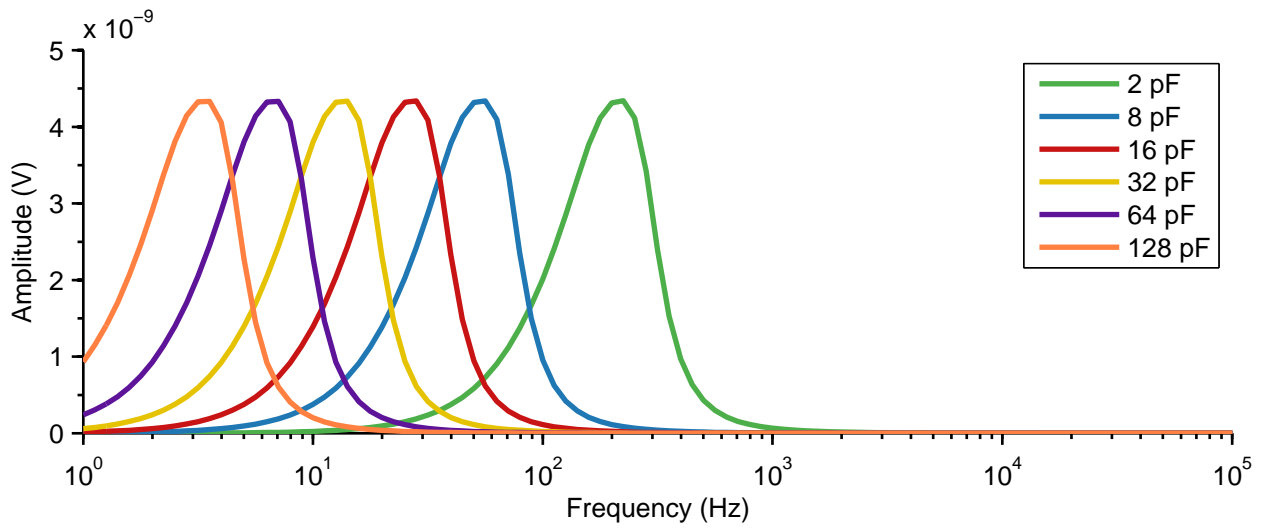
shown in Figure 4.9(b). Observe how the peaks in the coefficients lines happen at different instants in time. This delay was not observed in the system simulations and affects the algorithm for Lipschitz exponent estimation, since the estimated  $\alpha$  at a given instant, when a critical point of interest occurs, will not correspond to its real value. When exporting the coefficients line in Figure 4.9 to MATLAB and running the system-level tests, the Lipschitz coefficient at the local maxima and minima (including the first and last points of the signal) were estimated as: 0.5, 1.07, 1.6, 0.96, and 1.63. The real values obtained at system-level, according to the data in Table 3.2, are, however: 1.25, 1.77, 1.997, 0.60, and 1.02.

Since there were some flaws in the gaussian function approximation for the sixth order  $L_2$  approximation, the following subsection presents a second attempt to implement the gaussian wavelets in analog filters, applying a higher order approximation obtained with the Padé method.





(a) Coefficients lines for all scales.



(b) AC simulation results.

Figure 4.9: Second-order wavelet transform of the test signal using analog wavelet filters with gaussian basis.

### 4.3.1.1 A higher order approximation

In this subsection, the same tests from the previous subsection were repeated for a 10<sup>th</sup> order [6/8] Padé approximation, obtained from the book by Haddad and Serdijn [5]. The transfer function for gaus1, in this case, is the one in Equation 4.6:

$$H_{gaus1}(s) = \frac{-4.4820s^6 - 34.446s^5 + 444.32s^4 - 4708.1s^3 + 16977s^2 - 52829s + 3.6809}{-0.77104s^{10} - 15.627s^9 - 158.57s^8 + 1043.7s^7 - 4880.8s^6 - 16787s^5 - 42703s^4 - 78851s^3 - 100660s^2 - 79880s - 29823} \quad (4.6)$$

Multiplying Equation 4.6 by  $s$  yields the transfer function for gaus2:

$$H_{gaus2}(s) = \frac{-4.4820s^7 - 34.446s^6 + 444.32s^5 - 4708.1s^4 + 16977s^3 - 52829s^2 + 3.6809s}{-0.77104s^{10} - 15.627s^9 - 158.57s^8 + 1043.7s^7 - 4880.8s^6 - 16787s^5 - 42703s^4 - 78851s^3 - 100660s^2 - 79880s - 29823} \quad (4.7)$$

The impulse responses to the transfer functions in Equations 4.6 and 4.7 are shown in Figure 4.10. In this case, there was no need for approximations in the first derivative transfer function in order to produce the transfer function for the second derivative. Both impulse responses correspond to the theoretical expected behaviour of the gaussian functions. The asymmetry observed in the 6<sup>th</sup> order L<sub>2</sub> approximation for gaus2 is not observed for this approximation.

The orthogonal state-space representation for the gaus1 filter is:

$$A = \begin{bmatrix} 0 & 1.071 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.071 & 0 & 1.492 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.492 & 0 & 1.799 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.799 & 0 & 2.069 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.069 & 0 & 2.379 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.379 & 0 & 2.836 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2.836 & 0 & 3.641 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3.641 & 0 & 5.380 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5.380 & 0 & 11.78 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -11.78 & -20.27 \end{bmatrix}$$

$$B^T = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 2.540]$$

$$C_{gaus1} = [0.9739 \ 2.220 \ -1.216 \ 0.6866 \ -0.1700 \ 0.02687 \ 0.009916 \ 0 \ 0 \ 0]$$

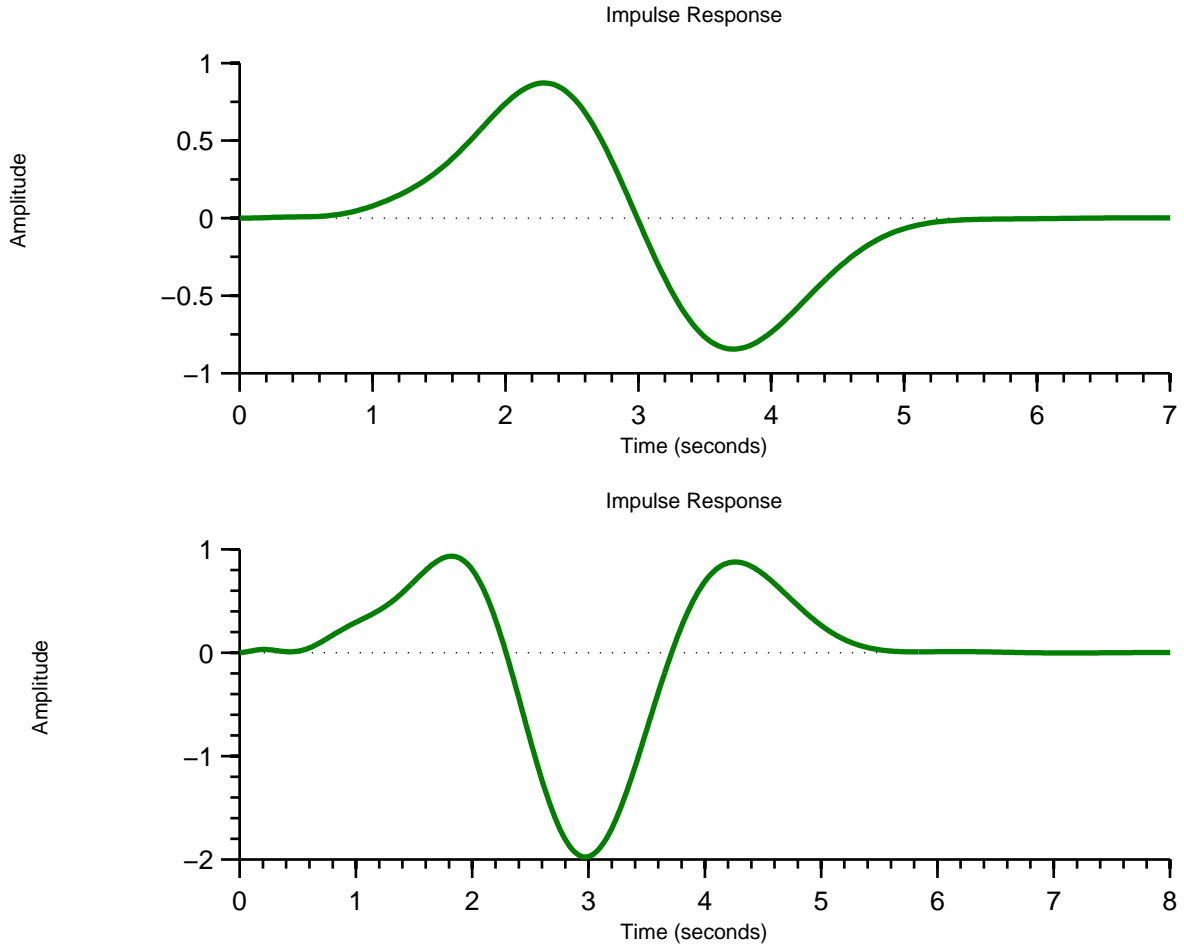


Figure 4.10: First (top) and second (bottom) derivatives of the gaussian function for 10-th order filter.

$$D = [0]$$

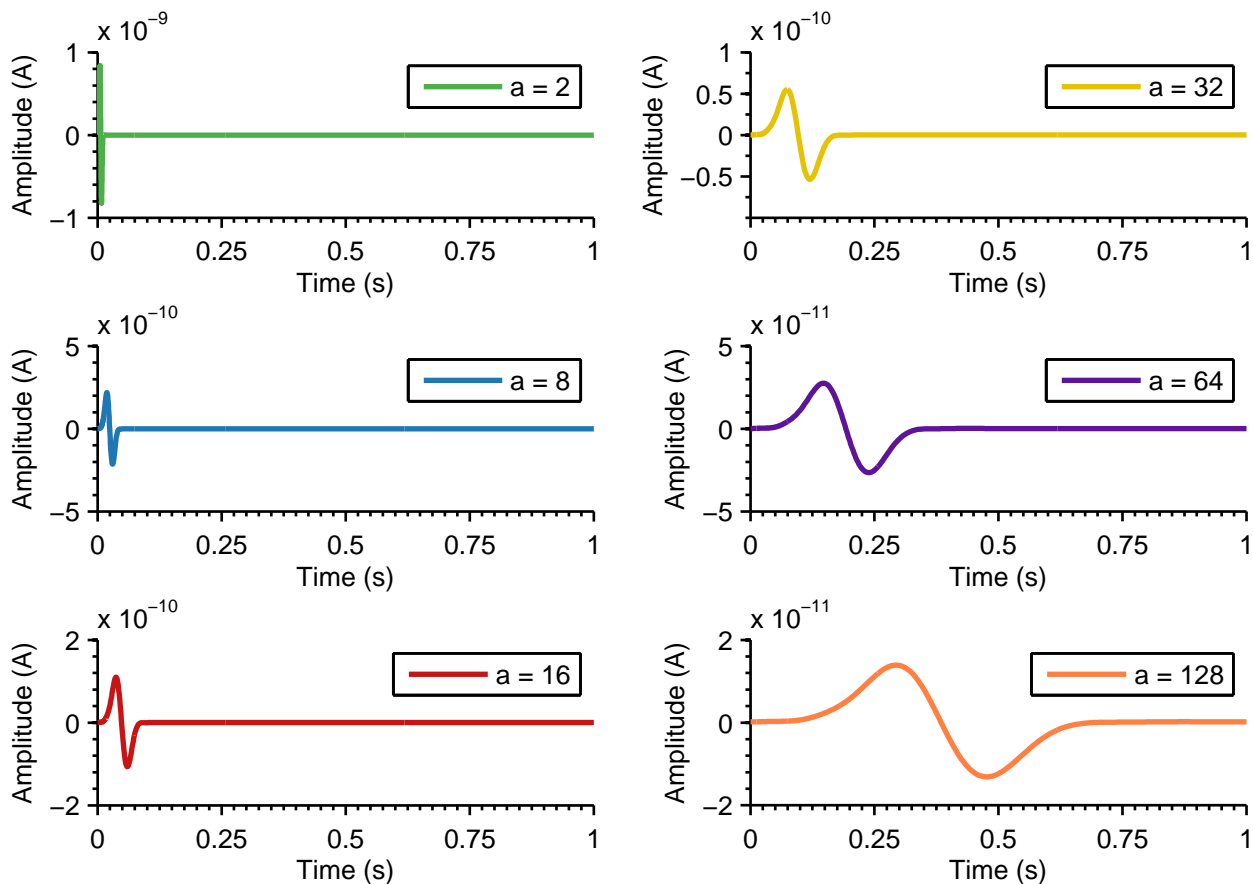
As expected for the second derivative, matrices A and B are the same as the ones for the first derivative, with only matrix C being modified:

$$C_{gaus2} = \begin{bmatrix} -2.378 & 2.857 & 2.077 & -1.835 & 1.357 & -0.4326 & 0.07621 & 0.0361 & 0 & 0 \end{bmatrix}$$

