



AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

DIGITAL PULSE DETECTION FOR PARTICLE PHYSIC EXPERIMENTS

BY

JOSÉ MARÍA GÓRRIZ ARTIGOT

20105317

MASTER'S THESIS

IN

COMPUTER ENGINEERING

SUPERVISOR: ASSOCIATE PROFESSOR KIM BJERGE

Aarhus University School of Engineering

July 6, 2015

Abstract

A radiation detector is a device which can detect high energy particles and generate electrical charge pulses with different amplitudes. Nuclear physics applications extract from these pulses different parameters of interest like the energy and time of the particles. These calculated parameters are converted into digital format and they are acquired, saved and analyzed by a computer. This process has been traditionally done with analog devices. The aim of this thesis is to analyze and implement a model which will be able to estimate the energy and time from generated pulses.

In the first part artificial pulses are generated with an electrical pulse generator and Matlab to have available data to test the acquisition system. Then an overview of the implemented model is presented where two filters are considered to estimate the parameters of interest of the pulses. Afterwards the theory and the different implementations of the trapezoidal filter and the $CR - (RC)^n$ filters for orders from $n=1$ to $n=4$ are described in detail. The trapezoidal filter is used to calculate the energy of a pulse and the $CR - (RC)^n$ filters are used to estimate the arrival time of a pulse.

Finally the last part shows the test results where the accuracy of the different implementation of the filters is measured. On the basis of the results, it can be concluded that the trapezoidal filter generated with transfer function where the input pulse signal is based on a mathematical model is the most accurate on measuring the energy of a pulse with an accuracy of 100 %. On the other hand the $CR - (RC)^1$ is the most accurate within the $CR - (RC)^n$ filters for orders from 1 to 4 to measure the arrival time of a pulse with an accuracy of 50 ns.

Preface

This project was initiated by the Physics Department at Aarhus University. They have provided a detailed project description along with the existing analog pulse processing system and the filters used to calculate the energy, charge and timing of a pulse signal. This description [24] can be found in the enclosed CD.

The report is a proof of concept of the possible solutions when calculating the energy and the time of a pulse signal and it will be the first stage on implementing the data acquisition system digitally instead of the current analog methodology used at present.

This report has been made by José María Górriz Artigot during the Thesis Project of the Computer Engineering education at Aarhus University. The project is made in co-operation with the Physics Department.

I would like to thank my academic supervisor Kim Bjerge for his attention and for providing valuable feedback on this thesis project report. I also would like to thank him for his advice and guidance during the Self-Study course and the Research and Development project which I did under his supervision. I would like to thank the Physics Department for their collaboration. Finally I would like to extend my appreciation to my family and friends for their patience and support.

Table of Contents

Abstract	i
Preface	iii
Table of Contents	v
List of Figures	viii
List of Tables	x
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 Particle Physic Experiments	1
1.1.2 Analog Pulse Detector	3
1.1.3 Digital Pulse Detector	4
1.2 Problem Formulation	5
1.3 Thesis goal, Approach and Scope	6
1.3.1 Thesis Goals	6
1.3.2 Approach	6
1.3.3 Scope	7
1.4 Reading Guide	7
1.5 Related Work	9
1.5.1 Work done in previous research	9
1.5.2 Why does this thesis differ from previous work?	11
Chapter 2 Pulse Generation	13
2.1 Introduction	13
2.2 Exponential Functions	13
2.2.1 Exponential Decay	13
2.2.2 Exponential Growth	15
2.3 Deterministic Pulses Generated Using Matlab	15
2.3.1 Pulse with Fast Rise Time and Exponential Decay	15
2.3.2 Bi-exponential Pulse with Different Time Constants	16
2.3.3 Single Exponential Decay	16
2.4 Addition of Noise to Signals	17
2.5 Pulses Generated Using Random Pulse Generator	18
2.5.1 Experimental Setup	18
2.5.2 Random Generated Pulses	20

Table of Contents

Chapter 3 Energy and Time Adquisition System	23
3.1 Introduction	23
3.2 General Overview of the System	24
3.3 Moving Average Filter for Noise Reduction	26
3.4 Baseline Restoration	27
Chapter 4 Pulse Shape Analysis	29
4.1 Introduction	29
4.2 $CR - (RC)^n$ Shaper	29
4.2.1 Digital implementation in Time and Frequency Domain	30
4.2.2 Digital Implementation with Transfer Functions	32
4.2.3 Filter Test with Generated Pulses	32
4.3 Zero Crossing Calculation	33
Chapter 5 Puse Height Analysis	35
5.1 Introduction	35
5.2 Trapezoidal Filter	35
5.2.1 Digital Implementation with Recursive Algorithms	36
5.2.2 Digital Implementation with Transfer Functions	39
5.2.3 Filter Test with Generated Pulses	43
5.3 Least Square Fitting	45
5.4 Peak detection	47
5.5 Energy Calculation	48
Chapter 6 Results and Discussion	51
6.1 Introduction	51
6.2 Distortion of Filter Output by Gaussian Noise	52
6.3 Accuracy of $CR - (RC)^n$ Filters	53
6.4 Accuracy of Trapezoidal Filter with Fix Rise Time and Fix Decay Time	55
6.4.1 Trapezoid Accuracy of Single Exponential Decay Pulse	55
6.4.2 Trapezoid Accuracy of Bi-Exponential Pulse	56
6.5 Accuracy of Trapezoidal Filter with Rise Time and Decay Time Variations	57
6.5.1 Single Decay Exponential Pulse Test	57
6.5.2 Bi-Exponential Pulse Test	60
6.6 Pile Up Reduction	64
6.6.1 Minimize Pile Up by Reduction of Flat Time in Trapezoidal Output	64
6.6.2 Minimize Pile Up by $CR - (RC)^n$ filter	65
6.6.3 Accuracy $CR - (RC)^n$ Filter for Pile-Up	66
6.7 Summary of the Results	67
Chapter 7 Conclusion and Future Work	69
7.1 Introduction	69
7.2 Revisiting to the Goals of the Thesis	69
7.3 Evaluation of the Achievement of the Goals	69
7.3.1 Goal 1: Data acquisition model	70
7.3.2 Goal 2: Evaluate the proposed solutions	70
7.3.3 Goal 3: Improve transferable skills	70
7.4 Future work	71
7.4.1 Digital Pulse Detection Model	71

Table of Contents

7.4.2 Implementation in the FPGA	71
7.5 Final Remarks	73
Bibliography	75
A Development and System Specification	79
B Parameters of the Pulse	83
C Accuracy of the Implemented Filters	85
D Approach in the Implementation of the Transfer Function of the Trapezoidal Filter	91
E Pulses generated with Pulse Generator BNC	93

List of Figures

1.1	Mechanism of detection	2
1.2	Radiation detector located at Aarhus University (Department of Physics)	2
1.3	Analog measurement system for particle physics experiments located at Aarhus University (Department of Physics)	3
1.4	Block diagram of a traditional acquisition system for spectroscopy	3
1.5	Block diagram of a digital acquisition system for spectroscopy	4
1.6	Report Structure Diagram	9
2.1	Single exponential decay function	14
2.2	Pulse with fast rise time and exponential decay	16
2.3	Bi-exponential pulse with different time constants	16
2.4	Pulse with exponential decay	17
2.5	Pulse with ramp and exponential decay	17
2.6	Bi-exponential pulse with added Gaussian noise	18
2.7	Random Tail Pulse Generator BNC Model DB-2	19
2.8	Pulses generated with different rise and decay times	20
2.9	Pulses generated with different rise and decay times and pile up (at the bottom) .	21
3.1	Block diagram of the Pulse Detection System	24
3.2	Pulse where added Gaussian noise has been removed with a moving average filter	27
4.1	Step function responses	31
4.2	Frequency responses	31
4.3	$CR - (RC)^n$ shaper filter with degrees 1,2,3 and 4	33
4.4	Ampliation of $CR - (RC)^2$ filter	34
5.1	Convolution of single exponential function with rectangular function	36
5.2	Convolution of single exponential function with ramp function	37
5.3	Trapezoidal pulse to digitally synthesize	39
5.4	Trapezoidal filter output with recursive algorithms and TF implementations	41
5.5	Trapezoidal filter output with TF implementation	42
5.6	Signal after Trapezoidal filter (TF biexponential model based)	43
5.7	Bi-exponential Input signal $adc1_{10}$	44
5.8	Filter outputs with the different implemetations of the trapezoidal filter	44
5.9	Estimated bi-exponential pulse	47
6.1	Trapezoidal filter output with added Gaussian noise ($SNR = 20$)	52
6.2	Trapezoidal filter output with added Gaussian noise ($SNR = 5$)	52

6.3	Variations in trapezoidal output when $\tau_r \simeq 0$ and τ_d changes from 0 to $50 \mu s$ (Recursive Algorithms)	58
6.4	Variations in trapezoidal output when $\tau_r \simeq 0$ and τ_d changes from 0 to $50 \mu s$ (Transfer functions)	59
6.5	Variations in trapezoidal output when $\tau_r \simeq 0$ and τ_d changes from 0 to $50 \mu s$ (model based transfer functions)	59
6.6	Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (Recursive Algoritms)	60
6.7	Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to $50 \mu s$ (Recursive Algoritms)	60
6.8	Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (TF single decay)	61
6.9	Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to $50 \mu s$ (TF single decay)	61
6.10	Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (TF single decay)	61
6.11	Variations in trapezoidal output when $\tau_r = 0.001 \mu s$ and τ_d changes from 0 to $50 \mu s$ (TF single decay)	61
6.12	Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (Biexponential transfer function)	62
6.13	Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to $50 \mu s$ (Biexponential transfer function)	62
6.14	Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (Biexponential transfer function)	62
6.15	Variations in trapezoidal output when $\tau_r = 0.001 \mu s$ and τ_d changes from 0 to $50 \mu s$ (Biexponential transfer function)	62
6.16	Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (Model based transfer functions)	63
6.17	Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to $50 \mu s$ (Model based transfer functions)	63
6.18	Trapezoidal outputs with different flat times	65
6.19	Input signals with pile-up and $CR - (RC)^n$ filter outputs	65
7.1	Process of the system	72
A.1	Functional Requirements Diagram	80
A.2	Design Constraints Table	81
B.1	Parameters of a pulse	83

List of Tables

6.1	Table with trapezoid heights (T H) where noise has been added	53
6.2	Table with test results from $CR - (RC)^n$ filters with input signal $adc1_{10}$ where rise time = $5 \mu s$ and decay time = $10 \mu s$	54
6.3	Table with test results from $CR - (RC)^n$ filters with Matlab generated input signals where rise time = 0.1 to $5 \mu s$ and decay time = 5 to $20 \mu s$	55
6.4	Table with accuracy percentage of trapezoidal filter with Matlab generated input signal where rise time $\simeq 0$ and $\tau_d = 15 \mu s$	56
6.5	Table with accuracy percentage of trapezoidal filter with Matlab generated input signals where rise time = 0.1 to $5 \mu s$ and decay time = 5 to $20 \mu s$	57
6.6	Table with errors of trapezoidal filters when decay time constant changes in single exponential pulse	59
6.7	Table with test results from trapezoidal filters when $\tau_r=0.001 \mu s$	63
6.8	Table with test results from trapezoidal filters when $\tau_r=1 \mu s$	64
6.9	Table with test results from $CR - (RC)^n$ filters with input signal $adc1_{16}$ where rise time = $2 \mu s$ and decay time = $20 \mu s$	66
6.10	Table with Peak Estimation Average Time Offset from tables 6.2, 6.3 and 6.9	67
C.1	Table with test results from $CR - (RC)^n$ filters with $adc1_{10}$ bi-exponential pulse . .	85
C.2	Table with test results from $CR - (RC)^n$ filters with $PE1$ bi-exponential pulse . .	85
C.3	Table with test results from $CR - (RC)^n$ filters with $PE6$ bi-exponential pulse . .	85
C.4	Table with test results from $CR - (RC)^n$ filters with $PE15$ bi-exponential pulse . .	86
C.5	Table with test results from $CR - (RC)^n$ filters with $PE21$ bi-exponential pulse . .	86
C.6	Table with test results from $CR - (RC)^n$ filters with $PE26$ bi-exponential pulse . .	86
C.7	Table with test results from $CR - (RC)^n$ filters with $PE31$ bi-exponential pulse . .	86
C.8	Table with test results from $CR - (RC)^n$ filters with $PE41$ bi-exponential pulse . .	86
C.9	Table with test results from $CR - (RC)^n$ filters with $PE50$ bi-exponential pulse . .	86
C.10	Table with test results from $CR - (RC)^n$ filters with $PE71$ bi-exponential pulse . .	87
C.11	Table with accuracy percentage of trapezoidal filter with $adc1_{10}$ bi-exponential pulse	87
C.12	Table with accuracy percentage of trapezoidal filter with $PE1$ bi-exponential pulse .	87
C.13	Table with accuracy percentage of trapezoidal filter with $PE6$ bi-exponential pulse .	87
C.14	Table with accuracy percentage of trapezoidal filter with $PE15$ bi-exponential pulse	88
C.15	Table with accuracy percentage of trapezoidal filter with $PE21$ bi-exponential pulse	88
C.16	Table with accuracy percentage of trapezoidal filter with $PE26$ bi-exponential pulse	88
C.17	Table with accuracy percentage of trapezoidal filter with $PE31$ bi-exponential pulse	88
C.18	Table with accuracy percentage of trapezoidal filter with $PE41$ bi-exponential pulse	88
C.19	Table with accuracy percentage of trapezoidal filter with $PE50$ bi-exponential pulse	89
C.20	Table with accuracy percentage of trapezoidal filter with $PE71$ bi-exponential pulse	89

Chapter 1

Introduction

This chapter presents the work carried out during this thesis. It starts with a description of the equipment for pulse detection and the data acquisition system in the background in section 1.1, a problem formulation in section 1.2 and the thesis goals in section 1.3.1. Finally, the structure of the thesis is described in section 1.2 and the related work is described in section 1.5.

1.1. Background

This section is divided in two main parts with regards to the thesis background, presenting the operation of the analog and digital pulse detector and the equipment pulse detection located at Aarhus University.

1.1.1 Particle Physic Experiments

This section describes the radiation detector and the analog measurement system for particle physics experiments located at the Department of Physics of Aarhus University.

Nuclear spectrometry signal [4] processing has the goal of producing digital signals that describe the properties of radiation particles. The energy of radiation is measured in such a way that its energy is absorbed in solid state and converted into a pulse with electrical charge. All electrical pulses are contaminated with noise when they pass through the electronic components.

Apart from the energy, pulses also contain information about its position, the type of particle, arrival time, etc. In High Energy Physics [17] the type of detector used for the measurements depends on the type and the energy of the particle. There are different types of radiation [29] and they can be classified by the following criteria:

- The sensitivity of the detector depends on the type of radiation that it will detect. For example solid scintillation detectors are not used to detect α -particles because these particles cannot penetrate the covering of the detector.
- The energy resolution depends on the precision of the measurements. How accurate the detector can measure the arrival of the particles.
- The time resolution of the detector depends on how high counting rate the detector can measure without errors.

- The detector efficiency. If 100 α -particles will strike the detector , how many of them will be detected.

For example the silicon strip detector is used when charged particles such as α -particles and protons have to be detected. Figure 1.1 shows the composition of a silicon strip detector.

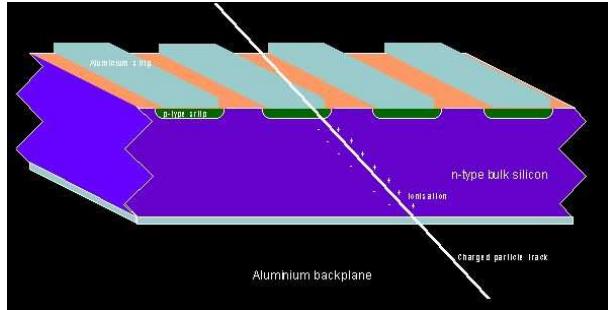


Figure 1.1: Mechanism of detection [2]

Most of the silicon detectors [2] contain n type bulk material. There is an aluminium plate which covers all the surface at the bottom. There are p type silicon strips implanted in the surface at the top which are separated by a thin insulator from the p type silicon strips. An electric field is then applied between those p strips and the bottom of the detector.

When a charged particle passes through the detector it ionizes the bulk of the silicon. This frees electrons from the atoms of the silicon and leaves these atoms with a vacancy of electrons. These vacancies are named "holes". The electric field between the aluminum strips and back surface, makes the holes flow towards the p-type strips which are negatively charged and the electrons toward the back surface. The aluminum strips are connected to electronic channels which are sensitive to electric fields. By checking which electronic channel has been active, it is possible to determine in which part of the detector passed the charged particle.

Figure 1.2 shows the radiation detector where the cable situated at the center of the metal plate (between the white sticks) is the input channel for the radiation particles.



Figure 1.2: Radiation detector located at Aarhus University (Department of Physics)

Figure 1.3 shows the analog measurement system available at Aarhus University. The system contains a considerable amount of analog cables and it takes much space as it can be observed in the picture. The task proposed by the Department of Physics was to change this costly analog system by a digital implemented in one or more FPGA's. The digital pulse processing is a cheaper replacement of the expensive analog processing.