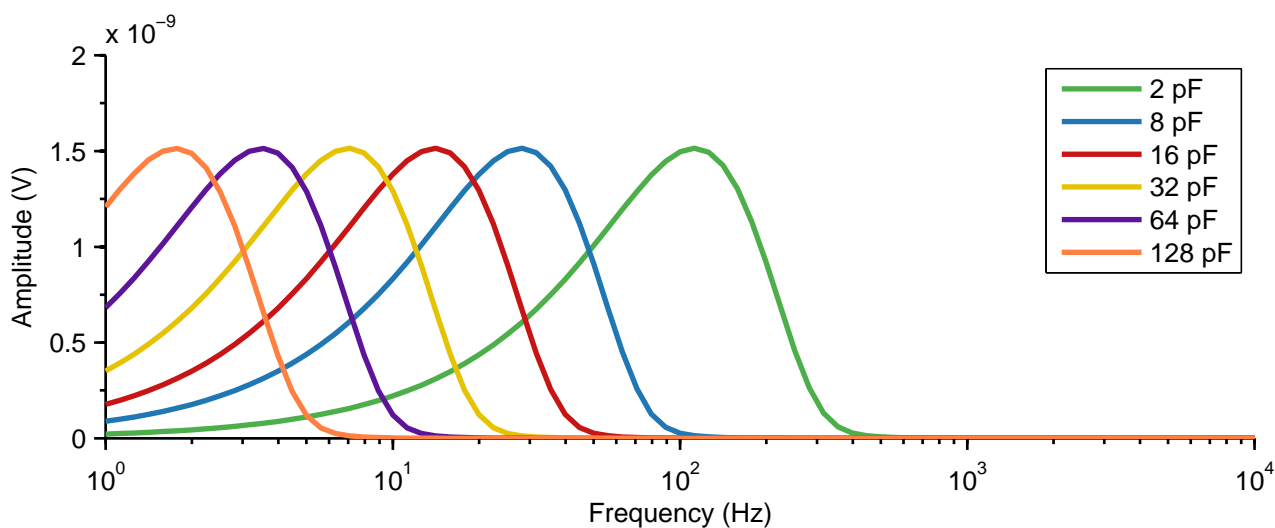
The circuit implementation of these filters follows the same method that was applied for the sixth-order filter in Figure 4.4. The value of the capacitances determine the WT filter bandwidth. Thus, different capacitances correspond to different frequencies, which correspond to different scales. The chosen values for the capacitances are the same as before: 2, 8, 16, 32, 64 and 128 pF, and the transconductances are in the order of  $nA/V$ .

Figure 4.11 shows the impulse responses for gaus1 at all tested scales in time and frequency domain obtained with the circuit implementation. Figure 4.12 shows the results of the same tests for gaus2. The first derivative of the gaussian function is clearly identified

in the impulse responses in Figure 4.11(a), as well as the second derivative, *gaus2*, in the impulse responses in Figure 4.12(a). The frequency responses for each case are shown in Figure 4.11(b) and Figure 4.12(b).

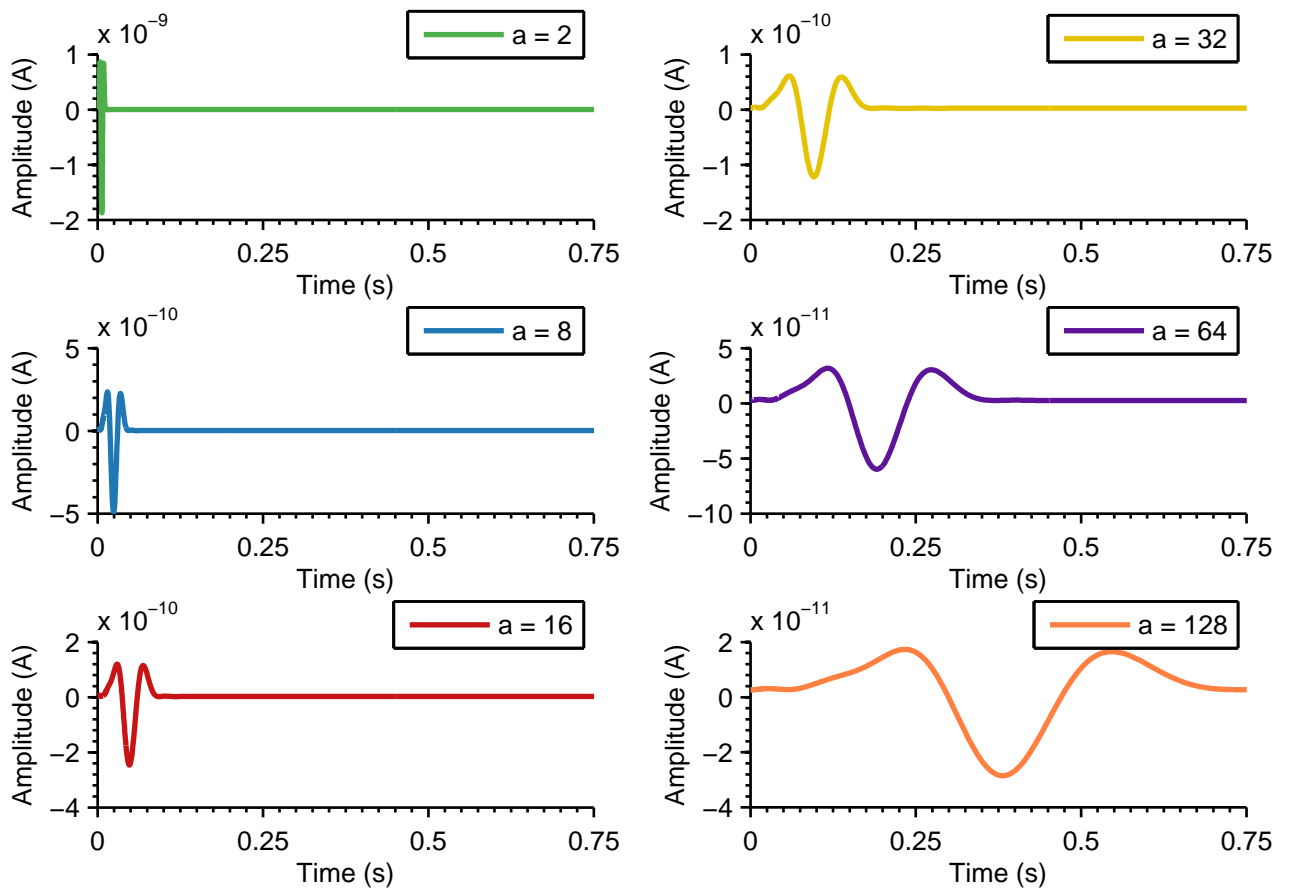


(a) Transient simulation results.

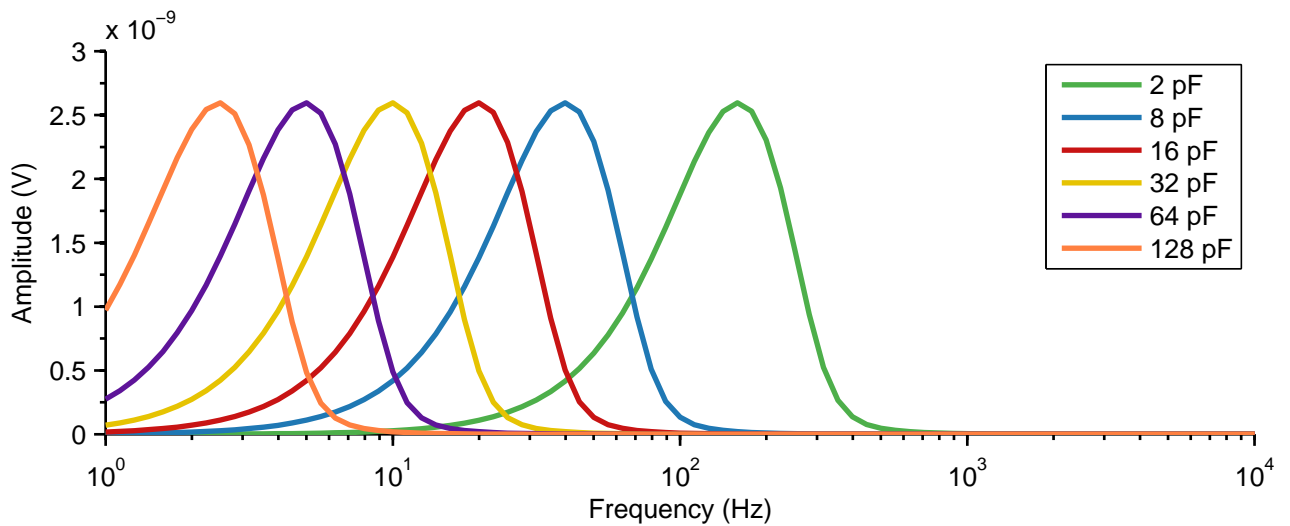


(b) AC simulation results.

Figure 4.11: Impulse response for *gaus1* for 10-th order filter.



(a) Transient simulation results.



(b) AC simulation results.

Figure 4.12: Impulse response for gaus2 for 10-th order filter.

The filter response to the test signal for the first-order wavelet transform are displayed in Figure 4.13. By evaluating the zero-crossing of the coefficients line for  $a = 2$ , or  $C = 2pF$ , in Figure 4.13(a), we can identify the local maxima/minima of the test signal at the instants:

$t = 0.23s$ ,  $t = 0.4s$ , and  $t = 0.7s$ . These are closer to the real values in Table 3.2— $t = 0.211s$ ,  $t = 0.446s$ , and  $t = 0.700s$ — than the ones obtained in the previous subsection for the 6<sup>th</sup> order  $L_2$  approximation.