Analog Pulse Detector

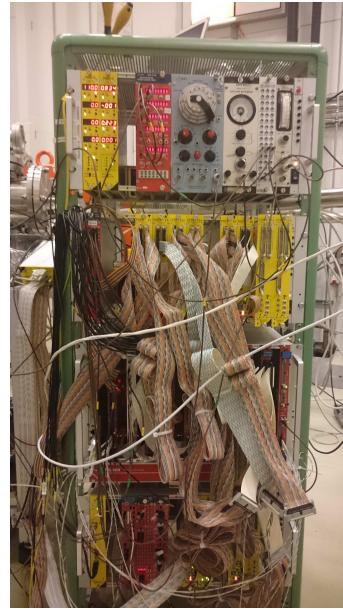


Figure 1.3: Analog measurement system for particle physics experiments located at Aarhus University (Department of Physics)

1.1.2 Analog Pulse Detector

This section describes the operation of the analog pulse detector. Traditionally, the calculation of the energy and the time [24] from pulse signals generated by a particle detector has been done with interconnected analog blocks having specific functions. The parameters are calculated analogically by the different blocks and the conversion from analog to digital is performed just before those parameters have to be transferred to the PC station.

Figure 1.4 illustrates the transition of the pulse signal from its generation by the detector, passing through the amplifiers and going to the different analog blocks which will calculate the parameters of interest.

As illustrated in figure 1.4, the analog chain contains at the beginning two preamplifiers close

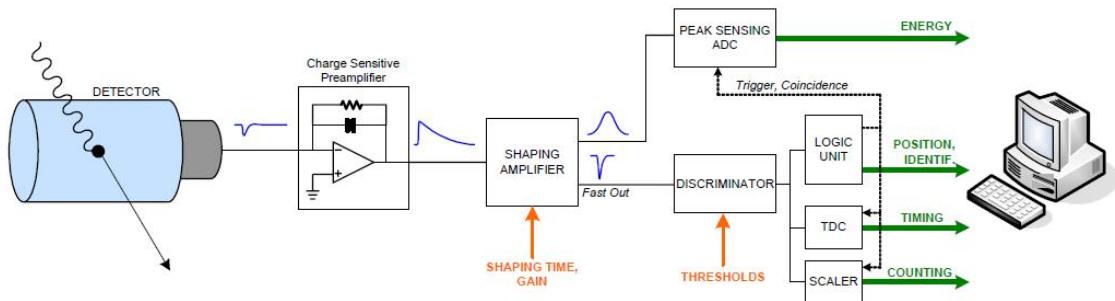


Figure 1.4: Block diagram of a traditional acquisition system for spectroscopy [24]

to the detector which amplify the pulse signal generated so that it can be transmitted through the cable to the analog blocks.

The first preamplifier, called charge sensitive preamplifier, converts the pulse signal into a signal with voltage amplitude, fast rise time and exponential decay where the energy is proportional to the pulse height.

The second preamplifier, called current sensitive preamplifier, treats the rising edge of the signal generated by the first amplifier and produces an output which can be used to estimate the time when the pulse arrives. This preamplifier produces a very short pulse and it does not change the shape of the signal which is normally used by the charge ADC to calculate the energy. This energy is proportional to the area below the signal. An integrating window is necessary in order to calculate the energy of the signal and this can be provided by the system when it knows in advance that the pulse is going to be generated. However most of the time this fact is unknown so the signals have to be split and the fast part is sent to the discriminator so that the integrating window can be generated. The signal which goes to the charge ADC has to be delayed to fit in the generated window.

1.1.3 Digital Pulse Detector

This section describes the operation of the digital pulse detector. The main idea of this thesis is to convert the analog measurement system of figure 1.4 into the digital measurement system of figure 1.5 where the analog blocks are substituted by a FPGA. Although the function of the digital system is the same as the analog, the implementation is different and some of the terminology and concepts are different.

Figure 1.5 shows the purposed digital measurement system for particle physics experiments.

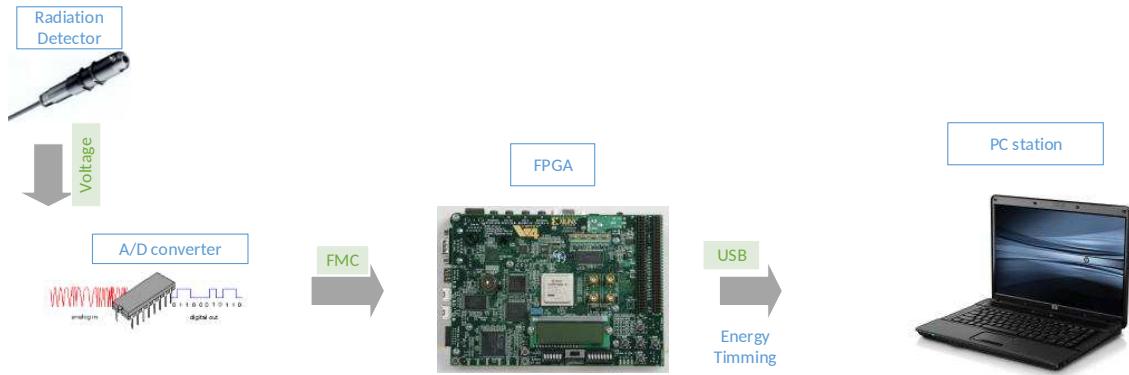


Figure 1.5: Block diagram of a digital acquisition system for spectroscopy

The digital system contains the following components:

- **Radiation detector:**
It detects the physic particles and generates a voltage signal $x(t)$ with fast rise time and exponential decay. The voltage of the signal depends on the type of particle detected.
- **Analog to digital converted:**
It converts the analog signal generated by the radiation detector into a digital signal $x(i)$ composed by samples where i represents the sample number. The A/D converter should be fast enough to sample the pulse accurately without loosing information. For this reason *a fast A/D converter [33] should have a resolution of 10 - 12 number of bits and sampling frequency 50 - 200 MSPS, for accurate energy measurement.*
In the digital acquisition system the A/D conversion occurs very close to the detector where in the analog acquisition system occurs close to the PC station.
- **FPGA (field-programmable gate array):**
It gets the digital sampled signal from the A/D converter and store the samples in the mem-

ory for processing. The energy and the time of the pulses are calculated by the FPGA and the parameters are presented at the output.

- PC Station: It stores the parameters calculated by the FPGA. The data can be used for different proposes depending on the needs of the research.

Some of the advantages that digital systems offer are:

- A single FPGA board can have the same functionality that all the analog devices which reduces cost and space.
- Digital systems are more stable and allow storing the data which can be reviewed later for analysis.
- Digital techniques allow better elimination of the noise introduced in the signal though the implementation of filters and allow better correction of pile-up and baseline fluctuation effects. Pile-up will be explained in section 6.6 of chapter 6 and baseline fluctuation in section 3.4 of chapter 4
- Digital systems allow real time operations to be executed at a higher counting rate. Due to this high data rate it is not possible to transfer data to the computer at real time so the calculation of the parameters should be performed in the FPGA.
- FPGA are more flexible and changes can be introduced in the system by modifying the code and adding or deleting hardware blocks.

However digital systems have some disadvantages like for example:

- The signal requires pre-filtering due to ADC limitations.
- Design procedure can be complex (requires knowledge of C/C++, VHDL/Verilog and Mat-lab).
- Implementation of hardware and software can be expensive.

1.2. Problem Formulation

Nuclear spectrometry systems [4] contain electronic devices and circuits which process the electrical signals generated by radiation detectors. This electronic devices are composed by subsystems that amplify, filter and shape these electrical signals to generate digital information about the type of the radiation stimulated by the radiation detector. These devices are designed to obtain the maximum information about the radiation with the highest possible accuracy.

In the past electronics devices has been composed of analog components. The performance delivered by these components has increased over time because more advanced electronics designs have been developed and implemented. However the complexity of the technology is increasing and devices are becoming faster, smaller and cheaper to produce. This leads to the need of developing systems which adequate to these changes and are able to be more effective and accurate. The development of FPGA's and digital signal processing methods has allowed the replacement of analog devices by digital electronics.

There are already in the market [19] devices called waveform digitizers which are able to calculate some parameters of interest from analog pulse signals. They store a certain number of samples when a trigger occurs into a buffer and they have FPGA's or DSP blocks which perform real time calculations based on those samples.

The thesis work will focus on developing a model for the acquisition of the energy and the time from a pulse. This model will be used to implement a digital system which can replace the current solutions which are implemented in analog where all pulse processing is done in analog domain and the ADC's are connected to simple data relay board for computer communication. Basically the effort will be based on the implementation of the model which is the first step on converting those analog solutions into digital solution. The model will not be implemented in the FPGA in this thesis but it will be verified in Matlab.

1.3. Thesis goal, Approach and Scope

This section describes the Master's thesis goals, the approach used and the scope of the work.

1.3.1 Thesis Goals

Goal 1: To propose a data acquisition model for experimental physics applications that will be able to calculate the time and the energy of pulse signals generated by a pulse generator.

Goal 2: To evaluate which of the proposed solutions is more accurate on estimating the energy and the time of the pulse.

Goal 3: Improve personal skills in general for being able to conduct research by identifying transferable skills that may be reusable in the future.

This thesis does not have the ultimate goal of implementing the system for industrial solutions.

1.3.2 Approach

The methodology applied in this thesis is structured in three phases.

Phase 1: Research and analysis

The first phase covers the research done in order to understand how the analog system works with the corresponding filters used to calculate the different parameters. Also research is carried out about previous work done, like projects and research papers, in relation to the implementation of analog and digital measurement systems.

An analysis about the different possible digital implementations is performed after the traditional analog system is understood.

Phase 2: Design and implementation

The second phase covers the design of the data acquisition system in terms of the different blocks where the system will be implemented in. Different exponential pulse signals are generated in order to have input data for testing. A model is done in order to analyze how the different implemented filters behaves with the input signals generated. A decision is arguably to be the best implementation design based on the data obtained from this model and the solutions available.

Scope

Phase 3: Test, verification and deployment

The third part covers the testing and verification of the implemented filters and here some measurements will be conducted to test the accuracy of the different implementations. The system has to be able to measure the energy and the time of a pulse. Based on the results of the mentioned measurements, decisions are reached as to find which filter is the most accurate.

1.3.3 Scope

This work will be focused on:

Proposing a measurement system which can measure energy and time of an electrical pulse signal.

Making decisions about the best filters to implement based on previous work done by others during the development of the system.

Focusing on the accuracy of the calculated parameters and compare the different implemented filters.

1.4. Reading Guide

The convention used in this thesis are presented in this section.

References

References are referred to by numbers, which correspond to numbers in the **Bibliography** of this thesis. As an example [5] refers to the reference number 5.

Emphasis

Words which are emphasized are written in *italic*, such as *emphasized*.

Quotations

A quotation from literature is written in *italic* and placed between double quotation, such as "*This is a quote*".

Figures

Figures are presented using the convention **Figure <C>.<N>**, in which C refers to the chapter number, and N is the order of the figure in chapter C. An example is the figure shown in figure 1.1, which is the first figure in chapter 1.

1.4.0.1 Report Structure

The suggested reading of this thesis is shown in figure 1.6. In this figure every chapter of this thesis is a solid box. Furthermore, the arrows between the solid boxes indicate the dependencies between the chapters.

The project is divided in 6 chapters:

Chapter 2: This chapter introduces the process of generating exponential pulse signals. The signals are generated by single exponential functions which contain a decay time constant τ_d and by bi-exponential functions which in addition to τ_d contain a rise time constant τ_r . These signals are generated by a signal generator and Matlab.

Chapter 3: This chapter contains a general overview of the developed system. It is possible to jump to chapter 6 from this chapter if the reader do not want to read a detailed description about how the energy and the time are calculated in chapters 4 and 5.

Chapter 4: This chapter describes the process of estimating the time when the pulse signal arrives. The time is calculated by the implementation of the $CR - (RC)^n$ filters for order from 1 to 4 which convert the exponential pulse into a bipolar pulse. The filters are generated by the implementation of transfer functions.

Chapter 5: This chapter describes the process of estimating the energy of the pulse. This is calculated by the implementation of the trapezoid filters which convert the exponential pulse into a trapezoid with flat top. The filters are generated by the implementation of recursive algorithms and transfer functions.

Chapter 6: This chapter contains the accuracy of the filters implemented in chapters 3 and 4. The filters are tested with the signals generated in chapter 2 and the results are presented in tables. The results of those tests are used to determinate the precision of the filters.

Chapter 7: This chapter concludes the work conducted in this thesis and the goals presented in section 1.3.1 are evaluated. Finally, possible future work directions are discussed.

Appendix A: This appendix shows the system specification which is used to make an exact set of requirements based on the needs of the system to be developed.

Appendix B: This appendix contains a description of the pulse parameters used during the development of the thesis.

Appendix C: The appendix shows the tables containing all the pulses which have been provided by Kim Bjerre from Aarhus University.

Appendix D: The appendix shows a new approach to reduce the transfer function implemented with a single exponential pulse from chapter 4.

The Master thesis report is also attached on the CD, if the reader wishes to have it as a pdf file.

Related Work

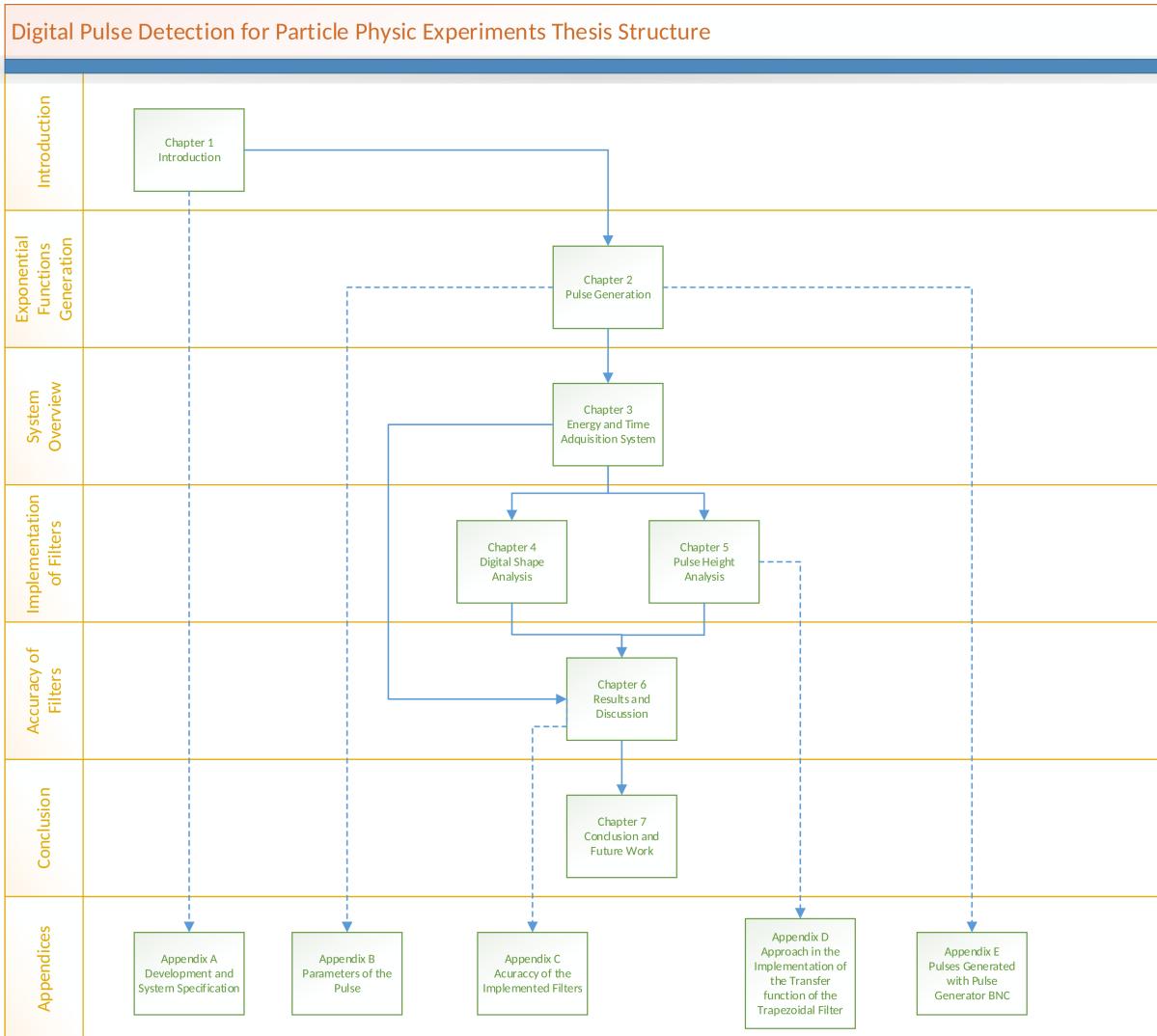


Figure 1.6: Report Structure Diagram

1.5. Related Work

This section is a review of the work done by others related to digital pulse detection. It identifies and evaluates past approaches to the problem.

1.5.1 Work done in previous research

The research has been focused on finding the different implementations of the filters used to calculate the energy and the time of a pulse signal. The trapezoidal filter is used to estimate the energy and the $CR - (RC)^n$ filter is used to estimate the time when the pulse arrives in most of the searched literature.

The topic is very specific and the research which have been conducted are mostly from Universities. There have been found books which are specialized in signal processing and books which describe the physical part of the pulses and the issues when dealing with them.

The information from research papers, books and previous thesis can be classified in four categories:

- Information about the different type of signals which include digital signal processing, nuclear physics and digital electronic literature.
- Theoretical information about how the outputs of the filters are mathematically calculated and generated by recursive operations in time domain. This includes mostly research papers from different Universities.
- The transfer functions in z-domain which generate the outputs of the filters. This includes mostly research papers.
- The graphs showing the filter outputs which have been implemented in hardware which includes projects and papers from private companies.

Mostly all the available information is very theoretical and it has been carefully read and sorted out during the development process. There is almost no information of the process about how to properly implement those filters so all the equations have to be well understood before starting with the implementation in Matlab.

The most relevant literature will be described and discussed in the following.

- Signals:

There are abundant literature of digital signal processing [12] [42] where it is described the theory behind the filters, the different type of signals and the implementation in Matlab [27]. In [26] deterministic pulse signals are described in detail and exponential signals are illustrated in graphs.

Deterministic pulses can be generated from different models of exponential equations. In [10] bi-exponential pulses are generated by two equations. The first equation generates the rising slope of the signal and the second equation generates the decaying slope. The equation have four unknown values which have to be calculated by large mathematical operations adding difficulty to the implementation so this method is not implemented by this reason. In [43] a model of bi-exponential equations is described and it is composed by a single equation where the amplitude, the rise time constant τ_r and the decay time constant τ_d can be adjusted to generate the desired pulse signal. This model is implemented in section 2.3 of chapter 2 for its simplicity.

- $CR - (RC)^n$ Filter:

The estimation of the arrival of the pulse signal by weighted least square is described in [32]. This algorithm [9] is difficult to implement in real time and it is not chosen for this reason.

The $CR - (RC)^n$ filter [3] is mostly used in digital pulse detection to estimate the arrival time of the pulse. The theory of the filter is described in [30] and [34] where the analog circuit composed by a high and a low pass filter is explained.

The transfer functions implemented in section 4.2.2 of chapter 4 to generate the $CR - (RC)^n$ outputs are described in [21]. Also this paper presents the recursive algorithms which generate these transfer functions and shows the filter outputs. In [31] and [8] it is described the process of extracting the arrival time from the bipolar signal with the zero crossing function. This function is implemented in section 4.3 of chapter 4.

- Trapezoidal Filter:

The accuracy of the measured energy depends on the shape of the signal that it is used so

Why does this thesis differ from previous work?

research has been carried out to find the best shape. It can be shown strictly mathematically that the optimal shape to estimate the energy is so called cusp which gives the best signal to noise ratio SNR as described in [22]. It means that changing the exponential signal into cusp and calculate the height of the cusp, the fluctuation of the result will be minimal (comparing to all other shapes and assuming the same standard deviation of the noise). However the cusp cannot be synthesized because it is infinite in the past and in the future as described in [28] and for this reason it is not implemented. In [16] the trapezoidal and triangular shapes to estimate the energy from a pulse are presented and the recursive algorithms for these shapes are described. Changing the shape into triangular then fluctuation will be a little larger comparing to the cusp (around 7 %) and changing the shape into trapezoidal it will be little more larger than triangular. The trapezoidal [28] is used because of longer charge collection time in the case of High Purity Germanium [29] detector. The optimum pulse shape for a finite pulse duration is the triangular shape [15], when the signal contains delta noise. Changing shape into Gaussian [14] the fluctuation will be even larger. Based on the above observations the trapezoidal filter is chosen to be implemented because the signals generated to test the filters do not contain delta noise. Gaussian noise has been added to some of them to test the accuracy of the filters when unwanted noise is present at these signals.

The transfer functions which will generate the trapezoidal filter are implemented in section 5.2.2 of chapter 5 and they are described in [11]. The paper only shows the final transfer functions so all the steps to obtain these transfer functions has not been obtained from any source of information and they have been calculated in this thesis. The accuracy of the trapezoidal filter implemented with these transfer functions decreases when the rise time of the input pulse signal increases. In order to improve the accuracy of the filter a method called, transfer function model based, has been developed in this thesis where the transfer functions from [11] are used.

In [5] the process of implement the trapezoidal filter bu using Simulink is described. The final transfer function when the input is a single decay pulse is shown in the paper. The process of obtaining this transfer function is described in appendix D and it is not obtained from any source of information.

1.5.2 Why does this thesis differ from previous work?

This subsection describes what it has been done in this thesis that has not been done in previous work.

The thesis and research papers written by master or PhD students are focused on a single topic of the digital pulse detection. They describe the behavior of a single implementation of a filter and some of them try to prove if it is feasible to implement the trapezoid or the $CR - (RC)^n$ filters in a FPGA. Most of the work done so far describes whether the mathematics behind the filters or the behavior of them through graphics once they are implemented in hardware. In the thesis it is explained the equations and transfer functions and it uses this theory to implement the filters in Matlab.

This thesis performs different implementations of the same filter.

The trapezoidal filter is generated by recursive algorithms in time domain and by transfer functions in Z domain.

Four $CR - (RC)^n$ filters are implemented with order from 1 to 4 and they are generated by transfer functions in Z domain.

The accuracy of those filters have been tested with single and bi-exponential pulses. The results have been compared in order to determinate the best implementation to obtain the desired param-

eters from the pulse. Also the trapezoidal filters designed to deal with single exponential pulses have been tested with bi-exponential pulses and vice versa. This accuracy comparison between the different implementation of the filters mentioned above has not been done before.

A new method has been implemented in order to improve the accuracy of the trapezoidal filter. The method, implemented for the first time in this thesis, can be used to estimate the energy of single and bi-exponential pulses with an accuracy of nearly 100 %.

Chapter 2

Pulse Generation

2.1. Introduction

This chapter presents how to generate different types of pulses. There have not been provided real data from a particle detector so pulses have to be generated in Matlab and by a signal generator in order to have some signals to test the filters which will be implemented in chapters 4 and 5.

The first part presents the theory behind the exponential decay functions and the exponential growth functions. All pulses are generated by exponential functions or they can be approximated to a model of exponential functions.

Afterwards two main types of pulses are shown:

- Deterministic pulses generated in Matlab by equations. Equations with one time constant (τ_d) generate single decay exponential pulses and equations with two time constants (τ_d and τ_r) generate bi-exponential. τ_d represents the decay time constant and τ_r represents the rise time constant.
- Random pulses generated by a pulse generator where the rise time, decay time and frequency are set prior to generation.

The different types of noise are addressed together with a description of the process performed to add noise to signals.

2.2. Exponential Functions

Exponential functions are used to mean the natural exponential function e^x , where e is the Euler's number and its value is 2.718281828. The variable x can be a real or complex number and the derivative of e^x is the same function.

The exponential function [41] is widely used in physics, chemistry, engineering, mathematical biology, economics and mathematics.

2.2.1 Exponential Decay

A quantity [41] is subject to exponential decay if it decreases at a rate proportional to its current value. This is illustrated in the following differential equation:

$$\frac{dN}{dt} = -\lambda N \quad (2.1)$$

where N is the quantity and λ (lambda) is the exponential decay constant which is a positive rate. Solving the equation it is obtained:

$$N(t) = Ae^{-\lambda t} \quad (2.2)$$

where $N(t)$ is the value of the equation at time t and A is the initial value when t is 0.

2.2.1.1 Estimation of Decay Rate

The following equations [41] show the relation between the exponential time constant τ_d and the decay rate λ , where the number of discrete elements $N(t)$ are decaying exponentially by the decay rate.

$$\tau_d = \frac{1}{\lambda} \quad (2.3)$$

Then it can be written equation 2.10 in terms of the exponential time constant τ_d , instead of the decay constant λ and it is obtained:

$$N(t) = Ae^{-t\tau_d} \quad (2.4)$$

In figure 2.6 it can be observed that τ is the time at which the voltage is reduced to $V_o/e = 0.367879441$ times its initial value. If the initial voltage V_o at time $t=0$ is 1 Volt then at time $t = \tau_d$ will be 0.368 Volts.

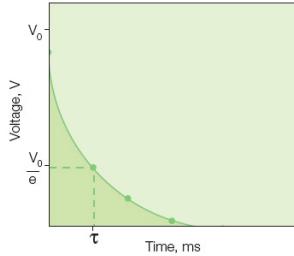


Figure 2.1: Single exponential decay function

Two methods are implemented to estimate the time constant τ_d characterizing the exponential decay from the measurements.

- The first method, based on the principle mentioned above, it is a function which is checking the output values until it is equal to $0.368 * \text{initial value}$. When that value is found then it returns the value of the time t at that time.
- The second method uses the least square fitting method to find the value of τ_d . This will be explained in detail in section 5.3 of chapter 5.

2.2.2 Exponential Growth

Exponential growth [41] occurs when the growth rate of the value of a mathematical function is proportional to the function's current value.

Exponential growth can be written:

$$N(t) = Ae^{\lambda t} = Ae^{t\tau_r} \quad (2.5)$$

where all the parameters are the same as in the exponential decay except τ_r and λ which are positive. It can be concluded that exponential growth occurs in the same way as with exponential decay but the decay rate is called growth rate because it is positive.

The time constant τ_r is the time it takes to grow by a factor e . This means that τ_r represents the time it takes the voltage to reach $V_o - V_o/e = 0.632120559$. If the initial voltage V_o at time $t=0$ is 1 Volt then at time $t = \tau_r$ will be 0.632120559 Volts.

2.3. Deterministic Pulses Generated Using Matlab

This section introduces the pulses implemented in Matlab which will be used in chapters 4, 5 and 6 to test the implemented filters. Filters respond to these signals by producing other signals. Deterministic pulses can be modeled exactly by a mathematical model and they are not always adequate to model real-world situations. These pulses are ideal pulses in the sense that they can be modified and adapted to test all possible scenarios that filters can face. They are functions with one or more independent variables and they are implemented in time domain.

Three types of exponential decay pulses have been generated with Matlab to have different data in order to test the behavior of the implemented filters.

2.3.1 Pulse with Fast Rise Time and Exponential Decay

The pulse is generated by two exponential functions with different time constants and where the difference between them is the sign of the exponential.

Exponential growth function

$$x_1(t) = A * e^{t_1/\tau_r} \quad (2.6)$$

Exponential decay function

$$x_2(t) = A * e^{-t_2/\tau_d} \quad (2.7)$$

where A, τ_r and τ_d are equal or greater than zero.

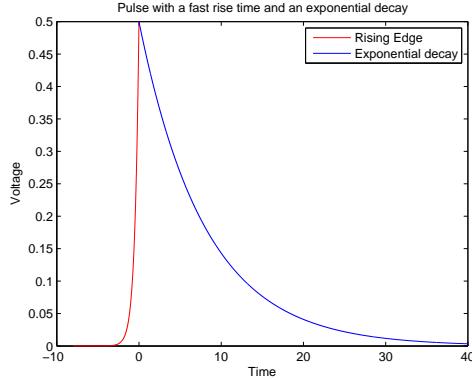


Figure 2.2: Pulse with fast rise time and exponential decay

Equation 2.6 corresponds to the growing part of the signal which is colored red and equation 2.7 corresponds to the decaying part which is colored in blue. The point where both equations meet is the zero in x-axes so the time t_1 of the first equation has negative values and the time t_2 for the second one has positive values.

2.3.2 Bi-exponential Pulse with Different Time Constants

A bi-exponential function is a function which contains the sum or the difference of exponential functions. In Matlab the pulse is generated by a subtraction of two exponential functions with different time constants which make the top of the pulse shaper as it can be observed in figure 2.3.

$$x(t) = A * (e^{-t/\tau_d} - e^{-t/\tau_r}) \quad (2.8)$$

where A, τ_r and τ_d are equal or greater than zero.

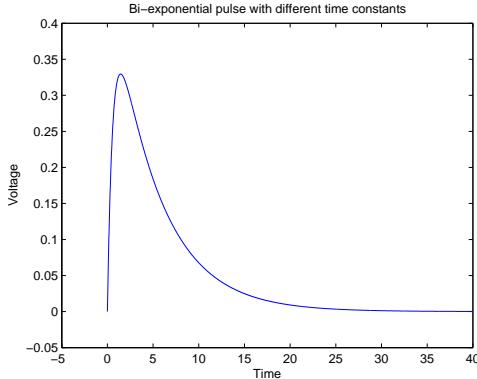


Figure 2.3: Bi-exponential pulse with different time constants

2.3.3 Single Exponential Decay

A single exponential decay is a function which contains only one exponent with a negative time constant. Two different versions of this function are implemented in this section.

Single decay function

The pulse is generated by a single decay function where the rise component is a horizontal line defined by the equality 2.9 and the rise time is equal to one unity. The value of A must be equal

Addition of Noise to Signals

to the value $Rise$ so that both equations can meet in the x-axis with value 20.

$$Rise = [20] \quad (2.9)$$

$$x(t) = A * e^{-t/\tau_d} \quad (2.10)$$

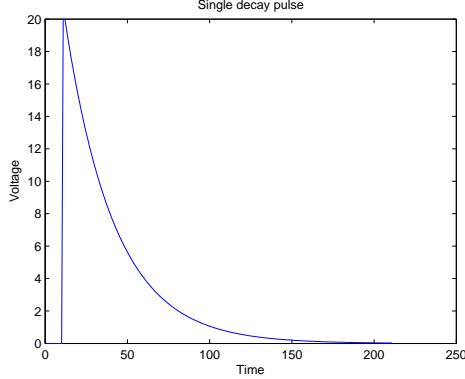


Figure 2.4: Pulse with exponential decay

Ramp function

The pulse is generated by a ramp function which contains a ramp and a single decay exponential function. The difference with the single decay function is that the rise time is equal to 20 time units as it can be observed in figure 2.5.

$$Ramp = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20] \quad (2.11)$$

$$x(t) = A * e^{-t/\tau_d} \quad (2.12)$$

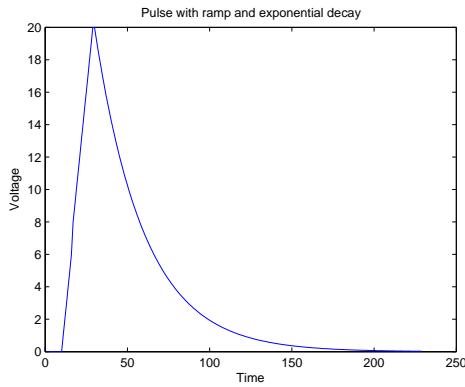


Figure 2.5: Pulse with ramp and exponential decay

2.4. Addition of Noise to Signals

In signal processing, noise [41] is a general term for unwanted (and, in general, unknown) modifications that a signal may suffer during capture, storage, transmission, processing, or conversion. This noise can be classified by its properties (color of the noise) and by how it modifies the signal:

- Additive noise which gets added to the intended signal like for example white Gaussian noise
- White noise, pink noise, black noise
- Gaussian noise, etc

In the simulation Gaussian white noise has been added to the pulses by using the Matlab build-in function "awgn".

The function is loaded with the parameter x (input signal) and the parameter SNR (signal to noise ratio):

$$Y = awgn(x, SNR) \quad (2.13)$$

The SNR per sample in dB can be chosen using the expression:

$$snr_{db} = 20 * \log \frac{A_{signal}}{A_{noise}} \quad (2.14)$$

where A_{signal} is the amplitude of the input signal and A_{noise} is the amplitude of the added noise. Figure 2.6 illustrates the bi-exponential pulse (on the top) and the same signal after Gaussian white noise with signal to noise ratio = 50 has been added (on the bottom).

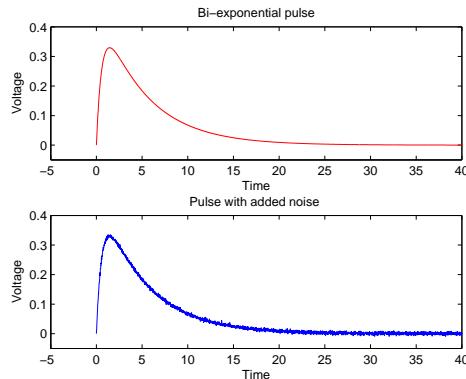


Figure 2.6: **Bi-exponential pulse with added Gaussian noise**

2.5. Pulses Generated Using Random Pulse Generator

After having gone through the theoretical pulses, this section will address the generation of pulses using an electrical pulse generator. These signals have been generated by a pulse generator with different rise and decay times and they have been provided by Kim Bjerge for testing proposes.

2.5.1 Experimental Setup

The Random Tail Pulse Generator BNC Model DB-2 has been used to generate the pulse signals as illustrated in figure 2.7.

Experimental Setup



Figure 2.7: **Random Tail Pulse Generator BNC Model DB-2**

The Model DB-2 [7] is a pulse generator which accurately simulates the random and pileup characteristics of pulses from a radiation detector. It provides pulses which are monoenergetic over a broad range of average count rates. Under high count rate conditions, the pulses will pileup. This characteristic is useful in determining pile-up or count-rate effects and in measuring the resolution of high count rate spectroscopy systems.