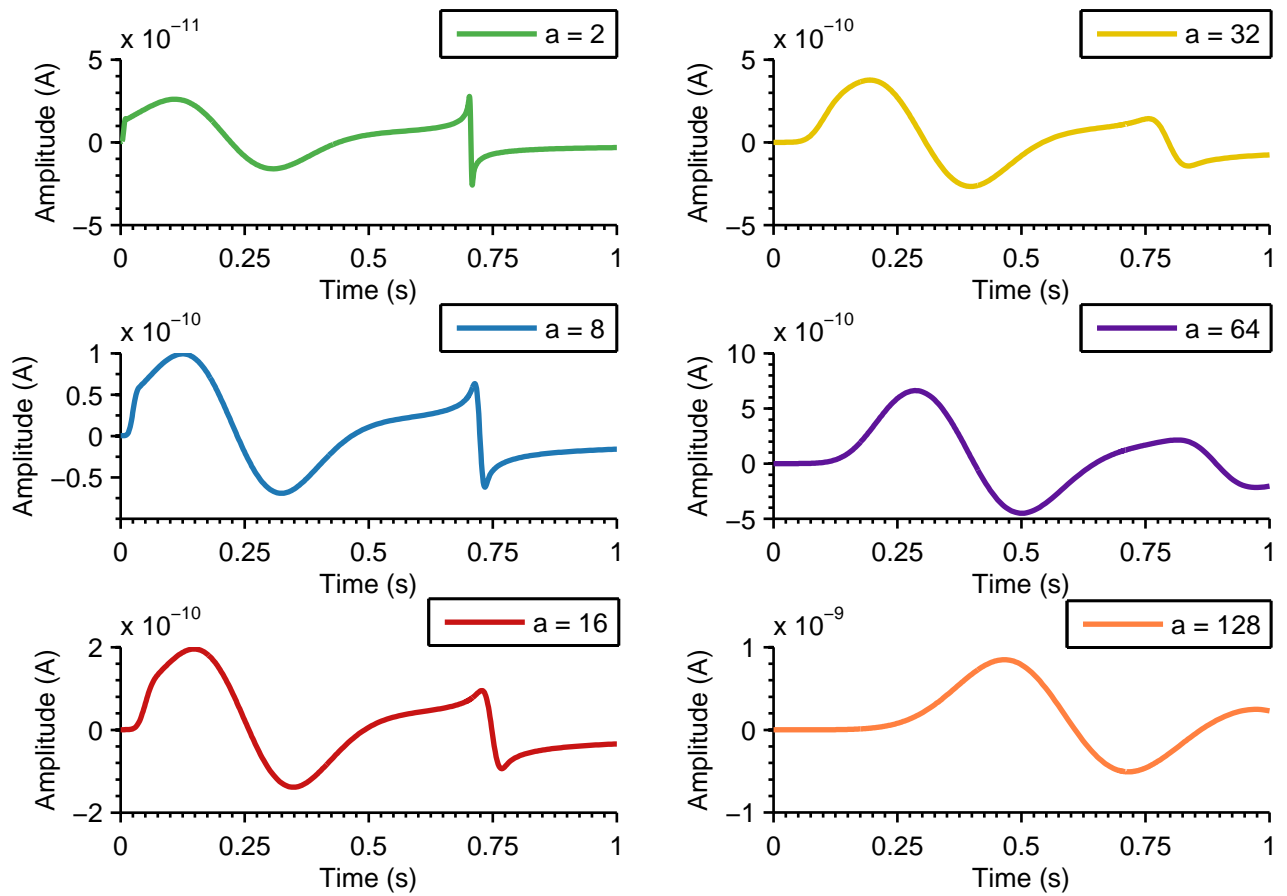
Figure 4.14 shows the responses to the test signal for the second-order wavelet. Notice that the delay between the coefficients lines in Figure 4.14(a) is larger than in the results obtained for the 6<sup>th</sup> order approximation. Not only can we not measure the Lipschitz exponent in a given instant, but also it is not possible to estimate  $\alpha$  for the final portions of the signal, since the coefficients line for the largest scale no longer carries this information. This variation in the pattern of the results for the coefficients lines for the second-order WT implemented by analog wavelet filters is an obstacle to implement a computational algorithm both for estimating the Lipschitz exponent and for the identification of the inflection points.

### 4.3.2 Lipschitz exponent estimation

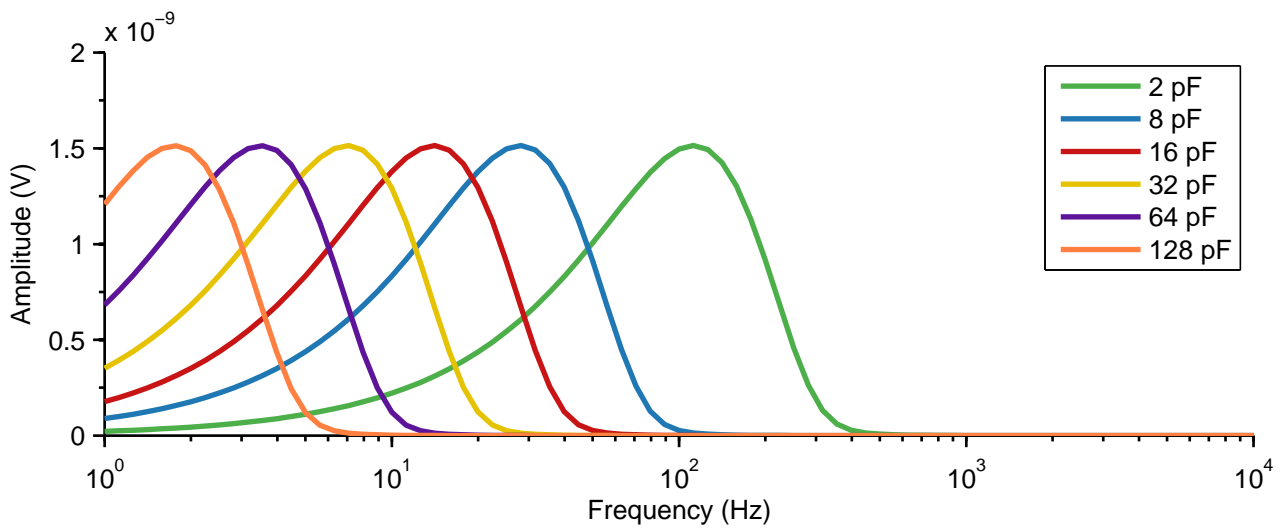
With the differences observed, the Lipschitz exponent can no longer be estimated by sampling the coefficients line at the same point (the point identified as the position of a local maximum or minimum), as was done in Chapter 3. In an attempt to verify if the coefficients lines for the second order WT, obtained in the previous subsection for  $L_2$  and Padé approximations, allow the correct estimation of the Lipschitz exponent, we manually calculate the values of  $\alpha$  by evaluating the peaks of the coefficients lines in Figures 4.14(a) and 4.9.

The extrema of each coefficient line for WT2 can be identified by taking the zero-crossing of the corresponding coefficients line for WT1. Before, in the system simulations results, there was no delay and these points of interest were identified at the maxima and minima points' positions. Now, because of the delay, these points are localized at such instants only for the smallest scales. The choice of which extrema to observe begins with identifying which ones on the coefficients line for scale  $a = 2$  are closer to the zero-crossing positions of the gaus1 coefficients line for  $a = 2$ . Then, the amplitude of these peaks across the other scales is measured and the Lipschitz coefficient at each of these points can be estimated by using the same method applied in Chapter 3. Also, the following results include the estimated Lipschitz coefficient at all maxima and minima points, except the initial and final points of the signal. For the test signal, this means that only three points were evaluated, identified hereafter as points 1, 2 and 3.

Table 4.1 contains the amplitude and positions of the coefficients lines' extrema at all tested scales for the 6<sup>th</sup> order  $L_2$  approximation. The positions at which these amplitudes were sampled demonstrate the presence of a delay, and allow a comparison between the position of a peak in the coefficients line for the second-order WT and the position of a zero-crossing in the coefficients line for the first-order WT. The Lipschitz coefficient values that were estimated with the data in Table 4.1 are shown in Table 4.2. This table contains the system results as a reference, which are identified as  $\alpha_{ref}$  and  $t_{ref}$ ; as well as the mean



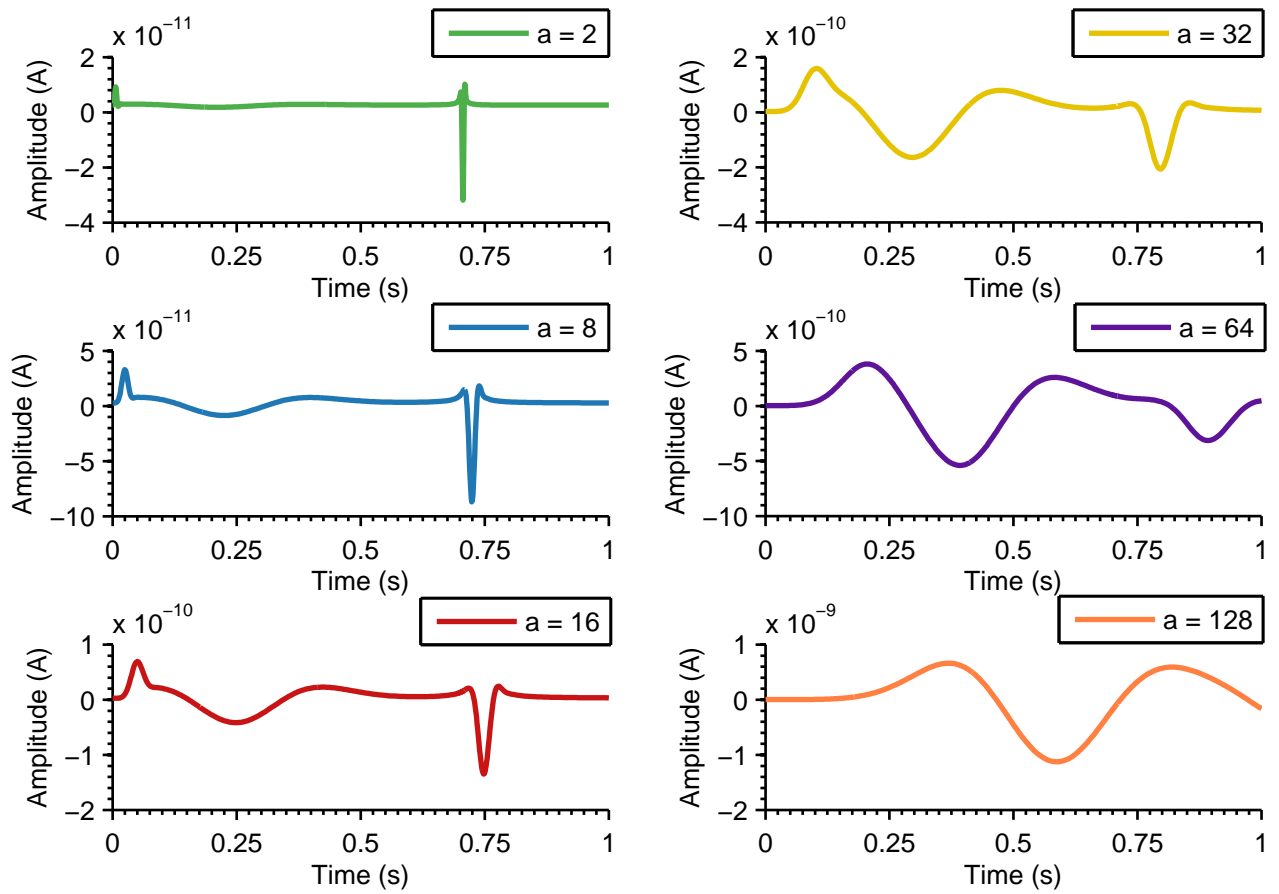
(a) Transient simulation results.



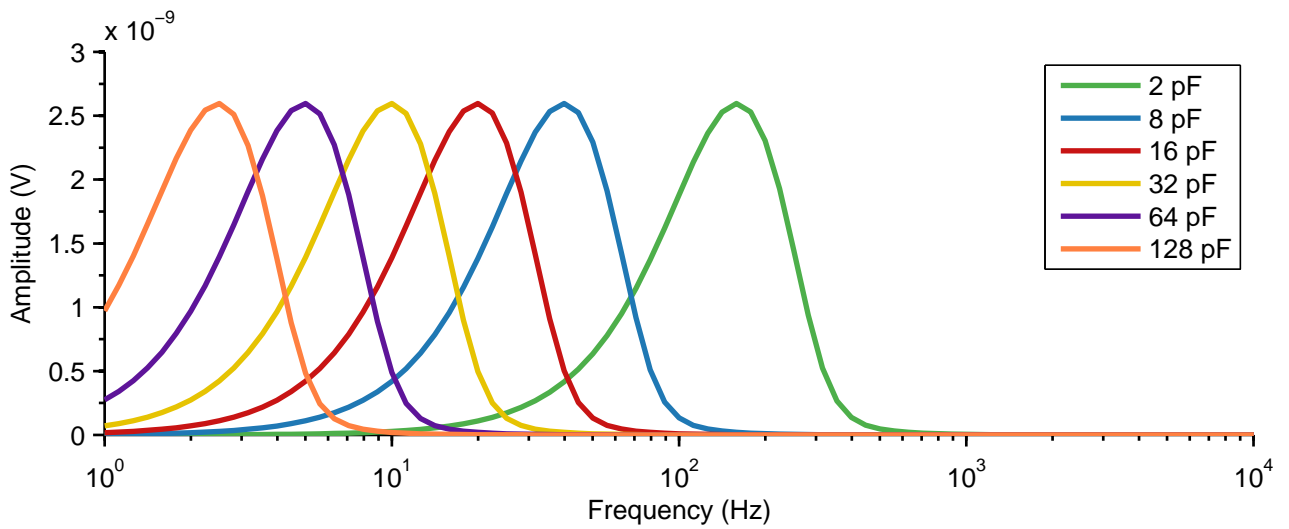
(b) AC simulation results.