The pulse generator has two modes of operation:

- Repetitive mode where it generates the same signal which repeats itself. The output range is from 0 to 10 V and it is useful to test stability and linearity.
- Random mode where it generates random signals. The output range is from 0 to 1 V and it is useful to produce a larger number of pulses to pile up on the tails of each other without saturation.

A typical application [7] of the random pulse mode is to connect the output of the Model DB-2 to a test capacitor input of a preamplifier in a pulse height analyzer system. The count rate is set low, about 1 kHz and the pole-zero adjustment of the post-amplifier is made for minimum broadening of the line width on the analyzer. Then the count rate is increased until the broadening exceeds an acceptable value.

The most relevant features of the DB-2 pulse generator are:

- The frequency of the signals can be adjusted between 10 Hz and 1 MHz
- Random or repetitive modes
- The rise time has a range between 100 ns and 20 μs and can be adjusted in 8 steps.
- The decay time has a range between 5 and 1000 μs and can be adjusted in 8 steps.

- Output Voltage Transition Range Repetitive 0 - 10 V or 0 - 0.25 V Random 0 - 0.25 V

- Output Polarity Positive or Negative

2.5.2 Random Generated Pulses

Random pulses cannot be described always by a mathematical model like the deterministic ones as it will be seen in section 5.2.3.

Some of the pulses are shown in this section, however all the pulses which have been provided in .txt format are illustrated in table E. These pulses have a rise time range between 0.1 and 0.5 μs and they are all combined with decaying times of 5, 10, 20 and 50 μs . Moreover additional random pulses with rise time of 5 μs and fall time of 50 μs have been provided too.

Figure 2.8 shows some of those signals with the rise time represented by the letter r and where the rise time value is in microseconds. The exponential decay time is represented by the letters f_l with the decay time value in microseconds. These pulses have been generated in repetition mode at a frequency of 10 KHz.

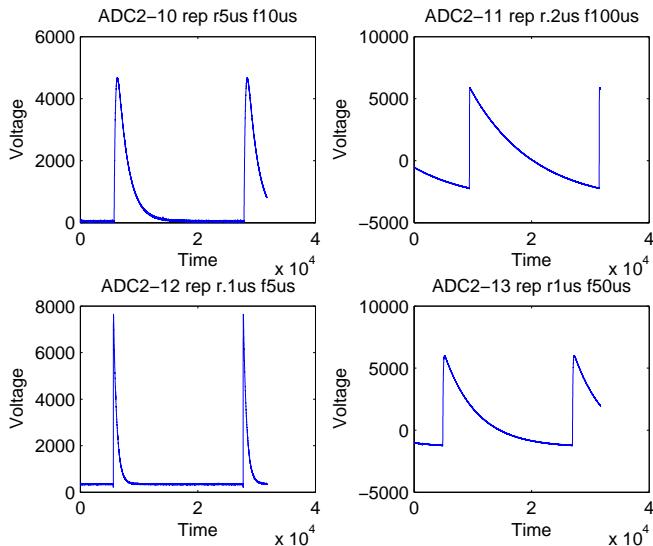


Figure 2.8: Pulses generated with different rise and decay times

It can be observed in figure 2.9 some of the pulses with pile up which occurs often when dealing with pulses at high rates. Those pulses will be used later in section 6.6 of chapter 6 for trying to reduce pile up between two or more pulses. The pulses have been generated in random mode at a frequency of 10 KHz.

Random Generated Pulses

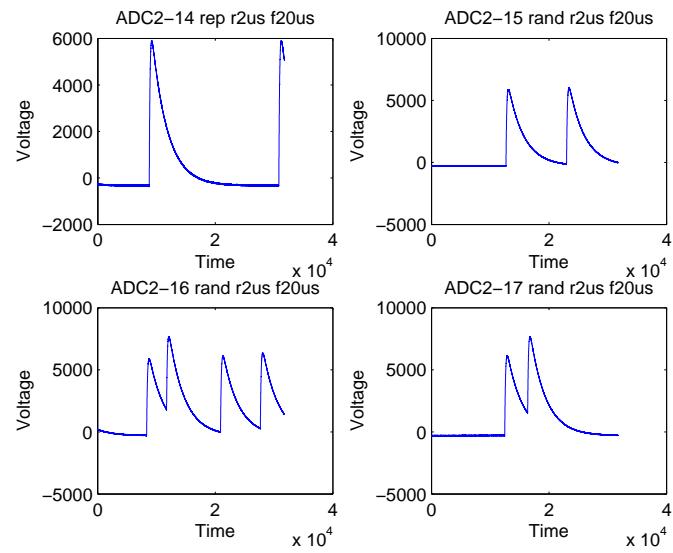


Figure 2.9: **Pulses generated with different rise and decay times and pile up (at the bottom)**

Chapter 3

Energy and Time Adquisition System

3.1. Introduction

The first step in answering the research question was to generate different pulse signals in Matlab and with a pulse generator as showed in chapter 2. The second step is to implement the filters which will extract the parameters of interest of those generated pulse signals.

This chapter describes the general operation of the pulse detection model and gives an overview about the different blocks of the system. There have been implemented two filters to calculate the energy and the time from a pulse:

- $CR - (RC)^n$ Filter

This filter transforms the pulses into bipolar signals whose zero crossing is used to establish the arrival time of the pulse.

The filter has been chosen to be implemented in this thesis based on the research done in books, papers and previous thesis where it has been successfully implemented. The $CR - (RC)^n$ filter is precise in measure the time as it will be shown in chapter 6 and it is able to reject the high frequency noise [24], cancel the low frequency fluctuations and restore the baseline.

The filter was also used in the description of the digital detection system [24] provided by the Physics Department.

- Trapezoidal Filter

This filter is used to calculate the energy from the pulse. The filter transform an exponential signal into a trapezoid with a flat top whose height is proportional to the amplitude of the input pulse. The energy is proportional to the height of the trapezoid.

The trapezoidal filter is used extensively in nuclear physics to extract the energy from the pulse and for this reason it is chosen to be implemented in this thesis. Also the filter was used in the description of the digital detection system [24] provided by the Physics Department.

The filter generates the flat top which gives a larger amount of points to measure the height than a single peak value calculated from a pulse. Pulses contain noise which generates unwanted peaks in the signal. These peak values produced by noise could be used to calculate the energy which may give an error value.

Afterwards the moving average filter is described. This filter [41] is used to smooth out short fluctuations of the signal by averaging a subset of data points.

Finally the baseline restoration process, which is not implemented in this thesis, is explained in section 3.4.

3.2. General Overview of the System

The functionality of the different blocks of the pulse detection system are described in this section. Figure 3.1 illustrates those blocks and how they are connected. The green blocks are the filters implemented in the system and the parameters colored brown situated above some of these blocks can be modified in order to adjust the functionality of those. The pulse shape structure is enclosed inside the red dashed lines and the pulse height structure is enclosed inside the yellow dashed lines.

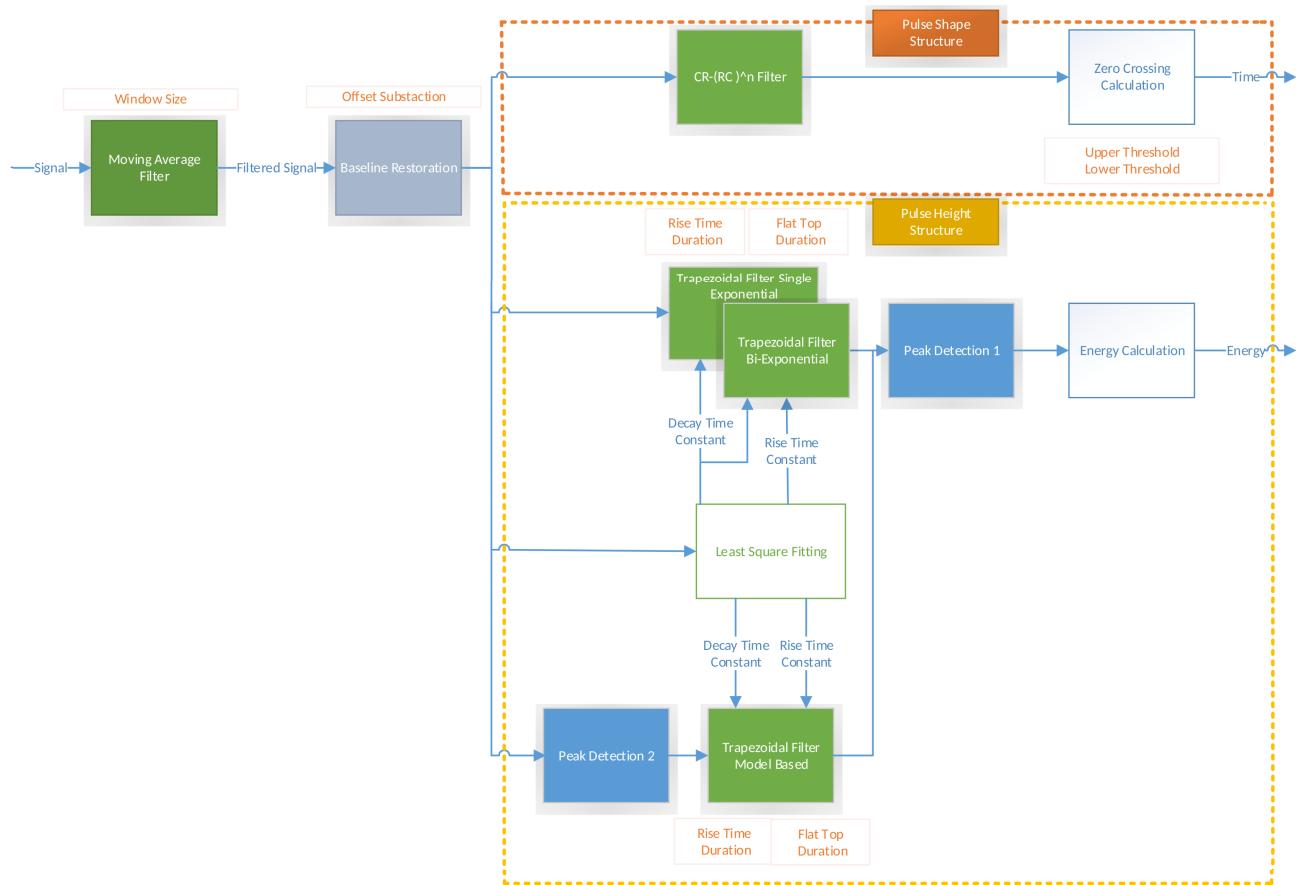


Figure 3.1: Block diagram of the Pulse Detection System

The blocks situated at the left of the figure which are used by both structures will be explained first. Then the pulse shape and the pulse height structures will be explained.

- Common Structure
 - Moving Average Filter: The signals generated with Matlab in chapter 2 are the input of this block and the moving average filter reduce the undesired noise from those signals. In fact they do not contain noise because they are generated with Matlab but the real pulses generated by the radiation detector may contain noise. Because real data have

not been provided by the Physics Department, noise has been added to the generated signals in order to test the filter.

- Baseline Restoration: The signals with reduced noise filtered by the moving average filter are the input of this block. The parameters of the energy and time are calculated based in a zero baseline. The baseline shifts over time so the functionality of the block is to return the baseline to the original zero value.

- Pulse Shape Structure

- $CR - (RC)^n$ Filter: The signals with restored baseline coming from the baseline restoration are the input of this block. The $CR - (RC)^n$ filters convert an uni-polar signal into a bipolar signal to extract the time when this occurs. The voltage of a uni-polar signal is either positive or negative and the voltage of a bipolar signal is both positive and negative.
- Zero Crossing Calculation: The bipolar signals coming from the $CR - (RC)^n$ filter are the input of this block. Zero Crossing is a Matlab function that calculates the time in μs when the signal crosses the zero x axis. This time is used to establish when the pulse arrived.
Peaks originated by noise contained in the signal cause unwanted crossing of the zero x axis. In order to avoid this problem the values of upper and lower threshold can be passed to the function as it can be observed in figure 3.1.

- Pulse Height Structure

- Trapezoidal Filter Single Exponential: The signals with restored baseline coming from the baseline restoration are the input of this block. The following filters are included here: trapezoidal filter recursive and trapezoidal filter TF single. These filters transform a single exponential pulse with decay time constant τ_d into a trapezoid with the same height as the maximum amplitude of the pulse. The height of the trapezoid is proportional to the energy of the pulse so this value is used to calculate the energy. The time the trapezoid takes to reach the top and the duration of the top flat can be modified and adjusted in all the trapezoidal filters implemented.
- Trapezoidal Filter Bi-Exponential: The signals with restored baseline coming from the baseline restoration are the input of this block. The following filter is included here: trapezoidal filter TF biexponential. This filter transform a bi-exponential pulse with rise time constant τ_r and decay time constant τ_d into a trapezoid with the same height as the maximum amplitude of the pulse.
- Least Square Fitting: The signals with restored baseline coming from the baseline restoration are the input of this block. Least Square estimates the rise time constant τ_r and the decay time constant τ_d of the pulse. These parameters are used by the trapezoidal filters to generate the trapezoid output.
- Peak Detection 1: The trapezoidal signals generated by the trapezoidal filters are the input of this block. Peak detection 1 is a Matlab function that calculates the height of the trapezoid.
- Peak Detection 2: The signals with restored baseline coming from the baseline restoration are the input of this block. Peak detection 2 is a Matlab function that calculates the height of the signal.

- Trapezoidal Filter Bi-Exponential model based: The signals coming from the peak detection 2 are the input of this block. The following filter is included here: trapezoidal filter TF biexponential model based. The filter presents the same features like the trapezoidal filter bi-exponential but it has been optimized to minimize the error produced by the filter as much as possible. It uses the height calculated by peak detection 2 to generate a model of bi-exponential equations which will be used as an input signal for the trapezoid filter.
- Energy Calculation: The signals coming from the peak detection 1 are the input of this block. The height calculated by peak detection 1 and the time constants τ_r and τ_d estimated by the least square fitting are used here to calculate the energy of the exponential pulse.

3.3. Moving Average Filter for Noise Reduction

The problem of the noise in the signals is addressed in this section. Measurements taken by the radiation detector are never perfect and may contain errors. Basically there two types of measurement errors:

- Systematic error: where each measurement is less than or greater than the correct value by a certain percentage or amount.
- Random error: where there are variations in the measured signal which can be greater in some parts and smaller in others. This type of error is known as noise or acoustic noise.

There are many sources of noise in physical measurements [38] such as electric power fluctuations, stray radiation from nearby electrical equipment, static electricity and one of the fundamental problems in measuring the height of the signal is how to separate the noise from the signal. The signal generated by the preamplifier which converts the signal generated by the radiation detector in a signal with voltage amplitude contains noise. Noise in signals can obscure important features like peak widths, valleys or maximum peak value. Noise can make more difficult to calculate signal features such as peak widths, rise time or area inside the pulse. The moving average filter is used to reduce the Gaussian white noise from the signal. This filter is chosen because it is easy to implement and it retains a sharp step response. However, the moving average filter is not suitable for frequency domain because it has little ability to separate one band of frequencies from another. The filter averages a number of points from the input signal to produce each point in the output signal. The equation below illustrates this process:

$$y(i) = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \quad (3.1)$$

Where x is the input signal, y the output signal and M the number of points used in the moving average.

The figure below illustrates the bi-exponential pulse with added noise (on the top) and the same signal filtered with moving average filter where the value of M is 10 (on the bottom).

Baseline Restoration

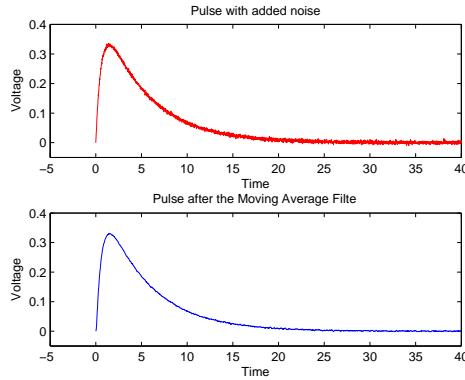


Figure 3.2: **Pulse where added Gaussian noise has been removed with a moving average filter**

There are different techniques which can be used to remove the noise, however smoothing and filtering are two most commonly used signal processing techniques to remove the unwanted noise from a contaminated signal and help obtain important signal features and components.

We describe two filters which can be used instead of the moving average filter

- The standard median filter [37] is a simple nonlinear smoother which is mostly used to suppress noise while retaining sharp sustained changes (edges) in signal values. It is mostly effective in reducing impulsive-type noise. The output generated by the standard median filter at a certain point is the result of the median value of the input data inside the window centered at the point.
- Exponential smoothing [25] is a technique which is mostly used to produce smoothed data to time series data. In the moving average filter the past observations are weighted equally but the exponential smoothing assigns exponentially weights over time in a decreasing order. The number of data points [41] of the filter have to be determinate so that will remove a sufficient amount of noise without affecting significant features of the original signal.

3.4. Baseline Restoration

In nuclear pulse analysis systems [15] the amplitude of the pulse contains the basic information. This amplitude is measured relative to a true zero baseline so shifts on this baseline can influence the calculation of the pulse amplitude and the estimation of the time. Basically baseline is the reference value to calculate the amplitude and it is an important element for calculating the pulse parameters. The baseline will shift over time, and therefore, in order to calculate the most accurate pulse parameters the baseline should be calculated continuously.

In pulse processing the baseline restoration process can be performed by sampling the baseline between pulses digitally and then from this measured pulse amplitude an appropriate value should be subtracted. This process can be performed for each pulse or the same value can be subtracted from all samples during a certain period and then calculate the baseline. The best accuracy will be obtained by estimating the baseline each sample but this would require a relatively long time interval between pulses and most of the times short intervals occur more frequently than long ones. Also frequent measurements of baseline will increase the processing time so there is a trade-off

between the accuracy of the measurements and the processing speed. Digital filters can be used for baseline estimation to optimize the measurement accuracy.

Sometimes it is difficult to find the baseline for the pulse if there is too much noise around the baseline so filters to reduce that noise should be used before baseline estimation like the moving average filter which has been implemented in section 3.3. Baseline calculation is not implemented in this thesis so we assume that it is zero in all processes used to extract the parameters from the filters.

In principle, baseline shifts can be eliminated [15] if the pulse shape is made to be bipolar rather than monopolar as performed with the $CR - (RC)^n$ filters. Bipolar pulses contain both positive and negative lobes as illustrated in the 4.3 (outputs of the filters) and monopolar pulses contain only positive or negative lobes (input signal).

Chapter 4

Pulse Shape Analysis

4.1. Introduction

This chapter describes the implementation of the $CR - (RC)^n$ filters for orders from n=1 to n=4 which are used to extract the arriving time from the pulses.

At first the theory behind the different implementations of the filters is presented and afterwards it is explained how the outputs are calculated based on this theory.

The implementations of the $CR - (RC)^n$ filters have been done by two methods:

1. Filter implementation by transfer functions where the filters are generated by transfer functions in frequency and S-domain.
2. Filter implementation by transfer functions where the filters are generated by transfer functions in Z domain.

Finally the zero crossing calculations used to determine the arriving time of the pulses is discussed in section 4.3.

4.2. $CR - (RC)^n$ Shaper

Filters are devices used to extract useful information and eliminate unwanted information from a signal.

The filters implemented in this chapter are designed to transform the input signal to a desired output signal to obtain information.

Analog filters are made by using components such as capacitors, resistors and or inductors and they filter continuous time varying analog signals. Digital filters operate on sampled, discrete-time signals to change some aspects of that signal and they are implemented by transfer functions.

One of the pulse shaping methods which has been extensively used in the analogue nuclear systems, is the $CR - (RC)^n$ shaping. With the development of the digital electronics, the filter has been implemented digitally to be used in FPGA's and other embedded systems.

This section introduces the transfer functions for real time implementation of $CR - (RC)^n$ filters in digital nuclear systems. These filters are used in our system to extract the timing information from the pulses. Also they provide information about the exact time, with a minimum error, when the pulse reaches its maximum voltage. This information can be used to count the number of pulses in a time interval.

The reason of not using more than four integration stages is because the improvement of the performance of the filter will be very small. $CR - (RC)^n$ where n is greater than 4 may cause a delay in the output and increase the number of components and the complexity of the circuit when implemented in a FPGA.

4.2.1 Digital implementation in Time and Frequency Domain

A $CR - RC$ filter contains a CR circuit followed by an RC circuit. The signal generated by the amplifier first passes through a CR shaper and then through the RC shaper.

By using the transfer functions of the CR and RC filter it is obtained the transfer function of the $CR - RC$ filter in S-domain which is given by:

$$G_{RC-CR}(s) = \frac{1}{1 + \tau_2 s} \frac{\tau_1 s}{1 + \tau_1 s} \quad (4.1)$$

where τ_1 is CR time constant and τ_2 is RC time constant.

The output signal of the step input in time domain becomes:

$$V_{out}(t) = \frac{V_o \tau_1}{\tau_1 - \tau_2} (e^{-t/\tau_1} - e^{-t/\tau_2}) \quad (4.2)$$

From the transfer function in S-domain it can be obtained the transfer function in frequency domain:

$$G_{RC-CR}(i\omega) = \frac{i\omega\tau}{(1 + i\omega\tau)^2} \quad (4.3)$$

where resolving the denominator it is obtained (notice that $i^2 = 1$)

$$G_{RC-CR}(i\omega) = \frac{\omega\tau}{1 + \omega^2\tau^2} \quad (4.4)$$

By the same procedure it can be calculated the transfer function of a $CR - (RC)^n$ filter by multiplying the transfer function of a CR and n times the transfer function of (RC) filters.

$$H(s) = \frac{s\tau}{(1 + s\tau)^{n+1}} \quad (4.5)$$

n is the number of the integrator.

The step response to the filter becomes:

$$V_{out}(t) = \frac{1}{n!} \left(\frac{t}{\tau} \right)^n e^{-t/\tau} \quad (4.6)$$

These filters are called Gaussian filters because the pulse shape is an approximation to a Gaussian pulse.

Figure 4.1 illustrates the responses in time domain of a CR , RC and $CR - RC$ filter where the input is a step function.

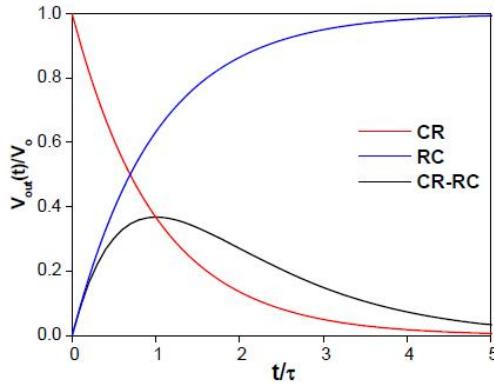


Figure 4.1: Step function responses [6]

It can be observed in the figure that the *CR* filter decreases the decay time of the pulse and the *RC* filter stretches the rising edge of the pulse.

The value of τ determinate how fast the output signal will cross the zero x-axis.

Figure 4.2 shows the frequency responses of the different filters.

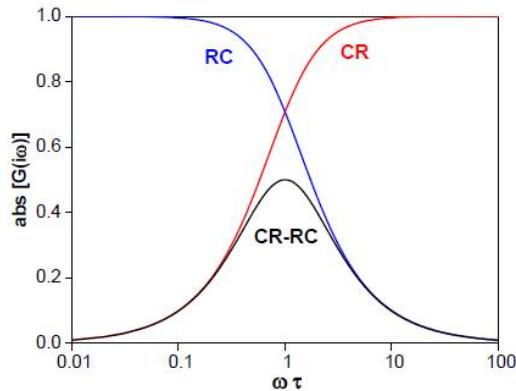


Figure 4.2: Frequency responses [6]

At the frequency of $w = 1/\tau$, the output signal for the *RC* and *CR* filters is about 0.71 level of the input signal. The *CR* filter which performs as a high pass filter attenuates low frequency signals and allows the high frequency signals to pass through the filter which improves the signal-to-noise ratio because the noise is low frequency.

The *RC* filter has the opposite way of performing and it behaves as a low-pass filter. It also improves the signal-to-noise ratio because high frequencies contain noise too. The result of combining the functionality from a *CR* filter and a *RC* filter is a band pass filter which passes the frequencies within a range and reject the frequencies which are outside that range.

CR – (RC)ⁿ filters have infinite impulse responses and this means that their equations are recursive which can be calculated from the Z-transform of the impulse response.

Equation 4.7 is the Z-transform of the impulse response function $H(z)$ where the variable τ is the sampling interval of the digital filter, m is an integer and z is in general a complex number

$$H(z) = \sum_{m=0}^{\infty} h(m\tau)z^{-m} \quad (4.7)$$

4.2.2 Digital Implementation with Transfer Functions

Equations 4.8, 4.9, 4.10 and 4.11 are the Z-transforms [21] of the $CR - (RC)^n$ where n takes values from 1 to 4, $a = 1/\tau$ which represents the decay rate, τ is the shaping time constant and $\alpha = \exp(-T/\tau)$ is an exponential function.

$$H(z)_{CR-RC} = \frac{z^2 - z\alpha(1 + aT)}{(z - \alpha)^2} \quad (4.8)$$

$$H(z)_{CR-(RC)^2} = \frac{z^2 T^2 \alpha (2 - aT) - z T^2 \alpha^2 (2 + aT)}{2(z - \alpha)^3} \quad (4.9)$$

$$H(z)_{CR-(RC)^3} = \frac{z^3 T^2 \alpha (3 - aT) - 4\alpha z^2 T^3 \alpha^2 - z T^2 \alpha^3 (3 + aT)}{2(z - \alpha)^3} \quad (4.10)$$

$$H(z)_{CR-(RC)^4} = \frac{z^4 \alpha T^3 (4 - aT) + z^3 \alpha^2 T^3 (12 - 11aT) + z^2 \alpha^3 T^3 (-12 - 11aT) + z \alpha^4 T^3 (-4 - aT)}{24(z - \alpha)^5} \quad (4.11)$$

The transfer function in z domain $H(z)$ is obtained by dividing the output of the filter in z domain by the input of the filter in z domain:

$$H(z) = Y(z)/X(z)$$

Where it is obtained that

$$Y(z) = H(z)X(z)$$

By calculating the inverse Z transform of $Y(z)$ the recursive algorithms [35] of the filters where the number of computation increases as the order of filter increases are obtained.

These recursive sequences of the $CR - RC$ filter are computed by performing four additions and three multiplications and the $CR - (RC)^4$ sequences perform nine additions and nine multiplications which is more than the double. This aspect [15] has to be taken in consideration when performing operations in very high rates where the interval between pulses are greater than the width of the pulse signals and where results should be presented in real time.

4.2.3 Filter Test with Generated Pulses

This section introduces the implementation in Matlab of the $CR - (RC)^n$ filters and the theory explained in 4.2 is now applied here.

Figure 4.3 shows on the top left the input signal colored in red which contains two bi-exponential pulses. These pulses correspond to the signal $adc1_{10}$ from appendix E which has a rise time of 5 μs and a decay time of 10 μs . The outputs of the $CR - (RC)^n$ filter are colored in blue in the figure. P1 and P3 correspond to the upper threshold crossing points when the output signal rises and P2 and P4 are the lower threshold crossing points when the output signal decreases.

Zero Crossing Calculation

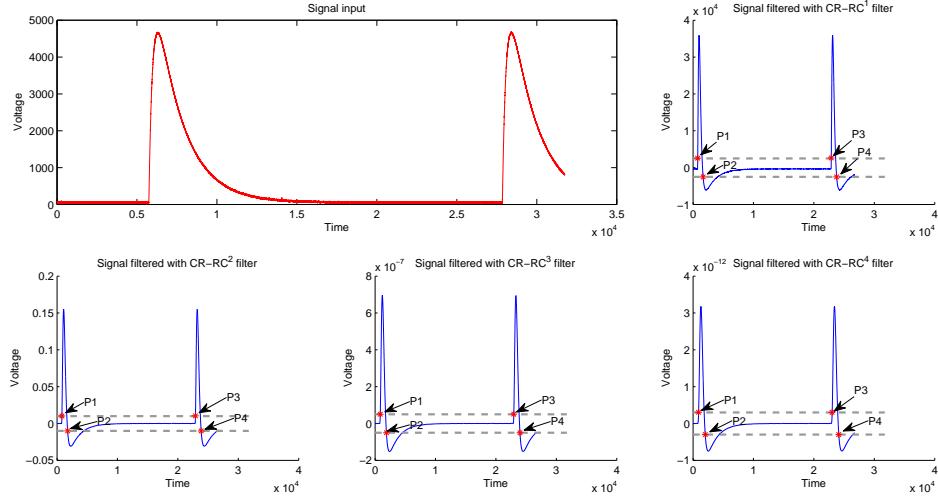


Figure 4.3: $CR - (RC)^n$ shaper filter with degrees 1,2,3 and 4

The filters have been implemented by using the following transfer functions:

- $CR - (RC)^1$ implemented with transfer function 4.8 situated at the top right where the thresholds are set to $-0.25e^4$ and $0.25e^4$.
- $CR - (RC)^2$ implemented with transfer function 4.9 situated at the bottom left where the thresholds are set to $-0.1e^{-1}$ and $0.1e^{-1}$.
- $CR - (RC)^3$ implemented with transfer function 4.10 situated at the bottom in the middle where the thresholds are set to $-0.5e^{-7}$ and $0.5e^{-7}$.
- $CR - (RC)^4$ implemented with transfer function 4.11 situated at the bottom right where the thresholds are set to $-3e^{-13}$ and $0.10e^{-13}$.

It is observed in figure 4.3 that all the outputs have the same shape but when the order of the filter increases the height of the peaks decreases and therefore the threshold values become smaller. This happens because the amplitudes of the output pulses are independent from the amplitude of the input pulse. The time when the signal crosses the zero x axis is also independent from the amplitude of the input pulse and it depends on the time constant τ_d of the input signal.

$CR - (RC)^n$ also reduces considerably the pulse width and the top length making the pulse shaper as it can be seen in the figure so this method cannot be used to estimate the energy of the pulse. One of the advantages of implementing the digital filter is that even this method reduces significantly the amplitude of the signal it does not introduce noise in the system as with the analog implementation where the differentiation part of the filter allows all high frequency components to pass through the network causing the signal to noise ratio to become very poor.

4.3. Zero Crossing Calculation

The zero crossing function implemented in Matlab will be introduced in this section. The bipolar pulse is obtained after the bi-exponential signal has passed though the $CR - (RC)^n$ filter as explained in section 4.2. A zero crossing function has been implemented in order to calculate the

time when the particle arrives to the radiation detector.

The function runs through all samples of the input signal and checks which sample (y positions) is equal to zero (or nearest to zero) and returns a vector with the times (x positions) when the signal crossed the x axis. The input signal $adc1_{10}$ of figure 4.3 used to test the $CR - (RC)^n$ function has 31744 samples so the function perform this number of interactions.

When there is noise in the signal then this crosses several time the x axis as it can be observed in the figure 4.4. A threshold should be set in order to avoid this problem.

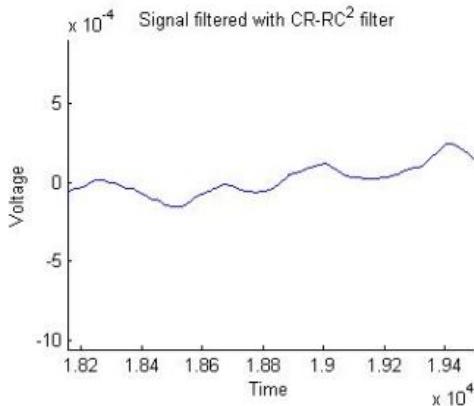


Figure 4.4: **Ampliation of $CR - (RC)^2$ filter**

It is possible to pass a positive and a negative threshold value to the zero crossing function so the function will return two vectors. The first vector contains the x positions when the signal crossed the first threshold (positive) and the second vector contains the x positions when the signal crossed the second threshold (negative).

Chapter 5

Pulse Height Analysis

5.1. Introduction

This chapter describes the implementation of the trapezoidal filter which is used to extract the energy from the pulses.

First the theory behind the different implementations of the trapezoidal filter is presented and afterwards it is explained how the outputs are calculated based on this theory.

The implementations of the trapezoidal filters have been done by two methods:

1. Filter implementation by recursive algorithms where the filters are generated by recursive formulas in discrete time domain.
2. Filter implementation by transfer functions where the filters are generated by transfer functions in Z domain.

The decay time constant τ_d value and the rise time constant τ_r value are used by the trapezoidal filter to generate the trapezoid output. These constants are calculated from the generated pulse by the least square fitting function which is explained in section 5.3. After it is shown how the peak function to detect the peaks is implemented and how this function behaves when noise is added to the signal.

Finally in the last part it is shown how the energy can be calculated from the pulses.

5.2. Trapezoidal Filter

Trapezoidal shaping method is mostly applied to extract the amplitude and the energy of the pulse in digital nuclear spectrometer system. The shaping parameters have to be selected adequately to improve the energy resolution as it is seen in 5.5 and to avoid pile up as it is explained in 6.6.

Digital processing of nuclear signals [39] has been used for 40 years to process the nuclear signals. At that time, digital filters were designed to change the nuclear signal into trapezoidal pulse, and a set of X-ray energy spectrum measuring system was developed based on this digital filter.

The trapezoidal filter [39] has some advantages like pulse throughput and energy resolution and it is mostly used to process the digital nuclear signal worldwide. For example, the companies of Canberra, XIA and CAEN [23] have used the trapezoidal shaping filter into the series of products

of the digital nuclear spectrometer.

In this section the implementation of a trapezoidal filter is performed. The trapezoidal filter converts the pulse with short rise time and exponential decay generated by the preamplifier in a pulse with trapezoidal shape. The pulse high of the trapezoidal output is proportional to the energy of the pulse generated by the radiation detector.