Figure 4.13: First-order wavelet transform for the test signal and gaus1 wavelet basis in selected scales, using a 10<sup>th</sup> order Padé approximation.

value of  $\alpha$  for each pair of consecutive scales,  $\bar{\alpha}$ ; the maximum value of  $\alpha$  for each pair of consecutive scales,  $\alpha_{max}$ ; and these values subtracted by 0.5. In theory, the estimated



(a) Transient simulation results.



(b) AC simulation results.

Figure 4.14: Second-order wavelet transform for the test signal and gauss1 wavelet basis in selected scales, using a 10<sup>th</sup> order Padé approximation.

Lipschitz exponent should be the one in the column  $\bar{\alpha} - 0.5$ . However, the closest values to the references are observed in the column  $\bar{\alpha}$ : for points 1 and 3, the results are fairly



close to  $\alpha_{ref}$ ; however, this is not true for  $\alpha$  in point 2, which is estimated way below its reference value of 1.997. It is also worth noting that the positions of the extrema in the coefficients line for points 1 and 3 correspond to the reference ones, which does not happen for point 2 as well.

Table 4.1: Extrema of the WT2 coefficients lines across scales for the analog filter with sixth order transfer function approximation.

a	amp1 ( $10^{-10}$ A)	t1 (s)	amp2 ( $10^{-10}$ A)	t2 (s)	amp3 ( $10^{-10}$ A)	t3 (s)
128	-14.74	0.4415	6.985	0.6550	-6.235	0.9337
64	-5.421	0.3231	2.605	0.5035	-4.136	0.8168
32	-1.466	0.2611	7.418E-1	0.4327	-2.699	0.7587
16	-3.465E-1	0.2304	2.104E-1	0.4034	-1.764	0.7292
8	-6.182E-2	0.2184	7.824E-2	0.3861	-1.139	0.7151
2	2.692E-2	0.2101	3.754E-2	0.3812	-3.993E-1	0.7037

Table 4.2: Lipschitz exponents obtained with manual calculations for the analog filter with sixth order transfer function approximation.

point	t (s)	$\alpha_{ref}$	$t_{ref}(s)$	$\bar{\alpha}$	$\bar{\alpha} - 0.5$	$\alpha_{max}$	$\alpha_{max} - 0.5$
1	0.2101	1.7654	0.2110	1.699	1.199	2.487	1.987
2	0.3812	1.997	0.446	1.402	0.902	1.818	1.318
3	0.7037	0.6024	0.700	0.642	0.142	0.756	0.256

The results obtained for the 10<sup>th</sup> order Padé approximation are displayed in Tables 4.3 and 4.4, in the same fashion as the results for the 6<sup>th</sup> order approximation. The analysis of the results for this approximation yields similar conclusions: the best results for the estimated Lipschitz exponent are obtained by taking the mean of the calculated values for every pair of scales and not subtracting 0.5 from this result. Also, the values for  $\bar{\alpha}$  for points 1 and 3 are acceptable when compared to  $\alpha_{ref}$ , while there is significant error for point 2. This also happens for the positions of the coefficients line extrema (t) when compared to the reference position values ( $t_{ref}$ ). There is no data for the third point at scale  $a = 128$  because of the delay observed.

Comparing the results from Tables 4.1 and 4.3, we notice that the sampled values for amplitude are similar, indicating again that both transfer function approximations are good in general, differing in some details. Also, comparing the results from Tables 4.2 and 4.4, we notice that there is a larger difference between the sampling position and the reference position for point number 3 for the tenth order approximation, as well as the difference between  $\bar{\alpha}$  and the reference value for point number 1, also for the tenth order approximation.

Table 4.3: Extrema of the WT2 coefficients lines across scales for the analog filter with tenth order transfer function approximation.

a	amp1 ( $10^{-10}$ A)	t1 (s)	amp2 ( $10^{-10}$ A)	t2 (s)	amp3 ( $10^{-10}$ A)	t3 (s)
128	-11.26	0.5873	5.929	0.8216	—	—
64	-5.39	0.3922	2.587	0.5808	-3.14	0.8928
32	-1.639	0.2965	7.982E-1	0.4764	-2.059	0.7976
16	-4.149E-1	0.2477	2.28E-1	0.4284	-1.347	0.7477
8	-8.639E-2	0.2243	7.636E-2	0.4072	-8.703E-1	0.7240
2	1.815E-2	0.2108	2.886E-2	0.3897	-3.186E-1	0.7063

Table 4.4: Lipschitz exponents obtained with manual calculations for the analog filter with tenth order transfer function approximation.

point	t(s)	$\alpha_{ref}$	$t_{ref}(s)$	$\bar{\alpha}$	$\bar{\alpha} - 0.5$	$\alpha_{max}$	$\alpha_{max} - 0.5$
1	0.2108	1.7654	0.2110	1.630	1.130	2.264	1.764
2	0.3897	1.997	0.446	1.396	0.896	1.808	1.308
3	0.7603	0.6024	0.700	0.644	0.144	0.725	0.225

These results show that the Lipschitz exponent estimation, performed manually with the data from the extrema of the coefficients lines for gauss2 across scales, does not behave as expected from the theory. Also, there were only acceptable values of  $\alpha$  for two of the three points studied in this subsection. This characterizes a new limitation to the system. The research of why this happens and of other ways to estimate the Lipschitz exponent is left as future work. Other alternatives to estimate the signal's morphology can be investigated in the future, as well; one possibility would be to apply a form of interpolation, e.g. spline interpolation, to reconstruct the signal using the identified critical points information.

## 4.4 Discussion

In Chapter 3, the sampling algorithm was introduced. The test results show that the test signal's critical points are identified by the algorithm, and that the signal morphology around the local maxima and minima are well described by the estimated Lipschitz exponents. The tests also show that a signal can be recovered after sampling with the proposed polynomial reconstruction algorithm, with high correlation coefficients and low RMS error verified between the original and reconstructed signals. This also characterizes the first limitation of this method, however: the signal can not be recovered from its samples using a traditional Digital-to-Analog Converter (DAC). The polynomial reconstruction algorithm was only implemented in system-level. Nevertheless, the proposed system processes the sig-

nal with the wavelet transform, yielding as outputs the main characteristics of the signal, which can be useful for applications that do not need the analog signal to be recovered, but simply to be analyzed.

The tests to verify the quality of the reconstructed signal and the behaviour of the sampling scheme were run for the same scheme, but for different wavelet bases, sets of scales, and quantization resolutions. The results confirm some expectations: first of all, since the wavelet transform measures the correlation between a function and compressed or expanded versions of the wavelet basis, depending on the scale parameter, the critical points identification yields better results for the pairs of bases with a higher correlation with the input signal. Since the test signal did not present any sharp transitions and is in part composed of a gaussian function, the best results were verified for the pairs of gaussian wavelet bases, when compared to the pairs of Daubechies and biorthogonal bases. Still, the results for these bases was satisfactory in general, apart from the problem in the Lipschitz exponent estimation in the borders of the signal (first and last points, which were sampled as local minima), which altered the concavity of the signal in the first and last reconstruction polynomials.

Other results, even though not expected, led to interesting conclusions: first of all, the local maxima and minima points identification was expected to be more accurate for the a smaller scale, but there were no significant differences from the results with  $a = 1$  and  $a = 16$ , what shows that, for this signal, good results can be obtained for a scale as small as 16. Also, the Lipschitz exponent estimation for the set of 64 scales should yield better results than those obtained with the set of only four uniformly distributed scales. Again, there were no significant improvements in the results. Since the wavelet transform is implemented in circuit-level by analog wavelet filters, these results indicate that it could be possible to design filters that operate in not so high frequencies to localize the maxima and minima, and that a filter bank of four wavelet filters can be sufficient to estimate the morphology of the signal, instead of a bank with 64 filters. Regarding the effects of the quantization resolution on the samples, an interesting result was observed: the system, independently of the pair of wavelet bases applied, appears to be robust to quantization, since the results obtained for both the sampled amplitudes and Lipschitz coefficients at medium resolution (8 bits) are already of high quality, and very close to the results labeled as ideal, which do not take into account the effect of quantization. For the Lipschitz exponent, we are able to go even further in this analysis and say that a resolution of 4 bits is enough to describe the morphology of the signal with little error and high correlation.

These results were achieved with the sampling of the amplitude at nine critical points and of the estimated Lipschitz exponent at five of them, the local maxima and minima, a total of 14 samples. Since the signal has a duration of 1s, this yields 14 sps. With Figure 3.16, we can estimate that the Nyquist rate for this signal is around 40 sps, since for sampling rates equal to or higher than this, the recovery error is practically zero. This means that the sampling rate for this signal is approximately 3 times sub-Nyquist.

In Chapter 4, a circuit block diagram to implement the sampling algorithm was proposed. The first block to be tested was the one that implements the wavelet transforms of first and second orders, the analog wavelet filters. In order to do so, the transfer functions for the gaussian wavelet bases, which yielded the best results in the system simulations, were obtained via a 6<sup>th</sup> order  $L_2$  approximation and a 10<sup>th</sup> order Padé approximation. After obtaining the transfer functions, their orthonormal state-space representation matrices were determined, and the matrices coefficients were implemented as the values of the transconductor elements of a Gm-C filter.

The impulse response for the first order wavelet transform was correct for both approximations, yielding compressed and expanded versions of the gaus1 wavelet basis. However, the impulse response of the second order wavelet transform filter for the  $L_2$  approximation was asymmetric, what does not happen in the gaus2 wavelet basis. This was the main reason why a higher order approximation for the transfer function was included and, in fact, the impulse response for the Padé approximation of the second derivative of the gaussian function was symmetric. The transfer function for gaus2, in both cases, was obtained by multiplying the transfer function for the first derivative by  $s$ , instead of performing the approximation algorithms again.

When changing the input signal of the filters from an impulse to our test signal, the filter outputs must be the coefficients lines for the first and second order wavelet transforms. Since each filter operates at a different frequency, determined by the value of the capacitances, a bank of filters with different frequencies yields the coefficients lines at different scales. The results for the filters designed with both transfer function approximations appeared to be correct: the zero-crossings of the coefficients line at scale  $a = 2$  for gaus1 identified the positions of local maxima and minima, and the zero-crossings of the coefficients line at scale  $a = 128$  for gaus2 identified the inflection points. However, the coefficients lines present a delay across scales that was not observed in the system simulation results. This delay affects the Lipschitz estimation algorithm, since the extrema of each coefficient line can no longer be identified at the local maxima and minima positions. One way to identify their positions is to observe the zero-crossings of the first order WT for all scales, and not only at the smallest scale. However, the observed delay can also omit some of these extrema, as was the case for the  $a = 128$  coefficients line for the 10<sup>th</sup> order Padé approximation. This configures an obstacle in the development of a generic test, since there is no way to assure how this delay will affect the coefficients line for different input signals, bases, and transfer function approximations. In order to verify if the Lipschitz exponent could be correctly estimated by the extrema of the coefficients lines, these calculations were performed manually, and again yielded some unexpected results, since the values did not always correspond to the ones obtained in the system simulations and, in the cases when they did, they did not correspond to the ones obtained following the exact theoretical mathematical expression.

At last, there are some comments to be done regarding the circuit's total power consumption. Since the circuit implementation has not been completed, we cannot calculate the

total power consumption for the ADC. However, there are reasons to believe that this ADC should be suited for low-power applications. When evaluating the circuit block diagram in Figure 4.1, it is clear that many components are the same as the ones present in traditional ADC circuits: sample-and-hold blocks, comparators, logic gates. References [5, 27] show that it is possible to implement analog wavelet filters with low power consumption, with the same approach that was used in the presented project development. The Lipschitz exponent estimation is based on its mathematical definition. The operations of absolute value and division can be implemented with translinear circuits working in current mode, and the logarithm operation result can be obtained by using MOS transistors operating in weak inversion. Therefore, the proposed circuit should not dissipate much more static power than does the circuit of a traditional ADC. Furthermore, the dynamic power should be smaller than the one for a traditional ADC. The dynamic power is expressed as  $CfV_{dd}^2$ , where  $C$  is the capacitance between the output node and ground,  $f$  is the frequency, and  $V_{dd}$  is the supply voltage. The system-level results indicate that the sampling frequency necessary to represent and recover the signal is of 14 samples per second, roughly three times the Nyquist sampling rate. This means that the proposed wavelet-based ADC should dissipate approximately one third of the dynamic power dissipated by a Nyquist ADC for the same signal.

Moreover, an event-based sampling scheme can be advantageous when the input signal is sparse: for example, a signal such as an electrocardiogram signal has its information of interest densely located at its peaks, which are followed by long moments of nearly constant behaviour. Adaptive sampling allows such a signal to be sampled at a high rate during the peaks and at a low rate during the constant portions, while uniform sampling would acquire unnecessary samples of the latter. Another advantage of the proposed ADC is that it does not only convert the signal, but it also computes the wavelet-based analysis of the signal without the need of a digital signal processing block, which is necessary when the digital wavelet transform is applied.

# Chapter 5

## Conclusion

Analog-to-digital converters are fundamental components of the digital signal processing chains. Most of the ADCs in common appliances today follow the uniform sampling scheme obeying the Nyquist criterion. This project proposes a novel ADC with a wavelet-based sampling scheme, making use of the signal compression and regularity detection properties of the wavelet transform and inspired by the idea of undersampling, asynchronous sampling, and low power performance, uniting the fields of research of different groups.

The converter's system-level analysis, in Chapter 3, shows that the sampling algorithm correctly identifies the information necessary to represent the signal. To verify this, a polynomial reconstruction algorithm was proposed, and the results show low error and high correlation values between the input signal and the reconstructed signal. Also, the results allow a comparison between the effects of using different wavelet transform parameters (bases and scales) and different quantization resolutions to the quality of the signal reconstruction result. As expected, a basis with higher correlation with the input signal yields better results, as was the case of the gaussian wavelet bases with relation to the input test signal, when compared to the other wavelet bases in the tests. Also, the smaller the scale, the better the identification of maxima and minima points should be, although, for the test signal, there was not much difference between the two smallest scales of the tested sets, 1 and 16. A good Lipschitz coefficient estimation could be achieved with only four different scales, with no significant changes when compared to the results obtained with 64 scales. One unexpected result was that an 8-bit resolution for the amplitude values and a 4-bit resolution for the Lipschitz exponent are enough to yield a very good to high quality result.

Chapter 4 contains the circuit-level analysis. The wavelet transform is implemented with analog wavelet filters, which were evaluated for two different approximation methods for the transfer functions of the gaussian wavelet bases: the  $L_2$  method and the Padé method. The orthonormal state-space representation for each of the transfer functions obtained with these approximations were used to determine the values of the transconductances that compose the Gm-C filters, which implemented the first- and second- order wavelet transforms. The

impulse responses of the simulated filters were correct and in accordance to the expected behaviour, that is, versions of the first- and second-order wavelet bases, compressed for lower scales and expanded for larger scales. When the input signal of the filter is the test signal, the filter outputs the coefficients lines for each scale for the first- and second-order wavelet transforms of the test signal. Regarding the critical points identification, the results obtained with both approximations were similar and in conformity to the theoretical results. However, a delay that grows larger with the scales was observed in these results, and not in the system simulation results. This delay affected the Lipschitz exponent estimation, which had to have its algorithm adapted in order to be performed. Still, the results did not always correspond to the theoretical expected value. This problem could not be further investigated or fixed in due time, and is left as future work.

This project also resulted in a published conference paper at ISCAS, the IEEE International Symposium on Circuits and Systems, available in the IEEEXplore database [28].

## 5.1 Future works

The complete circuit analysis is left as future work. The investigation of the unexpected behaviour of the coefficients lines for the second order WT in the circuit simulation is included in the complete circuit analysis, as well as the study of possible alternatives not only to the circuit that was implemented, but also to the algorithm that requires the Lipschitz exponent estimation to implement the signal reconstruction. The complete circuit analysis will also allow a good estimation of the total power consumption, to determine if the proposed ADC is suitable for low-power applications, or if there are other better applications for it. Another future work includes the circuit-level implementation of the polynomial reconstruction algorithm, since it has only been implemented in system-level and the proposed ADC's outputs do not allow signal reconstruction with conventional DACs.

Regarding the system-level implementation, a possibility of future work is the investigation of a method to identify the inflection points which does not need approximations. Also, the tests run in this project only considered the effect of linear uniform quantization; future works can include the system simulations with other quantizer topologies.

# REFERENCES

- [1] ROSA, J. M. de la. Sigma-delta modulators: Tutorial overview, design guide, and state-of-the-art survey. *IEEE Transactions on Circuits and Systems—I: Regular Papers*, v. 58, n. 1, p. 1–21, 2011.
- [2] ALLEN, P. E.; HOLBERG, D. R. *CMOS Analog Circuit Design*. 2nd. ed. New York, NY: Oxford University Press, 2002.
- [3] TSIVIDIS, Y. Event-driven data acquisition and digital signal processing—a tutorial. *IEEE Transactions on Circuits and Systems—II: Express Briefs*, v. 57, n. 8, p. 577–581, 2010.
- [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] KAREL, J. M. H. et al. Implementing wavelets in continuous-time analog circuits with dynamic range optimization. *IEEE Transactions on Circuits and Systems—I: Regular Papers*, v. 59, n. 2, p. 229–242, 2012.
- [8] MARTINS, I. F. *Análise de Conversor Analógico-Digital Baseado em Transformada Wavelet*. 2014. <http://bdm.unb.br/handle/10483/13408>. Trabalho de Conclusão de Curso, Universidade de Brasília.
- [9] NYQUIST, H. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, v. 47, n. 2, p. 617–644, 1928.
- [10] SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, p. 379–423, 1948.



- [11] SAYINER, N.; SORENSEN, H. V.; VISWANATHAN, T. R. A level-crossing sampling scheme for a/d conversion. *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, v. 43, n. 4, p. 335–339, 1996.
- [12] CANDÈS, E. J.; WAKIN, M. B. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, v. 25, n. 2, p. 21–30, 2008.
- [13] STANFORD Analog-to-Information Project. <http://statweb.stanford.edu/~candes/A2IWeb/index.html>. Accessed: 24-Oct-2017.
- [14] YOO, J. et al. A compressed sensing parameter extraction platform for radar pulse signal acquisition. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, v. 2, n. 3, p. 626–638, 2012.
- [15] WAKIN, M. et al. A nonuniform sampler for wideband spectrally-sparse environments. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, v. 2, n. 3, p. 516–529, 2012.
- [16] SAYINER, N.; SORENSEN, H. V.; VISWANATHAN, T. R. A new signal acquisition technique. In: *Proceedings of the 35th Midwest Symposium on Circuits and Systems*. Washington, DC, USA: IEEE, 1992. p. 1140–1142.
- [17] SAYINER, N.; SORENSEN, H. V.; VISWANATHAN, T. R. A non-uniform sampling technique for a/d conversion. In: *1993 IEEE International Symposium on Circuits and Systems (ISCAS'93)*. Chicago, IL, USA: IEEE, 1993. p. 1220–1223.
- [18] EL-MASRY, T. E. E.; EL-DIB, D. A. High precision clockless adc using wavelet neural network. In: *28th International Conference on Microelectronics (ICM)*. Giza, Egypt: IEEE, 2016. p. 365–368.
- [19] MATHWORKS Documentation: Wavelet Toolbox. <http://www.mathworks.com/help/wavelet/index.html>. Acessado em 10/11/2014.
- [20] STRANG, G.; NGUYEN, T. *Wavelets and Filter Banks*. Revised edition. Wellesley, MA: Wellesley-Cambridge, 1997.
- [21] MALLAT, S.; HWANG, W. L. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, v. 38, n. 2, p. 617–643, 1992.
- [22] OGATA, K. *Modern Control Engineering*. 5th. ed. New Jersey, USA: Prentice Hall, 2009.
- [23] DAUBECHIES, I. *Ten Lectures on Wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992. (CBMS-NSF Regional Conference Series in Applied Mathematics).

- [24] 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.
- [25] 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.
- [26] LEON-GARCIA, A. *Probability, Statistics, and Random Processes for Electrical Engineering*. 3rd. ed. New Jersey, USA: Prentice Hall, 2008.
- [27] HADDAD, S. A. P.; BAGGA, S.; SERDIJN, W. A. Log-domain wavelet bases. *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 52, n. 10, p. 2023–2032, 2005.
- [28] MARTINS, I. F. et al. A novel wavelet-based analog-to-digital converter. In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. Baltimore, MD, USA: IEEE, 2017. p. 1428–1431. Disponível em: <<http://ieeexplore.ieee.org/document/8050622/>>.
- [29] ANDRADE, J. A. A. de. Projeto de um conversor a/d de baixo consumo para aplicações de alta resolução. Projeto Final de Graduação. Faculdade UnB Gama, Universidade de Brasília. 2014.

# APPENDICES

# I. RESUMO ESTENDIDO EM PORTUGUÊS

## Introdução

Os conversores analógico-digitais são tradicionalmente compostos por quatro blocos: um filtro anti-aliasing; um amostrador; um quantizador; e um codificador. Normalmente, o sinal é amostrado a uma taxa constante e uniforme, obedecendo ao Critério de Nyquist-Shannon. Apesar de ser este o processo que guia o funcionamento de conversores A/D em diversas aplicações comerciais na atualidade, ele estabelece apenas uma condição suficiente, e não necessária, para que um sinal seja amostrado de forma que possa ser recuperado sem perdas posteriormente. Vale lembrar, ainda, que o processo de quantização, que faz parte do processo de conversão analógico-digital, é um processo que acarreta, necessariamente, perda de informação.

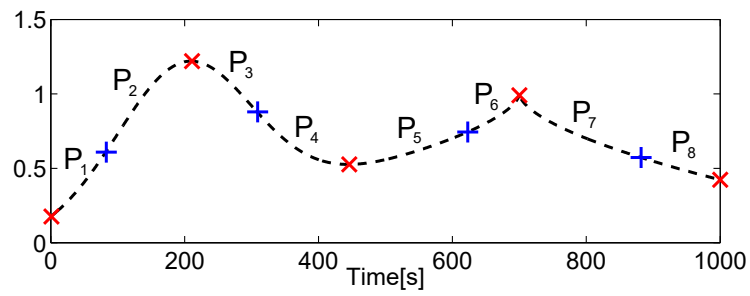
Outras formas de realizar a amostragem de um sinal são estudadas em outros campos de pesquisa, entre eles, podemos destacar o *compressive sensing* (CS) e o estudo de conversores assíncronos, como os conversores que operam por *level-crossing*. Este projeto propõe um conversor A/D com amostragem assíncrona baseada em propriedades da transformada wavelet (WT), com vistas a reduzir o consumo de potência do circuito. Esse processo de amostragem é realizado identificando-se os pontos críticos do sinal por meio do processamento com a transformada wavelet. Essa informação, juntamente com a descrição da morfologia do sinal nos trechos entre esses pontos, permite representar o sinal de uma forma que leve em consideração suas propriedades e peculiaridades. A morfologia do sinal é descrita pelo coeficiente de Lipschitz, que também pode ser estimado com auxílio da WT. A Figura I.1(a) ilustra o processo de identificação de máximos e mínimos locais e pontos de inflexão para um sinal de teste, e a Figura I.1(b) mostra uma função cujo comportamento varia de acordo com o coeficiente de Lipschitz. A expressão dessa função é o ponto de partida para o desenvolvimento de um algoritmo de reconstrução polinomial, também proposto neste projeto, que permite avaliar o algoritmo de amostragem sugerido. A Figura I.1(a) também ilustra como esse algoritmo funcionaria, reconstruindo o sinal por partes com os polinômios.