5.2.1 Digital Implementation with Recursive Algorithms

The procedure to obtain the recursive equations which will generate the trapezoidal output is described in this section.

The desired trapezoidal output $y(t)$ of a linear time-invariant system is given by the convolution of the input signal $x(t)$ and the impulse response of the system $h(t)$.

$$y(t) = \int_{-\infty}^{+\infty} x(t') h(t - t') dt' \quad (5.1)$$

The goal is to find the impulse response $h(t)$ that converts the exponential pulse in a pulse with flat shape.

This is accomplished by:

- The convolution of the input sampled signal $x(i)$ with the rectangular function is explained in 5.2.1.1
- The convolution of the input sampled signal $x(i)$ with truncated ramp function which is explained in 5.2.1.2

5.2.1.1 Convolution with Rectangular Function

Here it is considered the case where the impulse response is a rectangular function.

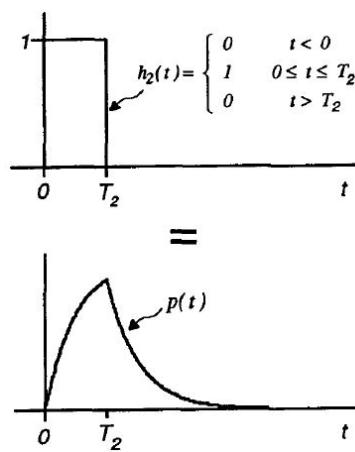


Figure 5.1: Convolution of single exponential function with rectangular function [16]

The output response of the system [16] is illustrated at the bottom of the figure 5.1 which is the convolution of the rectangular function (at the top of the figure) with a single decay exponential

Convolution with Ramp Function

function.

The result of the convolution is described by the equations:

$$p(t) = \int_0^t e^{(t'-t)/\tau_d} dt' = \tau_d(1 - e^{-\frac{t}{\tau_d}}) \quad (5.2)$$

for $0 \leq t \leq \tau_2$

$$p(t) = \int_0^{\tau_2} e^{(t'-t)/\tau_d} dt' = \tau_d e^{-\frac{t}{\tau_d}} (e^{-\frac{\tau_2}{\tau_d}} - 1) \quad (5.3)$$

for $0 \leq t \leq \tau_2$

The initial conditions are considered to be zero when the time $t < 0$. The decay time constant τ_d of the output signal $p(t)$ is the same that the time constant of the input signal $x(t)$.

5.2.1.2 Convolution with Ramp Function

Here it is considered the case where the impulse response is a truncated ramp function as illustrated in the figure 5.2.

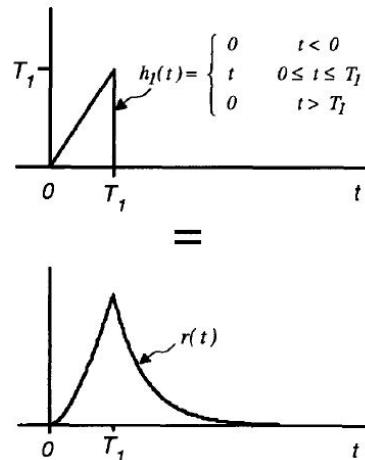


Figure 5.2: Convolution of single exponential function with ramp function [16]

The output response of the system [16] is illustrated at the bottom of the figure 5.2 which is the convolution of the ramp function (at the top of the figure) with a single decay exponential function. Also in this case initial conditions are considered to be zero for $t < 0$. The result of the convolution with the truncated ramp function is described by the equations:

$$r(t) = \int_0^t e^{(t'-t)/\tau_d} dt' = \tau_d t - \tau_d^2 (1 - e^{-\frac{t}{\tau_d}}) \quad (5.4)$$

for $0 \leq t \leq \tau_1$

$$r(t) = \int_0^t e^{(t'-t)/\tau_d} dt' = \tau_d e^{(t'-t)/\tau_d} (\tau_d + e^{\frac{T_1}{\tau_d}}) \quad (5.5)$$

for $t > \tau_1$

5.2.1.3 Digital Design of the Impulse Response of the Trapezoidal Filter

The signal is sampled at equal intervals of time in the digital implementation. The first task is to convolve the input signal $x(i)$ with the rectangular function. The following recursive convolution algorithm is used because this operation has to be performed in real time.

$$p(n) = \sum_{l=0}^n x(i) - x(i-l) \quad (5.6)$$

or

$$p(n) = p(n-1) + x(n) - x(n-l), n \geq 0 \quad (5.7)$$

Where $x(n)$ is sampled at time n and $x(n-l)$ is delayed by n relative to l . The initial conditions are $x(n) = 0$ for $n \geq 0$.

The next step is to convolve the input sampled signal $x(i)$ with the truncated ramp function. The recursive equation of the convolution is:

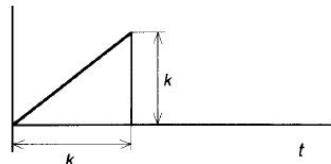
$$r(n) \sum_{l=0}^n \left\{ \sum_{j=0}^l x(j) - x(j-k) - x(i-k')k' \right\} \quad (5.8)$$

or

$$r(n) = r(n-1) + p(n) - x(n-k')k', n \geq 0 \quad (5.9)$$

Where $p(n)$ is the recursive convolution algorithm with the rectangular function with length k and k' is a delay parameter and it is equal to the value of k .

The figure below shows the impulse response of a truncated ramp system when $k = k'$ where x axis represents the rising time and the y axis the amplitude of the trapezoid.



The response of the system can be written in discrete time domain in form of recursive algorithm from equation:

$$y(n) = r(n) + Mp(n) + (k-M)p(n-k) - r(n-l) \quad (5.10)$$

Where M is equal to the decay time constant τ_d of the exponential input signal, k is the rise time of the trapezoidal shape and $m = l - k$ is the duration of the flat shape.

From equations 5.7 and 5.9 the response of the system can be written in discrete domain with the equations:

$$y(n) = y(n-1) + p'(n) - d^{k,l}(n)M, n \geq 0 \quad (5.11)$$

$$d^{k,l}(j) = x(j) - x(j-k) - x(j-l) + x(j-k-l) \quad (5.12)$$

$$p'(n) = p'(n - 1) + d^{k,l}(n), n \geq 0 \quad (5.13)$$

Equations 5.11 and 5.13 represent the recursive algorithms to obtain a symmetric trapezoidal shape from an exponential decay function.

5.2.2 Digital Implementation with Transfer Functions

This section describes three different implementations of the trapezoidal filter in z-domain.

- Subsection 5.2.2.1 describes the process of obtaining a transfer function when the input signal is a single exponential pulse.
- Subsection 5.2.2.2 describes the process of obtaining a transfer function when the input signal is a bi-exponential pulse.
- Subsection 5.2.2.3 describes the process of generating a trapezoid output when the input signal is a bi-exponential pulse.

The filter has the feature that it transforms a exponential signal with decay time constant τ_d and rise time constant τ_r (in the case of bi-exponential pulse) into a trapezoidal signal with rising time t_a , decaying time $t_c - t_b$ and flat-top $t_b - t_a$. Figure 5.3 shows the trapezoid output where it is observed that the x axis represent the time t and the y axis the amplitude A which is proportional to the voltage of the pulse represented by V_u .

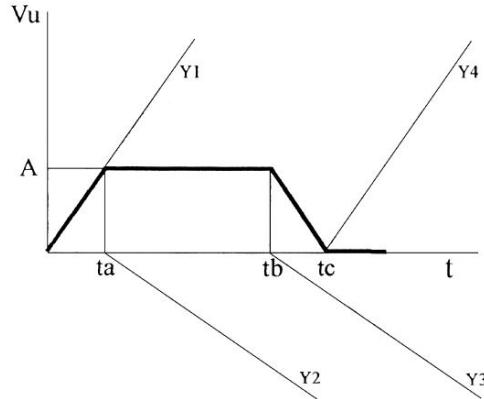


Figure 5.3: Trapezoidal pulse to digitally synthesize [11]

The output of the filter $y(t)$ can be represented as a sum of the four straight lines Y_1, Y_2, Y_3 and Y_4 as shown in figure 5.3. Each line is zero on the left side of its intersection with t axis so the unit Heaviside function $U(t)$ is used.

$$y(t) = t \frac{A}{t_a} U(t) - \frac{A}{t_a} (t - t_a) U_{t-t_a} - \frac{A}{t_a} (t - t_b) U_{t-t_b} + \frac{A}{t_a} (t - t_b - t_a) U_{t-t_b-t_a} \quad (5.14)$$

5.2.2.1 Transfer function implementation with single exponential pulse

The transfer function of the trapezoidal filter is calculated by dividing the output of the trapezoid by the single exponential input signal.

The single decay input signal $x(t)$ from equation 2.10 is defined by:

$$x(t) = A e^{(\frac{-t}{\tau_d})} \quad (5.15)$$

Taking the z-transform of the input $x(t)$ it is obtained:

$$X(z) = A \frac{z}{z - e^{(\frac{-Ts}{\tau_d})}} \quad (5.16)$$

where Ts represents the sample period. Ts is calculated in equation 6.1.

The z-transform of the output series $y(t)$ is:

$$Y(z) = \frac{A}{n_a} \frac{z}{(z-1)^2} - \frac{A}{n_a} z^{-n_a} \frac{z}{(z-1)^2} - \frac{A}{n_a} z^{-n_b} \frac{z}{(z-1)^2} - \frac{A}{n_a} z^{-n_a-n_b} \frac{z}{(z-1)^2} \quad (5.17)$$

where $n_a = \frac{t_a}{Ts}$, $n_b = \frac{t_b}{Ts}$ and $n_c = \frac{t_c}{Ts}$.

$Y(Z)$ can be reduced to obtain:

$$Y(z) = \frac{A}{n_a} \frac{z}{(z-1)^2} (1 - z^{n_a} - z^n + z^{(n_a-n_b)}) \quad (5.18)$$

Using those equations the transfer function can be written:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{A}{n_a} \frac{z}{(z-1)^2} (1 - z^{n_a} - z^{n_b} + z^{(n_a-n_b)})}{A \frac{z}{z - e^{(\frac{-Ts}{\tau_d})}}} \quad (5.19)$$

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-Ts}{\tau_d})})(1 - z^{n_a} - z^{n_b} + z^{(n_a-n_b)})}{(z-1)^2} \quad (5.20)$$

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-Ts}{\tau_d})})(1 - z^{-n_a})(1 - z^{-n_b})}{(z-1)^2} \quad (5.21)$$

Multiplying numerator and denominator by z^{-1} then the final equation becomes:

$$H(z) = \frac{z^{-1}}{n_a} \frac{(1 - e^{(\frac{-Ts}{\tau_d})} z^{-1})(1 - z^{-n_a})(1 - z^{-n_b})}{(1 - z^{-1})^2} \quad (5.22)$$

Figure 5.4 shows the input signal $x(t)$ colored in red with a very short rise time close to zero, a decay constant $\tau_d = 90 \mu s$ and an amplitude $A=6$. The figure situated at the bottom left shows the output of the trapezoidal filter implemented with the calculated recursive equations. The duration of the flat top of the trapezoidal shape can be adjusted by changing the parameter m and the rise time of the trapezoidal shape can be modified by changing the parameter k in equation 5.10. The trapezoidal output implemented with the calculated transfer function is situated at the bottom right. The duration of the flat top and the rise time can be adjusted by the parameters na and nb in equation 5.17.

Transfer function implementation with bi-exponential pulse

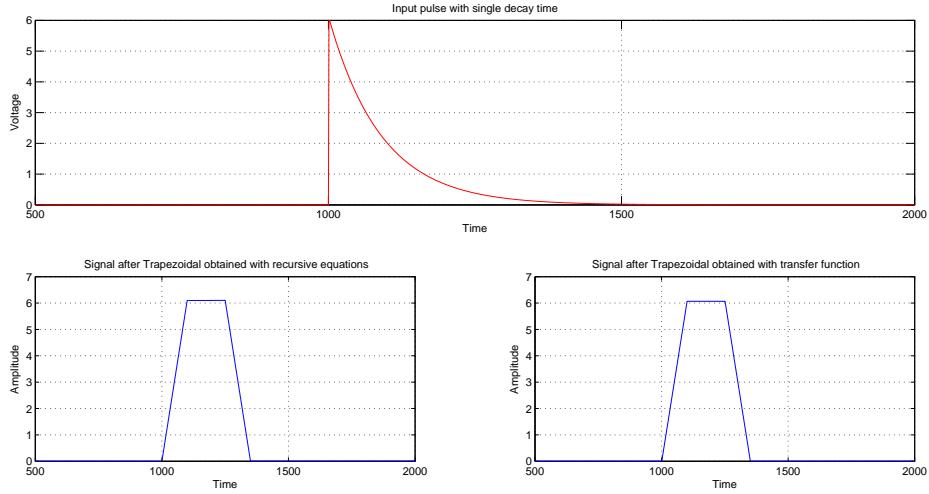


Figure 5.4: Trapezoidal filter output with recursive algorithms and TF implementations

Both implementations produce a trapezoidal shape which is proportional to the height of the input signal. In the next chapter it is analyzed the accuracy of the different implementations of those filters.

5.2.2.2 Transfer function implementation with bi-exponential pulse

The transfer function of the trapezoidal filter is here calculated when the input is a bi-exponential signal.

The input series of the bi-exponential pulse $x(t)$ is defined by the equation:

$$x(t) = A(e^{(\frac{-t}{\tau_d})} - e^{(\frac{-t}{\tau_r})}) \quad (5.23)$$

where τ_d is the decaying time constant and τ_r is the rise time constant as illustrated in equation 2.8.

After taking z-transform of the input $x(t)$ we obtain

$$X(z) = A \frac{z(e^{(\frac{-T_s}{\tau_d})}) - (e^{(\frac{-T_s}{\tau_d})} e^{(\frac{-T_s}{\tau_r})})}{(z - e^{(\frac{-T_s}{\tau_d})})(z - e^{(\frac{-T_s}{\tau_r})})} \quad (5.24)$$

The transfer function becomes using equation 5.17 and 5.24

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{A}{n_a} \frac{z}{(z-1)^2 (z^{n_a+n_b})} (1 - z^{n_a} - z^{n_b} + z^{(n_a+n_b)})}{A \frac{z(e^{(\frac{-T_s}{\tau_d})}) - (e^{(\frac{-T_s}{\tau_d})} e^{(\frac{-T_s}{\tau_r})})}{(z - e^{(\frac{-T_s}{\tau_d})})(z - e^{(\frac{-T_s}{\tau_r})})}} \quad (5.25)$$

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-T_s}{\tau_d})})(z - e^{(\frac{-T_s}{\tau_r})})(1 - z^{n_a} - z^{n_b} + z^{(n_a-n_b)})}{(z - 1)^2 (z^{(n_a+n_b)}) (e^{(\frac{-T_s}{\tau_d})} - (e^{(\frac{-T_s}{\tau_d})} e^{(\frac{-T_s}{\tau_r})}))} \quad (5.26)$$

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-T_s}{\tau_d})})(z - e^{(\frac{-T_s}{\tau_r})})(1 - z^{n_a} - z^{n_b} + z^{(n_a-n_b)})}{(z^{(n_a+n_b)}) (e^{(\frac{-T_s}{\tau_d})} - e^{(\frac{-T_s}{\tau_r})})} \quad (5.27)$$

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-T_s}{\tau_d})})(z - e^{(\frac{-T_s}{\tau_r})}) z^{n_a+n_b} (1 - z^{n_a} - z^{n_b} + z^{(n_a-n_b)})}{(z^{(n_a+n_b)})(e^{(\frac{-T_s}{\tau_d})} - e^{(\frac{-T_s}{\tau_r})})} \quad (5.28)$$

This can be factorized in terms of z

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-T_s}{\tau_d})})(z - e^{(\frac{-T_s}{\tau_r})})(z^{n_a+n_b})(1 - z^{n_a})(1 - z^{n_b})}{(z^{(n_a+n_b)})(e^{(\frac{-T_s}{\tau_d})} - e^{(\frac{-T_s}{\tau_r})})} \quad (5.29)$$

Finally if we divide numerator and denominator by $z^{(n_a+n_b)} * z^2$

$$H(z) = \frac{z^{-1}}{(z^{-2})n_a} \frac{(1 - e^{(\frac{-T_s}{\tau_d})} z^{-1})(1 - e^{(\frac{-T_s}{\tau_r})} z^{-1})(1 - z^{n_a})(1 - z^{n_b})}{(e^{(\frac{-T_s}{\tau_d})} - e^{(\frac{-T_s}{\tau_r})})} \quad (5.30)$$

Figure 5.5 shows on the top the generated bi-exponential input signal $x(t)$ colored in red with rise time constant $\tau_r = 5 \mu s$, decay time constant $\tau_d = 50 \mu s$ and $A = 1$ and at the bottom the trapezoidal output colored blue.

It is possible to observe at the figure that the height of the bi-exponential pulse is not equal to height of the trapezoid. This is determined by the parameter A as one can observe in the figure 5.3. Here A is not the maximum amplitude of the input signal $x(t)$. It is multiplied by $(e^{(\frac{-t}{\tau_d})} - e^{(\frac{-t}{\tau_r})})$ as shown in equation 2.8 which decreases the value of A .