## Referencial teórico

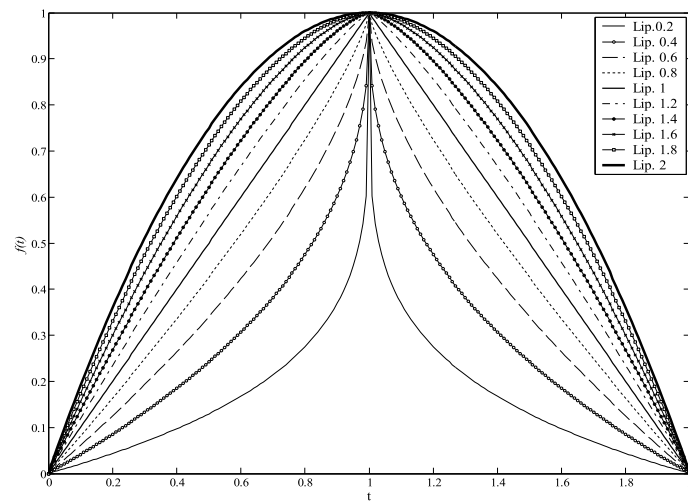
A transformada wavelet tem dois parâmetros: escala  $a$  e posição  $u$ , como expresso pela Equação I.1.

$$Wf(u, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi^* \left( \frac{t-u}{a} \right) dt. \quad (\text{I.1})$$

A propriedade da WT que permite que ela seja usada para a identificação de pontos críticos é a de que ela funciona como um operador diferencial de  $n$ -ésima ordem. Assim,



(a) Detecção de pontos críticos do sinal.



(b) Uma função cuja morfologia varia com o expoente de Lipschitz.

Figura I.1: Informação amostrada do sinal de entrada pelo ADC proposto.

determinando os coeficientes que são iguais a zero para uma transformada realizada com uma base com um *vanishing moment* ( $n = 1$ ), os pontos máximos e mínimos do sinal são localizados. Analogamente, para um abase com  $n = 2$ , identificam-se os pontos de inflexão do sinal.

A escolha das escalas permite ajustar a resolução tempo-frequência da transformada. Para escalas pequenas, a base é comprimida e são identificados os detalhes do sinal, ou seus componentes de mais altas frequências; para escalas maiores, a base é expandida e são identificados os componentes de baixa frequência. Assim, na identificação de máximos e mínimos, é escolhida a menor escala; para os pontos de inflexão, é escolhida a maior escala.

O coeficiente de Lipschitz  $\alpha$  pode ser estimado por uma expressão derivada de sua definição, onde  $A > 0$  é uma constante:

$$\log_2 |Wf(u, a)| \leq \log_2 A + \left( \alpha + \frac{1}{2} \right) \log_2 a \quad (\text{I.2})$$

A função da Figura I.1(b) tem sua expressão definida pela Equação I.3, para valores de  $\alpha$  entre 0.2 e 2.

$$f(t) = |1 - t|^\alpha \quad (\text{I.3})$$

## Desenvolvimento

O fluxograma do processo de amostragem é mostrado na Figura I.2. Depois de serem escolhidas as bases e escalas da transformada wavelet, as transformadas de primeira e segunda ordem são computadas para a identificação dos pontos críticos e para o cálculo do coeficiente de Lipschitz. Algumas aproximações são necessárias na identificação dos pontos de inflexão: a detecção de máximos e mínimos tem preferência sobre a detecção de pontos de inflexão; ainda, o algoritmo assume que há apenas um ponto de inflexão entre um máximo e um mínimo consecutivos. Assim, se for identificado mais de um ponto de inflexão em um trecho, a posição da inflexão é aproximada para a posição média dos pontos detectados; se não for identificado nenhum ponto de inflexão no trecho, a posição da inflexão é aproximada para o ponto médio do trecho. As saídas do sistema de amostragem são a amplitude nos pontos críticos e o coeficiente de Lipschitz apenas nos máximos e mínimos. Após o processo de amostragem, essas saídas devem ser quantizadas e codificadas.

A Figura I.3 mostra o diagrama de blocos do circuito proposto para implementar o ADC, que segue os mesmos passos do fluxograma da Figura I.2. A transformada wavelet de primeira e segunda ordens para o sinal de entrada  $v_{in}$  é calculada, e as linhas de coeficientes das escalas  $a_1$  e  $a_2$ , respectivamente,  $\text{coeff}_1$  e  $\text{coeff}_2$ , passam por um comparador, que implementa a detecção de *zero-crossing*. Isso gera os sinais  $\mathbf{t}_m$  e  $\mathbf{t}_{inf}$ , que identificam os máximos e mínimos locais, e os pontos de inflexão. As amplitudes são amostradas em todos os pontos críticos, o que é representado pelo bloco de *sample-and-hold* controlado por  $\mathbf{t}_{amp}$ .

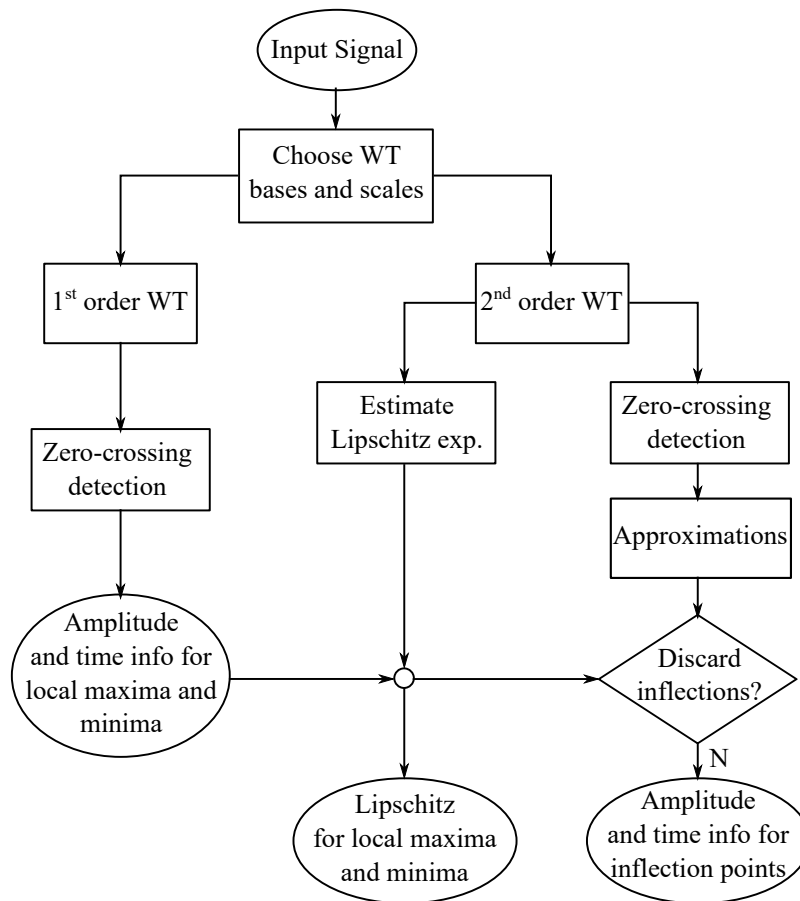


Figura I.2: Algoritmo de amostragem.

O coeficiente de Lipschitz é estimado em todos os pontos do sinal pela implementação de sua definição matemática com os blocos ‘abs’, ‘log’, e divisor, resultando no sinal  $lpz$ , que é amostrado apenas nos máximos e mínimos com o bloco de *sample-and-hold* controlado por  $\tau_m$ . A lógica de chaveamento, controlada por  $sync$ ,  $sw$ , é incluída com a intenção de se usar apenas um quantizador para as informações de amostragem de amplitude ( $amp$ ) e do coeficiente de Lipschitz ( $\alpha$ ), a fim de se reduzir o consumo de potência do circuito. Assim, um único sinal,  $v_{out}$ , é quantizado ( $v_{out,q}$ ) e codificado.

A transformada wavelet é implementada com filtros analógicos. As funções de transferência são aproximadas usando dois métodos diferentes: o método  $L_2$ , com uma aproximação de sexta ordem, e o método de Padé, com uma aproximação de décima ordem. Após se obter a aproximação da função de transferência para os filtros de primeira e segunda ordem, a sua representação ortonormal no espaço de estados é calculada, permitindo que o filtro seja implementado utilizando-se a topologia Gm-C (transcondutância-capacitância).

Para verificar que a informação extraída do sinal é suficiente para que ele seja recuperado, é desenvolvido também um algoritmo de reconstrução polinomial, expresso pela Equação I.4, na qual:

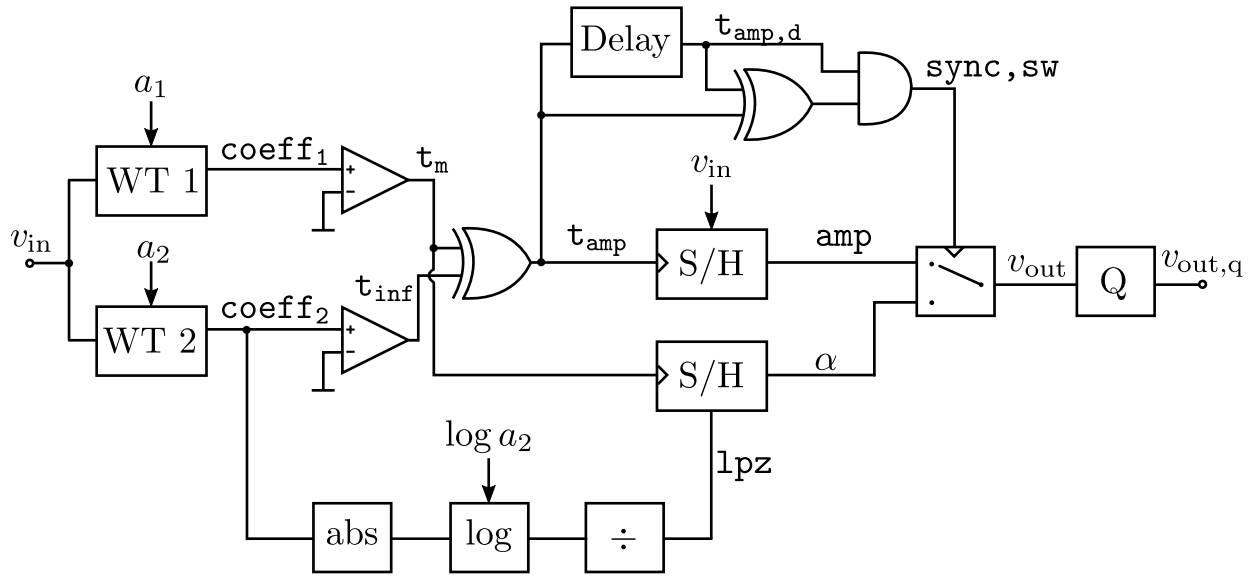


Figura I.3: Diagrama de blocos do circuito proposto para o conversor A/D.

- $A_i$  e  $A_m$  são as amplitudes amostradas nos pontos de inflexão, e nos pontos de máximos e mínimos locais;
- $\alpha_m$  é o valor do coeficiente de Lipschitz estimado nos máximos e mínimos locais;
- $t_0$  e  $t_f$  são os instantes inicial e final do trecho que o polinômio  $P_n$  reconstrói;
- $\mu$  é o suporte no tempo de  $P_n$  i.e.  $\mu = t_f - t_0$ ;
- $\tau$  é a posição do máximo ou mínimo local.

$$P_n(t) = A_i + (A_m - A_i) \left( 1 - \left| \frac{\tau - t}{\mu} \right|^{\alpha_m} \right), \quad t_0 \leq t \leq t_f \quad (\text{I.4})$$

Adicionar  $\tau$  e  $\mu$  à expressão da Equação I.4 permite o deslocamento no tempo, enquanto o deslocamento em amplitude é garantido pela multiplicação por  $(A_m - A_i)$ , seguida da soma de  $A_i$ .  $(A_m - A_i)$  também indica a concavidade do trecho.

## Resultados

A Figura I.4 mostra o resultado da reconstrução do sinal com as bases gaussianas, menor escala 16, maior escala 64, para o sinal de teste, para quantização de 8 bits. Neste caso, o coeficiente de correlação entre os sinais original e reconstruído é de 0.989, e o erro RMS de reconstrução é de 0.00874.

## Conclusão

Este projeto apresenta um conversor analógico-digital baseado em transformada wavelet. Esse conversor é avaliado a nível de sistema e a nível de circuito. Para que o método de amostragem proposto possa ser verificado, é desenvolvido também um algoritmo de



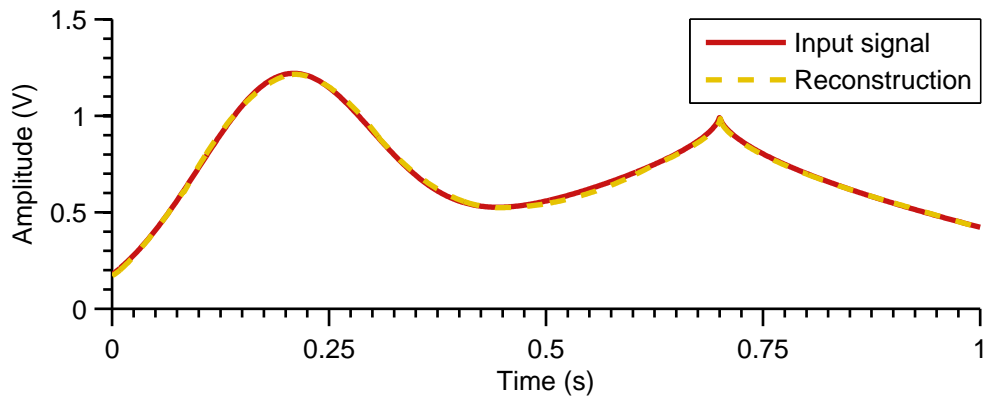


Figura I.4: Sinal de entrada (vermelho) e sua reconstrução (amarelo, linha tracejada).

reconstrução polinomial. Os resultados da implementação a nível de sistema mostram que é possível recuperar o sinal de entrada com alta qualidade, atingindo-se alta correlação e baixo erro de reconstrução.

## II. MATLAB SCRIPTS

This Appendix contains the MATLAB scripts used in the project, with due credit.

### II.1 Sampling algorithm

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % This MATLAB script implements a wavelet-based sampling algorithm.
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8
9 close all;
10 clear;
11 clc;
12
13 %% 1. Generate the test signal.
14 format long;
15 t = linspace(0,1,1000);
16 sig = gaussmf(t,[0.1 0.2])+fz_p_sandro(t,1,0.75,0.7,0.6);
17
18 %% 2. Select WT parameters.
19
20 % Uncomment the desired set of scales:
21
22 % escalas = [16:16:64];
23 escalas = [1:1:64];
24
25
26 % Uncomment the first and second order bases:
27
28 % base1 = 'gaus1';
29 % base1 = 'db1';
30 % base1 = 'bior1.1';
31 base1 = 'rbio1.1';
32
33 % base2 = 'gaus2';
34 % base2 = 'mexh';
35 % base2 = 'db2';
36 % base2 = 'bior2.2';
37 base2 = 'rbio2.2';
38
39
40 % Compute the continuous wavelet transform with MATLAB function 'cwt':
```

```

41 coefic = cwt(sig,escalas,base1);
42 coefic2 = cwt(sig,escalas,base2);
43
44 %% 3. Pick the coefficients line for critical points detection.
45
46 % 1st order WT: choose line at smaller scale:
47 linha_coef = coefic(1,:);
48
49 % 2nd order WT: choose line at larger scale:
50 s2 = size(coefic2);
51 linha_coef2 = coefic2(s2(1),:);
52
53 %% 4. Zero-crossing detection.
54
55 sing = +(linha_coef >= 0);           % zero-crossing for local maxima and minima
56 inflexao = +(linha_coef2 >= 0);     % zero-crossing for inflection points
57                                     % '+' casts the variables from binary to double
58
59 % Identify transitions...
60 idx = 1:length(sig);
61 tran = idx.*sing;
62 tran2 = idx.*inflexao;
63
64 x = diff(tran);
65 up = find(x>1);
66 down = find(x<0);
67 bordas = sort([up down]);
68
69 x2 = diff(tran2);
70 up2 = find(x2>1);
71 down2 = find(x2<0);
72 bordas2 = sort([up2 down2]);
73
74 % This includes the first and last points of the signal as local maxima and/or
75 % minima:
76 if bordas(1)~=1
77     bordas = [1 bordas];
78 end
79 if bordas(end)~=length(sig)
80     bordas = [bordas length(sig)];
81 end
82
83 lmaxmin = bordas;           % lmaxmin: identified local maxima/minima positions
84
85 %% 5. Approximations:
86
87 % 5.1: If any inflection point coincides with a local maxima/minima, remove it.
88 x = intersect(bordas,bordas2);
89
90 if ~isempty(x)
91     for k = 1:length(x)
92         id = find(bordas2==k);

```

```

93     bordas2(id) = [];
94     end
95 end
96
97
98 % 5.2: Only one inflection between consecutive maximum and minimum:
99
100 inflection = [];
101 for k = 2:numel(bordas)
102     int_i = bordas(k-1);    % beginning of portion
103     int_f = bordas(k);    % end of portion
104
105     test_i = find(bordas2 >= int_i);
106     test_f = find(bordas2 < int_f);
107
108     inf = intersect(test_i,test_f);
109
110     if numel(inf)==0
111         inflection(k-1) = round((int_i+int_f)/2);
112     elseif numel(inf)==1
113         inflection(k-1) = bordas2(inf);
114     else
115         inflection(k-1) = round(sum(bordas2(inf))/numel(inf));
116     end
117 end
118
119 %% 6. Calculate the Lipschitz coefficient for every point in the signal.
120
121 % applies custom function 'fz_lpz_calc_modif'
122 lpz = abs(fz_lpz_calc_modif(coefic2,escalas) - 0.5);
123
124 %% 7. Sampling at critical points.
125
126 sampling_clk = sort([lmaxmin inflection]);
127
128 % Amplitude sampling...
129 amp = sig(sampling_clk);
130 amp_graph = [];
131
132 for k = 1:length(sampling_clk)-1
133     amp_graph(sampling_clk(k):sampling_clk(k+1)-1) = amp(k);
134 end
135 amp_graph = [amp_graph amp(end)];
136
137
138 % Lipschitz coefficient sampling...
139 lip = lpz(lmaxmin);
140
141 lip_graph = [];
142 for k = 1:length(lmaxmin)-1
143     lip_graph(lmaxmin(k):lmaxmin(k+1)-1) = lip(k);
144 end

```

```

145 lip_graph = [lip_graph lip(end)];
146
147 %% 8. Quantization (applies custom function 'quant').
148
149 % Amplitudes quantization...
150 n = max(amp_graph);
151 amp_n = amp_graph/n;
152 amp_quant4 = n*quant(amp_n,4);
153 amp_quant8 = n*quant(amp_n,8);
154 amp_quant12 = n*quant(amp_n,12);
155
156 % 4 bits
157 ampq4 = amp_quant4(lmaxmin);           % amplitude @ local maxima/minima
158 infq4 = amp_quant4(inflexion);        % amplitude @ inflection
159
160 % 8 bits
161 ampq8 = amp_quant8(lmaxmin);
162 infq8 = amp_quant8(inflexion);
163
164 % 12 bits
165 ampq12 = amp_quant12(lmaxmin);
166 infq12 = amp_quant12(inflexion);
167
168
169 % Lipschitz coefficient quantization...
170 n = max(lip_graph);
171 lip_n = lip_graph/n;
172 lip_quant4 = n*quant(lip_n,4);
173 lip_quant8 = n*quant(lip_n,8);
174 lip_quant12 = n*quant(lip_n,12);
175
176 lipq4 = lip_quant4(lmaxmin);           % 4 bits
177 lipq8 = lip_quant8(lmaxmin);           % 8 bits
178 lipq12 = lip_quant12(lmaxmin);         % 12 bits

```

### II.1.1 Quantization function

This function was developed by José E. G. de Medeiros.

```

1 function [ output_code ] = quant(input, N)
2 %ADC_ELETRONICA2 Converte um valor analógico em inteiro
3 %   input = input value
4 %   N = resolution
5 %   output_code = output value
6
7 LSB = 1/(2^N);
8 code = zeros(1,length(input));
9
10 %% Adjust RNG to generate always the same sequence based on the seed number

```

```

11 % s = RandStream('mt19937ar', 'Seed', seed);
12 % RandStream.setGlobalStream(s);
13 %
14 % error_vector = randi(2^N - 1, 1, 5);
15 % erro = zeros(1, 2^N);
16 % erro(error_vector(1)) = 0.1*LSB;
17 % erro(error_vector(2)) = -0.3*LSB;
18 % erro(error_vector(3)) = -0.5*LSB;
19 % erro(error_vector(4)) = 0.5*LSB;
20 % erro(error_vector(5)) = 0.3*LSB;
21
22 % Quantizer process
23 for j = 1:length(input)
24     for k = 1:(2^N)-1
25         if(input(j) > k*LSB)
26             code(j) = k*LSB;
27         else
28             break;
29         end
30     end
31 end
32
33 % Output vector
34 % output_code = (code) * 2^N;
35 output_code = (code);
36 end

```

## II.1.2 Lipschitz exponent estimation function

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % The fz_lpz_calc_modif function estimates the Lipschitz exponent
7 % at all points in a signal.
8 %
9 % Inputs:
10 % coef ..... WT coefficients matrix
11 % scales ..... WT scales vector
12 %
13 % Output:
14 % lpz ..... vector containing the Lipschitz coefficients
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17 function [lpz] = fz_lpz_calc_modif(coef,scales)
18
19 dim = size(coef);
20

```

```

21 coef = abs(coef);
22
23 a = log2(coef);
24
25 for n=1:dim(2)
26     alpha(:,n) = diff(a(:,n)') ./ diff(log2(scales));
27 end
28
29 lpz = sum(alpha) / (dim(1)-1);
30
31 end

```

### II.1.3 Function to generate test signal

This function was developed by José Alberto [29].

```

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

```

## II.2 Reconstruction algorithm

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % This function implements the piecewise polynomial reconstruction
7 % algorithm proposed in the manuscript.
8 %
9 % Inputs:
10 % num_pol ..... number of polynomials
11 % A_i ..... amplitudes sampled at inflection points
12 % A_s ..... amplitudes sampled at local maxima/minima
13 % am_lpz ..... sampled Lipschitz coefficients
14 % tempos ..... critical points positions
15 % delta ..... increment in time
16 % tau ..... local maxima and minima positions

```