This does not happen with the trapezoidal filter implemented with the single decay input signal because in this case the parameter A is the maximum value of the input signal $x(t)$. A is multiplied by the decaying function $e^{(\frac{-t}{\tau_d})}$ as illustrated in 2.10 where $A * e^{(\frac{-t}{\tau_d})}$.

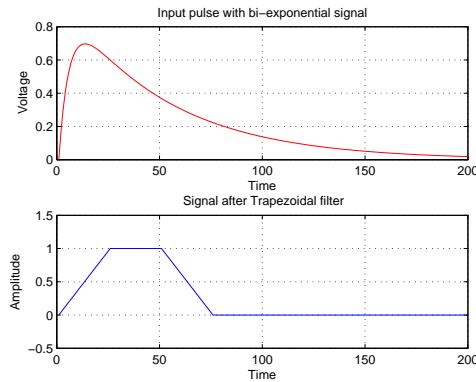


Figure 5.5: Trapezoidal filter output with TF implementation

5.2.2.3 Transfer function model based implementation with bi-exponential pulse

The process of generating a trapezoid output when the input signal is a bi-exponential pulse is described in the following steps.

This process is called model based because it uses a *model of bi-exponential equations* as an input for the trapezoidal filter instead of the input exponential signal $x(t)$ generated by the pulse generator.

The process contains the following steps:

Filter Test with Generated Pulses

1. Find the peak (or amplitude) value A of the bi-exponential input pulse $x(t)$ generated with the signal generator with the peak function. This function will be explained in 5.4 and basically it calculates the peak from a pulse.
2. Find the rise time constant τ_r and the decay time constant τ_d values of the bi-exponential input pulse $x(t)$ generated with the pulse generator with the least square fitting function. This function will be explained in 5.3 and it is used to calculate the values of τ_r and τ_d from a pulse.
3. The model equations $x_{model}(t)$ will be used as input for the trapezoidal filter instead of the signal $x(t)$. This *model of bi-exponential equations* is defined by $x_{model}(t) = A * (e^{-t/\tau_d} - e^{-t/\tau_r})$ which is the equation 5.30 explained in section 2.3.2.
4. The trapezoidal filter implemented with the transfer function 5.30 will generate a trapezoid output with a height value equal to the value of A present at the *model of bi-exponential equations* $x_{model}(t) = A * (e^{-t/\tau_d} - e^{-t/\tau_r})$ as it can be observed in figure 5.3. By this reason the value of A calculated in step 1 and the values of τ_r and τ_d calculated in step 2 from the input signal $x(t)$ generated with the pulse generator will be used in the *model of bi-exponential equations* of step 3.

These steps are done by the following reason: When the value of the rise time constant τ_r increases then the value of the parameter A decreases and the filter will reduce its accuracy.

The input signal from figure 5.4 is used to test the Trapezoidal filter (TF biexponential model based).

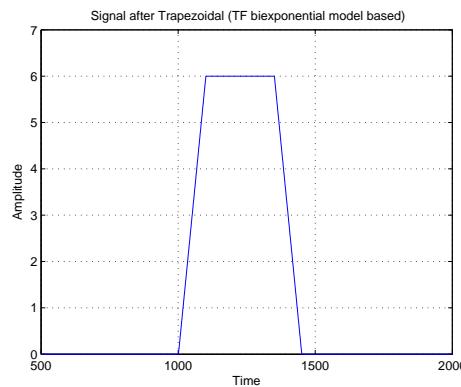


Figure 5.6: Signal after Trapezoidal filter (TF biexponential model based)

It can be observed that the height of the trapezoid output is 6. This equal to the value of the amplitude of the input pulse.

5.2.3 Filter Test with Generated Pulses

In this section the implementation in Matlab of the trapezoidal filter is shown and the theory explained in 5.2 is used.

Figure 5.7 shows the input signal $adc1_{10}$ from appendix E generated by the pulse generator used to test the four different implementations of the trapezoidal filter. This bi-exponential signal has been chosen because its rise time is $5 \mu s$ which is the largest provided by the University and the decay

Chapter 5. Pulse Height Analysis

time is $10 \mu s$ which is the second shortest. This is the worst possible provided scenario to perform the tests of the filters. Generally when the rise time increases and the decay time decreases then the accuracy of the filter becomes worse as it is seen in chapter 6.

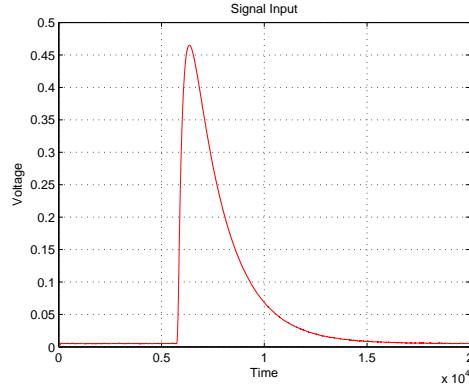


Figure 5.7: **Bi-exponential Input signal $adc1_{10}$**

In the figure 5.8 it can be observed the four trapezoid outputs for the input signal of figure 5.7 corresponding to the following filters:

- Trapezoidal filter recursive implemented with equations 5.11 and 5.13 situated at the top left.
- Trapezoidal filter TF Single implemented with transfer function 5.22 situated at the top right.
- Trapezoidal filter TF Bi-exponential implemented with transfer 5.30 situated at the bottom left.
- Trapezoidal filter TF Bi-exponential model based implemented with transfer function 5.30 situated at the bottom right.

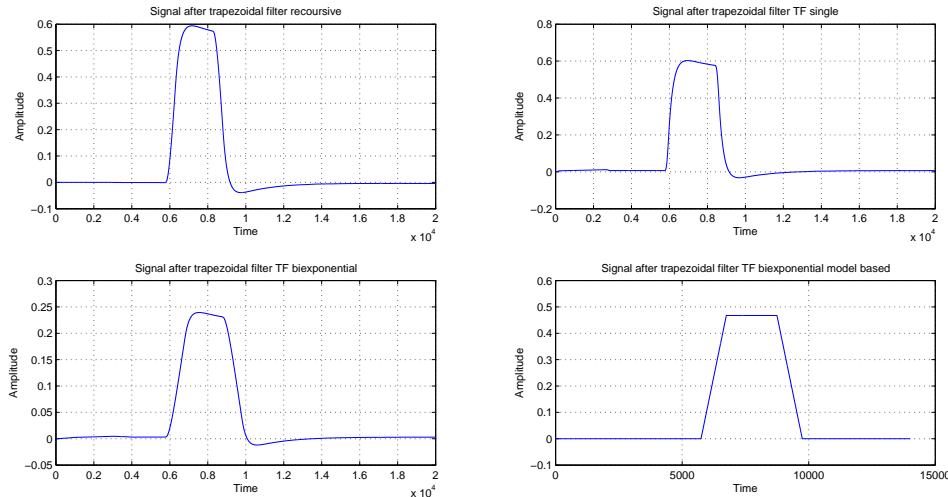


Figure 5.8: **Filter outputs with the different implementations of the trapezoidal filter**

It can be seen in the picture that all the outputs, except the one implemented with TF Bi-exponential model based, don't have the form of perfect trapezoids as it happened with the theoretical implementation of the filter. There are basically two reasons which explain this fact:

- The trapezoidal filters implemented with the recursive equations and with the single decay transfer function are designed to deal with signals where the rise time is very short with values close to zero. When the rise time increases then the corners of the outputs start becoming rounded. Also the top of the trapezoids are becoming less flat making more difficult to calculate the height of these.
- The value of τ_d has to be calculated from the input signal $x(t)$, and furthermore, also the value of τ_r has to be calculated when the input signal is bi-exponential. Those values extracted by nonlinear data-fitting (will be explained in 5.3) are approximations because not all the signals fit perfectly in the model of bi-exponential equations defined by $x(t) = A * (e^{-t/\tau_d} - e^{-t/\tau_r})$ (equation 5.30).

It can be observed in the figure that the height of the trapezoid output implemented with TF Bi-exponential model based is 0.46. This equal to the value of the amplitude of the input pulse $x(t)$ from figure 5.7. Also the output of the filter is a perfect trapezoid with flat top which has not been modified by rise time growth.

5.3. Least Square Fitting

One of the most difficult topics in all of data analysis is fitting data to nonlinear models. When defining the parameters of the trapezoidal filter an important issue is to extract the decay time constant τ_d when dealing with single decay pulses and furthermore the rise time constant τ_r if dealing with bi-exponential pulses.

This is important because the energy of the pulse is proportional to the amplitude of the trapezoid output and to the value of the τ_d as illustrated in section 5.5. This means that if the values of τ_r and τ_d are not precise then this will influence:

- The amplitude of the trapezoidal filter because the equations or the transfer functions which generate the trapezoid out contain the parameters τ_r and τ_d .
- The value of the energy measured in the pulse because E is proportional to the parameters τ_d and τ_r .

In particle physics applications the time constants τ_r and τ_d depend on the type of particle detected by the radiator detector. For example alpha particles have different time constants values than protons .

Curve fitting is used in this thesis and the idea is to find a mathematical model that fits the data. The curve fit finds the specific coefficients (time constants) which make that function match the data as closely as possible

Two ways to estimate those time constants are proposed:

1. Solution approach [20] using least square equations.
2. Solution approach [20] finding the minimum of the multivariable function.

Least Square Equations:

The method of least squares [41] is an approach used to find the approximate solution of sets of

equations when there are more equations than unknowns. This method minimizes the sum of the squares of the errors made in the results of each equation.

The method is mostly used in data fitting because it minimizes the sum of the squared residuals. A residual can be defined as the difference between the value provided by a model and the observed value.

Polynomial least squares [41] describes the variance in a prediction of the dependent variable as a function of the independent variable and the deviations from the fitted curve. When the observations come from exponential functions [18] then least-squares estimates and maximum-likelihood estimates are identical. The following equation [20] is used to find the values of τ

$$\min x = \|F(x, xdata) - ydata\|_2^2 = \min x \sum_i (F(x, xdata_i) - ydata_i)^2 \quad (5.31)$$

where $xdata$ is the input data, $ydata$ is the observed output data, $F(x, xdata)$ is the function and x is a vector or a matrix containing the obtained result.

Finding the Minimum:

The method of finding the minimum [41] is based in the Levenberg-Marquardt. Basically this algorithm search for the minimum value of the following equation.

$$\sum_i (ydata - ydata_i)^2 \quad (5.32)$$

where $ydata$ is a fitted value (model value) for a given point and $ydata_i$ is the measured data value for the point.

Equation 5.32 defines a surface in a multidimensional error space. The process of finding the minimum consists on starting by an initial guess at the coefficient values and then search for the minimum value by calculating the result values from equation 5.32 surface starting from that point.

The goal is to find the deepest valley [40] in that surface. At that point, the coefficient values of the fitting function minimize which means that the difference between the experimental data and fit data (the model) is minimum. There are some functions which have only one valley so the best fit is found when the bottom of the valley has been found. There are also some functions with multiple valleys and it can happen that the fit found is not the best because there are valleys with better fits. When the bottom of this valley is found then the fit is complete even though there can be a valley which is deeper elsewhere on the surface. Initial guesses determine which valley is found first.

Figure 5.9 shows how the nonlinear data-fitting is applied on finding the values of τ_r and τ_d of the bi-exponential equation (2.8) which best fits the input pulse signal colored in red. The input pulse used is the $adc1_{10}$ signal from appendix E which has a rise time of $5 \mu s$, a fall time of $10 \mu s$ and a frequency of $10 KHz$. The signal should start at $t = 0$ in order for Matlab to find the best approximation values so samples from 5732 to 14000 from the $adc1_{10}$ signal have been used.

The signal in blue dashed lines represents the best approximation of the equation found by Matlab.

Peak detection

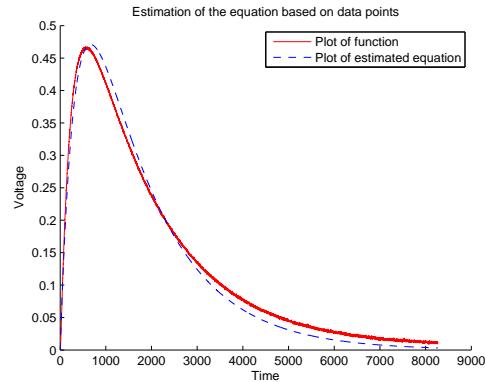


Figure 5.9: Estimated bi-exponential pulse

It can be seen that the signal does not fit perfectly in the model of bi-exponential equations defined by $x(t) = A * (e^{-t/\tau_d} - e^{-t/\tau_r})$ (equation 5.30). This means that the values of the time constants τ_r and τ_d calculated by least square fitting will be the best approximations.

5.4. Peak detection

It will be introduced in this section the process to detect the peaks when there is one or more pulses. In particle physics applications peak detection can be used to detect pulses which are detected with a very short time between them causing pile up. These pulses can be recognized by the trapezoidal filter as a single pulse and peak detection can be used as a pulse rejection mechanism in those cases.

Peak detection [25] is the process of finding the locations and the amplitudes of local maxima and local minima in a signal that satisfies certain properties, like for example requiring that a peak exceeds a certain threshold value. The function implemented in Matlab detects peaks in a signal and measure their positions and heights. It is possible to define a threshold value to determine if each peak is larger or smaller than that value.

Peak detection is used in this project to measure the accuracy of the different implementations of the filters in the following ways:

- Trapezoid Filter: The height of the input signal $x(t)$ returned by the peak detection function is compared against the height of the trapezoid.
- $CR - (RC)^n$ filters: The positions of the input signal $x(t)$ returned by the peak detection function are compared against the positions returned by the zero cross function (explained in section 4.3) when the output of the $CR - (RC)^n$ filter crosses the thresholds.

A Priori conditions about the Input Signal

To use the peak detection function correctly some considerations about the input signal have to be taken into account.

The issues to consider are:

- The signal must be in time-domain and not in frequency-domain.
- The noise must be removed from the signal.

This is a common occurring problem in peak detection. When the signal has noise the results in the detection are a large number of peaks and only a few of these will be of interest. It is

preferable to smooth or process the data before applying the peak detection function. There are filtering techniques to effectively reduce noise of the data as performed with the moving average filter. The peak detection function then processes only the signal that is passed to it without noise.

The peak function has been tested with the signal of figure 6.2 where Gaussian noise has been added. The *peakMag* vector contains the heights of the four pulses where the signal has been tested without noise. The *peakMagwithnoise* vector contains the heights values of the noisy signal.

The results of the Peak detection function are:

- *peakMag* detected peaks at positions 16, 6, 8 and 12
- *peakMagwithnoise* detected 311 peaks

Increasing the threshold level will reduce the number of peaks values detected however the possibility of detecting more peaks than four is still high.

5.5. Energy Calculation

The goal of extracting the height of the pulse with the trapezoidal filter is to calculate the energy of the pulse. In this section the process of calculating this energy is introduced.

There is a relation between the power and the energy [26] of the exponential pulse signals used for testing the filters.

Considering complex values then the total normalized energy in the interval $t_1 \leq t \leq t_2$ is given by:

$$\int_{t_1}^{t_2} |x(t)|^2 dt \quad (5.33)$$

where $x(t)$ is the continuous input exponential pulse signal. The average normalized power is obtained by dividing the equation 5.33 by the duration of the interval $t_1 \leq t \leq t_2$. However if obtaining the energy of the exponential signals in an infinite period of time then there is no need to deal with time periods. In this case the total normalized energy is the same like equation 5.33 but with an infinite time interval. For continuous time the energy is given by:

$$E_\infty = \lim_{\tau \rightarrow \infty} \int_{-\tau}^{\tau} |x(t)|^2 dt = \int_{-\infty}^{\infty} |x(t)|^2 dt \quad (5.34)$$

It can be defined the average normalized power in an infinite interval in the continuous time by the equation:

$$P_\infty = \lim_{\tau \rightarrow \infty} \frac{1}{2\tau} \int_{-\tau}^{\tau} |x(t)|^2 dt \quad (5.35)$$

It will be calculated the energy of the non-periodic single decay pulse signal $A * e^{-t/\tau_d}$ (equation 2.10 of chapter 2) by using equation 5.34.

Energy Calculation

$$E_{\infty} = \int_{-\infty}^{\infty} \left| (A * e^{-t/\tau_d})^2 \right| dt \quad (5.36)$$

The energy of the single pulse is given by the equation:

$$E_{\infty} = \frac{A^2 * \tau_d}{2} \quad (5.37)$$

It will be calculated the energy of the bi-exponential pulse signal ($A * (e^{-t/\tau_d} - e^{-t/\tau_r})$) (equation 2.8 of chapter 2) by using equation 5.34.

$$E_{\infty} = \int_{-\infty}^{\infty} \left| (A * (e^{-t/\tau_d} - e^{-t/\tau_r}))^2 \right| dt \quad (5.38)$$

The energy of the bi-exponential pulse is given by the equation:

$$E_{\infty} = \frac{A^2 * \tau_d}{2} + \frac{A^2 * \tau_r}{2} - \frac{2 * A * \tau_d * \tau_r}{\tau_d + \tau_r} \quad (5.39)$$

It can be observed in the equations 5.36 and 5.38 that the energy of the pulse is directly proportional to the voltage which represents the amplitude of the pulse (parameter A in equations 2.8 and 2.10 representing the input signals $x(t)$) and the decay (τ_d) and rise time (τ_r) constants. The amplitude of the trapezoidal filter is proportional to the voltage of the pulse and that is why the filter is used to calculate the energy.