```

17 %
18 % Outputs:
19 % result ..... reconstructed signal
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21
22 function [result] = reconstruct(num_pol,A_i,A_s,am_lpz,tempos,delta,tau)
23
24 tempo = [];
25 pol = [];
26 result = [];
27
28 for k = 1:num_pol
29     tempo = tempos(k):delta:tempos(k+1);
30
31     nu = tempos(k+1) - tempos(k);    % time support
32
33     seli = ceil(k/2);                % select inflection
34     sels = floor(k/2 + 1);          % select maxima/minima
35
36     pol = A_i(seli) + (A_s(sels)-A_i(seli))*...
37         (1-(abs((tau(sels) - tempo)./nu).^am_lpz(sels)));
38
39     % Assembling the signal...
40     if k==1
41         result = pol;
42     else
43         result = [result pol];
44         result = result(1,1:length(result)-1);
45     end
46 end

```

## II.3 Reconstruction tests

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % This MATLAB script tests the reconstruction algorithm, implemented
7 % in the custom function 'reconstruct', for different output
8 % resolutions (ideal, 4, 8 and 12 bits). For each case, the input
9 % and reconstructed signals are plotted and the RMS error and the
10 % correlation coefficient between them are displayed on the command
11 % window.
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13
14 %% "Ideal" case (outputs before quantizations)
15

```



```

16 % Polynomial parameters
17 num_pol = numel(lmaxmin)+numel(inflexion)-1;
18 A_i = sig(inflexion);
19 A_s = sig(lmaxmin);
20 am_lpz = lpz(lmaxmin);
21 tempos = sampling_clk/numel(t);
22 delta = (t(end) - t(1))/(numel(t));
23 tau = lmaxmin/numel(t);
24
25 % Reconstruction...
26 result = reconstruct(num_pol,A_i,A_s,am_lpz,tempos,delta,tau);
27
28 rms = RMSError(sig,result);
29 erro_rel = RELError(sig,result);
30 rho = correlation(sig,result);
31
32
33 figure
34 plot(t,sig,'b-',t,result,'r-', 'LineWidth',2)
35 title('Reconstructed signal', 'FontSize',18)
36 xlabel('t(s)', 'FontSize',18)
37 set(gca, 'FontSize',16)
38 grid on
39
40 disp('Ideal case:')
41 x = ['Correlation coefficient: ', num2str(rho)];
42 disp(x)
43 x = ['RMS error: ', num2str(rms)];
44 disp(x)
45
46 %% 4 bits resolution
47
48 % Polynomial parameters:
49 num_pol = numel(lmaxmin)+numel(inflexion)-1;
50 % A_i = infq4;
51 % A_s = ampq4;
52 A_i = sig(inflexion);
53 A_s = sig(lmaxmin);
54 am_lpz = lipq4;
55 % am_lpz = lpz(lmaxmin);
56 tempos = sampling_clk/numel(t);
57 delta = (t(end) - t(1))/(numel(t));
58 tau = lmaxmin/numel(t);
59
60 % Reconstruction...
61 result = reconstruct(num_pol,A_i,A_s,am_lpz,tempos,delta,tau);
62
63 rms = RMSError(sig,result);
64 erro_rel = RELError(sig,result);
65 rho = correlation(sig,result);
66
67

```

```

68 figure
69 plot(t,sig,'b-',t,result,'r—','LineWidth',2)
70 title('Reconstructed signal','FontSize',18)
71 xlabel('t(s)','FontSize',18)
72 set(gca,'FontSize',16)
73 grid on
74
75 disp('Quant 4 bits')
76 x = ['Correlation coefficient: ',num2str(rho)];
77 disp(x)
78
79 x = ['RMS error: ',num2str(rms)];
80 disp(x)
81
82 %% 8 bits resolution
83
84 % Polynomial parameters:
85 num_pol = numel(lmaxmin)+numel(inflexion)-1;
86 % A_i = infq8;
87 % A_s = ampq8;
88 A_i = sig(inflexion);
89 A_s = sig(lmaxmin);
90 am_lpz = lipq8;
91 % am_lpz = lpz(lmaxmin);
92 tempos = sampling_clk/numel(t);
93 delta = (t(end) - t(1))/(numel(t));
94 tau = lmaxmin/numel(t);
95
96 % Reconstruction...
97 result = reconstruct(num_pol,A_i,A_s,am_lpz,tempos,delta,tau);
98
99 rms = RMSError(sig,result);
100 erro_rel = RELError(sig,result);
101 rho = correlation(sig,result);
102
103
104 figure
105 plot(t,sig,'b-',t,result,'r—','LineWidth',2)
106 title('Reconstructed signal','FontSize',18)
107 xlabel('t(s)','FontSize',18)
108 set(gca,'FontSize',16)
109 grid on
110
111 disp('Quant 8 bits')
112 x = ['Correlation coefficient: ',num2str(rho)];
113 disp(x)
114
115 x = ['RMS error: ',num2str(rms)];
116 disp(x)
117
118 %% 12 bits quantization
119

```

```

120 % Polynomial parameters:
121 num_pol = numel(lmaxmin)+numel(inflexion)-1;
122 % A_i = infq12;
123 % A_s = ampq12;
124 A_i = sig(inflexion);
125 A_s = sig(lmaxmin);
126 am_lpz = lipq12;
127 % am_lpz = lpz(lmaxmin);
128 tempos = sampling_clk/numel(t);
129 delta = (t(end) - t(1))/(numel(t));
130 tau = lmaxmin/numel(t);
131
132 % Reconstruction...
133 result = reconstruct(num_pol,A_i,A_s,am_lpz,tempos,delta,tau);
134
135 rms = RMSError(sig,result);
136 erro_rel = RELError(sig,result);
137 rho = correlation(sig,result);
138
139
140 figure
141 plot(t,sig,'b-',t,result,'r—','LineWidth',2)
142 title('Reconstructed signal','FontSize',18)
143 xlabel('t(s)','FontSize',18)
144 set(gca,'FontSize',16)
145 grid on
146
147 disp('Quant 12 bits')
148 x = ['Correlation coefficient: ',num2str(rho)];
149 disp(x)
150
151 x = ['RMS error: ',num2str(rms)];
152 disp(x)

```

### II.3.1 Error metrics

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % This function outputs the root eman square error 'rms' between the
7 % input vectors 'x' and 'y'.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 function rms = RMSError(x,y)
11
12 if numel(x)~=numel(y)
13

```

```

14 else
15     n = numel(x);
16     rms = sqrt(sum((x-y).^2)/n);
17
18 end
19
20 end

```

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % This function outputs the relative error 'erro_rel' between the
7 % input vectors 'x' and 'y'.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 function erro_rel = RELError(x,y)
11
12 erro_abs = abs(x-y);           % erro absoluto
13 %erro_abs_max = max(erro_abs)   % valor máximo do erro absoluto
14
15 erro_rel = 100*erro_abs./x;     % erro relativo percentual
16 %erro_rel_max = max(erro_rel)   % valor máximo do erro relativo (%)
17
18 end

```

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % This function outputs the correlation coefficient 'rho' between
7 % input vectors 'x' and 'y'.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 function rho = correlation(x,y)
11
12 Rxy = corrcoef(x,y);           % Correlation matrix
13
14 Ex2 = Rxy(1,1);
15 Ey2 = Rxy(2,2);
16 Ex = sqrt(Ex2 - var(x));       % E[X]
17 Ey = sqrt(Ey2 - var(y));       % E[Y]
18 Exy = Rxy(1,2);                % E[XY]
19
20 rho = (Exy - Ex*Ey)/(std(x)*std(y)); % Correlation coefficient

```

## II.4 Gaussian filters

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % A Novel Wavelet-Based Analog-to-Digital Converter
3 %                                     by Isadora Freire Martins
4 %
5 %
6 % This MATLAB script shows how to calculate the transfer function
7 % of the gaussian function and its orthonormal representation.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 % Transfer functions for 1st derivative 6th order filter:
11 num = [-0.09846 -0.1683 -8.326 6.642 139.6 0];
12 num2 = [-0.09846 -0.1683 -8.326 6.642 139.6 0 0];
13 denum = [1 5.927 30.52 83.11 163.6 176.6 93.29];
14 % gaus1
15 Hd1_comp = tf(num,denum);
16 % gaus 2
17 Hd2_nc = tf(num2,denum);
18
19 % Transfer functions for approximated 1st derivative 5th order filter:
20 % gaus1
21 Hd1 = tf([-0.1683 -8.326 6.642 139.6 0],denum);
22 % gaus 2
23 Hd2 = tf([-0.1683 -8.326 6.642 139.6 0 0],denum);
24
25 % Orthonormal representation for gaus2 (applies custom function orthonormal2)
26 h = orthonormal2([-0.1683 -8.326 6.642 139.6 0 0],denum)
27
28 %% 10th order 1st derivative filter
29 num=[-.44820e31 -.34446e32 +.44432e33 -.47081e34 .16977e35 -.52829e35 +.36809e31];
30 num2 = [num 0];
31 denum=[-.77104e30 -.15627e32 -.15857e33 -.10437e34 -.48808e34 -.16787e35 ...
32        -.42703e35 -.78851e35 -.10066e36 -.79880e35 -.29823e35];
33
34 Hgaus1_10 = tf(num,denum);      % gaus1
35 Hgaus2_10 = tf(num2,denum);    % gaus2
36
37 %Orthonormal representations
38 hgaus1 = orthonormal2(num,denum);
39 hgaus2 = orthonormal2(num2,denum);
```

The following function was developed by Sandro A. P. Haddad.

```
1 %calcula a representacao orthonormal do sistema a partir da funcao de
2 %transferencia
3 %ultima atualizaçao: 24/02/2012
4
5 function H = orthonormal2(num, den)
```

```

6 %% encontra a ordem do sistema
7
8 N=(length(den));
9 %% verifica necessidade de divisao do num pelo den
10
11 q = 0;
12 if(length(num)==N)
13     [q,r]=deconv(num,den); %realiza divisao para reduzir o grau do numerador
14     num = r(2:end);
15 end
16 %% deixa numerador com tamanho N-1
17
18 if(length(num)<(N-1))
19     num = [zeros(1,(N-1)-length(num)) num];
20 end
21 %% separa coeficientes de posicoes pares e impares do den
22
23 if(mod(N,2)==0) %den par
24     for i=1:ceil(N/2)
25         Dpar(i)=den(2*i);
26         Dimpar(i)=den(2*i-1);
27     end
28 else %den impar
29     Dimpar(1)=den(1);
30     for i=1:(ceil(N/2)-1)
31         Dpar(i)=den(2*i);
32         Dimpar(i+1)=den(2*i+1);
33     end
34 end
35 %Dpar
36 %Dimpar
37 %% decomposicao em fracao continuada
38
39 for i=1:N-1
40     %Dimpar
41     %Dpar
42     x(N-i)=Dimpar(1)/Dpar(1);
43     Dimpar=Dimpar(2:end)-x(N-i)*[Dpar(2:end),...
44         zeros(1,(length(Dimpar(2:end))-length(Dpar(2:end))))];
45     temp=Dimpar;
46     Dimpar=Dpar;
47     Dpar=temp;
48     %x(N-i)
49 end
50 %x
51 %% matriz A orthonormal
52
53 for i=1:N-2
54     A(i,i+1)=1/sqrt(x(i)*x(i+1));
55     A(i+1,i) = -A(i,i+1);
56 end
57 A(N-1,N-1)=-1/x(N-1);

```

```

58 %A
59 %% matriz B orthonormal
60
61 B = zeros(N-1,1);
62 B(N-1)=sqrt(abs(A(N-1,N-1))/pi);
63 %B
64 %% matriz F auxiliar
65
66 F(1,N-1)=sqrt(x(1)/pi)*den(N);
67 F(2,N-2)=F(1,N-1)/A(1,2);
68 for i=3:N-1
69     F(i,:)=( [F(i-1,2:end),0]+A(i-2,i-1)*F(i-2,:))/A(i-1,i);
70 end
71 %F
72 %% matriz C orthonormal
73
74 C=linsolve(F',(circshift(num',length(num))))';
75 %C
76 %% matriz D orthonormal
77 D = q;
78 %% Cria sistema orthonormal no espaço de estados
79
80 H = ss(A,B,C,D);

```