Chapter 6

Results and Discussion

6.1. Introduction

In this chapter the accuracy of the filters implemented in chapters 4 and 5 will be determined by the following procedures:

1. In the $CR - (RC)^n$ filters the peak detection function is used to find the peaks of the pulses. It returns the x and y position of those peaks. The values of the x axes are compared with the x values returned by the zero crossing function when crossing the thresholds.
2. In the trapezoidal filter the peak detection function is used to find the voltage (amplitude) of the input pulse which is used to calculate the energy of the pulse. Afterwards a new value of the energy is calculated based on the height of the trapezoid output and those values are compared to determine the accuracy of the filter.
3. In the trapezoidal filter the height of the trapezoid output is compared with the height (or amplitude) of the pulse to determine the accuracy of the filter.

Those methods have been chosen to be able to compare the values of the parameters extracted by the filters with some reference values which are considered to be the ones extracted from the input signal.

In the first method the reference value is the time in μs when the pulse reaches its maximum voltage. The difference between this value and the zero crossing value from the bipolar signal generated by the filter can be considered as an offset value.

In the second and third methods the reference value is the height of the input pulse signal. The noise in the signal can alter this value because a peak of noise can be considered as a true reference so the signal has been filtered with the moving average filter before performing the tests.

The tests to determine the accuracy of the filters are performed by two reasons:

- Know the margin of error in the parameters calculated by the filters like the energy of the pulse and time stamp.
- Analyze and evaluate which filter will be the most appropriate to implement in hardware based on the accuracy.

The first part of the chapter will evaluate the precision of the $CR - (RC)^n$ filters and the accuracy of the filters in reducing the pile up between pulses.

The second part will evaluate the trapezoidal filters when the input signals are single exponential

and bi-exponential pulses. Initially the tests are performed with fix values of rise and decay time constants and after they are performed with changing values of those time constants.

Pulses from a radiation detector [15] are randomly spaced in time and this can result in interfering effects between those pulses when dealing with high rates. These effects are called pile-up and this problem will be addressed at the last part of this chapter. Two methods will be proposed to reduce pile-up between signals and the accuracy of those methods will be tested.

6.2. Distortion of Filter Output by Gaussian Noise

In this section it is analyzed the effect of the noise in the signals by adding noise of different intensity to the signals.

Figure 6.1 shows on the top the input signal in red which contains four single decay pulses with different amplitudes and where a Gaussian noise of $SNR = 20$ has been added. The output of the trapezoidal filter is situated at the bottom where it can be observed the trapezoid output.

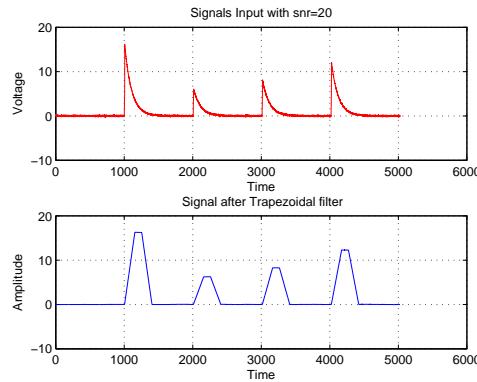


Figure 6.1: Trapezoidal filter output with added Gaussian noise ($SNR = 20$)

In the figure 6.2 a Gaussian noise of $SNR = 5$ has been added to the pulses and it can be observed that they have more noise added than the ones from figure 6.1.

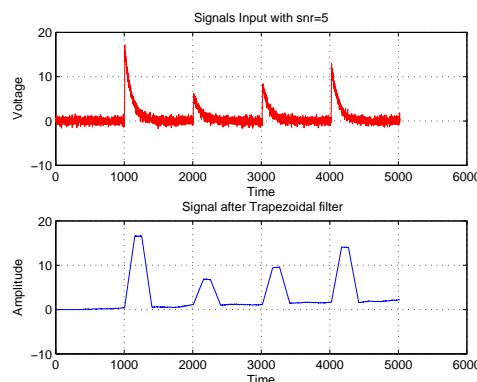


Figure 6.2: Trapezoidal filter output with added Gaussian noise ($SNR = 5$)

The height of the trapezoids have been measured in order to estimate the influence of the noise

on the output signal as shown in 6.1. This is important to evaluate the effect of the noise on the estimation of the energy because this is proportional to the height of the trapezoid outputs. The heights of the input generated single decay pulses which are colored in red are : 16, 6, 8 and 12 so the error caused by the added noise has been estimated based on those values.

Noise Added (SNR)	First T H	(Error %)	Second T H	(Error %)	Third T H	(Error %)	Fourth T H	(Error %)
1 Without added noise	16.29	1.81	6.23	3.83	8.24	3	12.26	2.16
2 20	16.20	1.25	6.03	0.5	7.93	0.875	11.81	1.58
3 5	16.44	2.75	6.82	3.66	8.97	12.12	13.32	10.16

Table 6.1: Table with trapezoid heights (T H) where noise has been added

The signal with an added noise of $SNR = 20$ presents errors which are even smaller than the signal without added noise. On the other hand the signal with added noise of $SNR = 5$ presents an error of 2.75 % in the decay pulse with height of 16, 3.66 % in the pulse with height of 6, 12.12 % in the decay pulse with 8 and 10.16 % in the pulse with 12. Those errors can be considered quite significant so in this case the output is affected by the added noise.

It can be concluded based on the above results that an amount of noise equal to $SNR = 20$ does not affect the output of the filter and an amount of noise equal to $SNR = 5$ produces an average error of 7.17 %. Therefore the signal has to be filtered and noise has to be removed before the signal pass through the trapezoidal and $CR - (RC)^n$ filters to minimize the error.

6.3. Accuracy of $CR - (RC)^n$ Filters

In this section will be described the process of measuring the precision of the $CR - (RC)^n$ filter of order 1,2,3 and 4 with a bi-exponential decay signal in order to determinate which one of the four filters is more accurate.

There have been performed eleven tests with the $CR - (RC)^n$ filters whose results are illustrated in two different tables:

- Table 6.2 contains the results of the test performed with the signal corresponding to figure 4.3. The input signal is the $adc1_{10}$ from appendix E which it is colored in red in the figure and it is situated in the top left. It has a rise time of $5 \mu s$ and a decay time of $10 \mu s$. The filter outputs are colored blue in the figure.
- Table 6.3 contains the average of the test results performed with ten different input signals from appendix E. The signals used for testing are:
 - $adc1_{10}$ with rise time equal to $5 \mu s$ and decay time equal to $10 \mu s$
 - $PE1$ with rise time equal to $0.1 \mu s$ and decay time equal to $5 \mu s$
 - $PE6$ with rise time equal to $0.1 \mu s$ and decay time equal to $10 \mu s$
 - $PE15$ with rise time equal to $0.1 \mu s$ and decay time equal to $20 \mu s$
 - $PE21$ with rise time equal to $0.2 \mu s$ and decay time equal to $5 \mu s$
 - $PE26$ with rise time equal to $0.2 \mu s$ and decay time equal to $10 \mu s$
 - $PE31$ with rise time equal to $0.2 \mu s$ and decay time equal to $20 \mu s$
 - $PE41$ with rise time equal to $0.5 \mu s$ and decay time equal to $5 \mu s$
 - $PE50$ with rise time equal to $0.5 \mu s$ and decay time equal to $10 \mu s$

Chapter 6. Results and Discussion

- PE71 with rise time equal to $1 \mu s$ and decay time equal to $20 \mu s$

The tables containing all the results of each individual test and the input signals used in those tests are in appendix C.

It is important to remark that the values set for the thresholds in each filter may affect the accuracy of the $CR - (RC)^n$ filters. Changing those values will change the measured points used to establish the accuracy of the filter. The values used for all the testings are the ones described in section 4.2.3 from chapter 4 and there have not been performed tests with different thresholds values.

The measurement points for the tests conducted in table 6.2 are the upper threshold crossing points when the output signal rises corresponding to P1 and P3 in the figure 4.3 and the lower threshold crossing points when the output signal decreases corresponding to P2 and P4.

The table contains six columns. Second and fourth columns contain the zero crossing time in μs between the output pulses and the thresholds corresponding to P1, P3 and P2, P4 which are the measurement points for each filter. The upper threshold crossing points when the output signal rises correspond to P1 and P3 in the figure 4.3 and the lower threshold crossing points when the output signal decreases correspond to P2 and P4.

Third and fifth columns contain the time offset of the filter in μs . The time is calculated by the following equations:

The sampling frequency is 125 Mhz so the sampling interval is calculated by the equation:

$$\text{Sampling interval } (Ts) = \frac{1}{\text{Sampling frequency } (fs)} \quad (6.1)$$

$$Ts = \frac{1}{125 \text{ Mhz}} = 8 \text{ Nanoseconds} \quad (6.2)$$

Then knowing the time between samples the time is calculating by the equation:

$$\text{Time} = n * Ts \quad (6.3)$$

Where n is the sample number.

The time offset is calculated based on the difference between the time when the voltage reaches its maximum value in the input pulses ($26.288 \mu s$ for the first pulse and $203.152 \mu s$ for the second pulse) and the time when the output signals cross the thresholds.

Finally the sixth column contains the peak average estimation time offset. This is the time offset on estimating the middle point between the values from column two and four and the time position of the maximum amplitude of the input pulse. It is calculated by the following equation:

$$\text{PeakEstimationTimeOffset} = \frac{\text{TimeOffset(column3)} + \text{TimeOffset(column5)}}{2} \quad (6.4)$$

The table presents the following results:

Filter Order	P1 and P3 Z-C t (μs)	Avg t Offset (μs)	P2 and P4 Z-C t (μs)	Avg t Offset (μs)	P E Avg t Offset (μs)
First order	-22.312 / -199.152	-3.988	30.008 / 206.832	3.700	- 0.144
Second order	-22.656 / -199.504	-3.640	30.936 / 207.760	4.628	0.494
Third order	-23.112 / -199.960	-3.184	31.968 / 208.800	5.664	1.24
Forth order	-23.512 / -200.360	-2.784	32.536 / 209.376	6.236	1.726

Table 6.2: Table with test results from $CR - (RC)^n$ filters with input signal $adc1_{10}$ where rise time = $5 \mu s$ and decay time = $10 \mu s$

It can be observed in the results that the $CR - (RC)^4$ filter is the most precise in the rising crossing points with a time offset of $2.784 \mu s$ and the $CR - (RC)^1$ filter is the most precise in the decaying crossing points with a time offset of $3.700 \mu s$. Filters in general are more accurate in P1 and P3 with an average between the four filters of $3.4015 \mu s$ than with P2 and P4 with an average of $5.057 \mu s$.

Observing the peak estimation time offset it can be concluded that the first order filter is the most accurate on estimating the peak of the input signal with a time offset of $-144 ns$ and the forth order filter presents the worst accuracy with a time offset of $1726 ns$.

Ten additional tests have been performed in order to have a larger number of results available. Table 6.3 contains the filter order together with the average time offset in μs of those tests. Second column contains the average time offset of P1 corresponding to the crossing with the upper threshold and third column contains the average offset of the filter in μs of P2 corresponding to the crossing with the lower threshold. The forth column contains the peak estimation average time offset in μs .

Filter Order	P1 Avg t Offset (μs)	P2 Avg t Offset (μs)	P E Avg t Offset (μs)
First order	-3.479	3.579	0.050
Second order	-3.507	4.510	0.501
Third order	-3.502	5.516	1.007
Forth order	-2.913	6.257	1.672

Table 6.3: Table with test results from $CR - (RC)^n$ filters with Matlab generated input signals where rise time = 0.1 to $5 \mu s$ and decay time = 5 to $20 \mu s$

These results are similar to the ones obtained in table 6.2. It can be observed that the $CR - (RC)^4$ filter is also the most precise in the rising crossing points with a time offset of $2.913 \mu s$ and the $CR - (RC)^1$ filter is the most precise in the decaying crossing points with a time offset of $3.579 \mu s$. The average of the four filters in P1 is $3.350 \mu s$ and in P2 the average is $4.960 \mu s$. Finally the first order filter is the most accurate with a time offset of $50 ns$ and the forth order filter is the less accurate with a time offset of $1672 ns$.

6.4. Accuracy of Trapezoidal Filter with Fix Rise Time and Fix Decay Time

In this section it is described the process of measuring the accuracy of the trapezoidal filters implemented in chapter 5. The input signals used for testing have a fix rise and decay times and they are generated in Matlab in section 6.4.1 and with the pulse generator in section 6.4.2. This is important because the energy of the pulse is proportional to the amplitude of the trapezoid output as explained in 5.5. This means that the accuracy of the filter will directly influence the accuracy of the measured energy.

6.4.1 Trapezoid Accuracy of Single Exponential Decay Pulse

The accuracy of four different implementations of the trapezoid filter when the input signal is a single decay pulse is described in this section. The input signal $x(t)$ used for testing is generated with Matlab and it has a very short rise time close to zero and a decay constant value of $15 \mu s$.

The trapezoidal filters generated by recursive equations and with single decay transfer function

which corresponds to the first two filters of the table 6.4 were implemented respectively in sections 5.2.1 and 5.2.1 of chapter 4 and they correspond to figure 5.4.

Furthermore two additional measurements have been performed with the same single decay pulse where the trapezoidal filters generated by transfer function biexponential (equation 5.30) and by transfer function model based (equation 5.30) have been used. These filters correspond to the last two rows of table 6.4.

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	1674.48	96.63
Trapezoidal filter TF single	165.59	97.77
Trapezoidal filter TF biexponential	1693.21	95.48
Trapezoidal filter TF model based	1620	100

Table 6.4: Table with accuracy percentage of trapezoidal filter with Matlab generated input signal where rise time $\simeq 0$ and $\tau_d = 15 \mu s$

The table shows in the second column the energy measured by each implementation of the trapezoidal filters and in the third column the accuracy of those measurements based on the value of 1620 Joules which corresponds to the energy measured from the input signal.

It is observed in the results that the trapezoidal with the model based transfer function has an accuracy of 100 % and it calculates the same energy as it was obtained from the input signal. If comparing the trapezoid generated by recursive equations with the one generated by single decay transfer function, which have been designed to deal with single exponential pulses, it is possible to see that the second is 1.14 % more precise.

6.4.2 Trapezoid Accuracy of Bi-Exponential Pulse

The accuracy of the same four trapezoid filters from section 6.4.1 is described here but now the input signals are ten bi-exponential pulses generated with the pulse generator from appendix C instead of the single decay pulses.

The signals used for testing are:

- $adc1_{10}$ with rise time equal to $5 \mu s$ and decay time equal to $10 \mu s$
- $PE1$ with rise time equal to $0.1 \mu s$ and decay time equal to $5 \mu s$
- $PE6$ with rise time equal to $0.1 \mu s$ and decay time equal to $10 \mu s$
- $PE15$ with rise time equal to $0.1 \mu s$ and decay time equal to $20 \mu s$
- $PE21$ with rise time equal to $0.2 \mu s$ and decay time equal to $5 \mu s$
- $PE26$ with rise time equal to $0.2 \mu s$ and decay time equal to $10 \mu s$
- $PE31$ with rise time equal to $0.2 \mu s$ and decay time equal to $20 \mu s$
- $PE41$ with rise time equal to $0.5 \mu s$ and decay time equal to $5 \mu s$
- $PE50$ with rise time equal to $0.5 \mu s$ and decay time equal to $10 \mu s$
- $PE71$ with rise time equal to $1 \mu s$ and decay time equal to $20 \mu s$

The tables containing all the results of each individual test and the input signals used in those tests are in appendix C.

The trapezoidal filters are implemented in section 5.2.3 of chapter 5 and they are illustrated in figure 5.8.

Table 6.5 shows the average values of the results of the tests performed to the ten bi-exponential pulses. The second column contains the average accuracy of the calculated energy extracted by each type of trapezoidal filter.

Filter Type	Accuracy in %
Trapezoidal filter recursive	80.82
Trapezoidal filter TF single	85.76
Trapezoidal filter TF biexponential	76.14
Trapezoidal filter model based	98.19

Table 6.5: Table with accuracy percentage of trapezoidal filter with Matlab generated input signals where rise time = 0.1 to 5 μs and decay time = 5 to 20 μs

It can be observed how the filter with the best performance is the trapezoid with model based transfer function with an accuracy of 98,19 %. The error of 1.81 % is due to the fact that the input signal does not fit 100 % the bi-exponential model of equations as it is illustrated in figure 5.9.

The trapezoidal filter with bi-exponential transfer function has an accuracy of 76.14 %. If taking into account the error of 1.81 % which is not caused for the filters then the accuracy of all filters rise that value.

The first and the second filters are not implemented to deal with rise times and the input signals used to perform the tests have different rise time values as it is possible to observe in appendix C but the accuracy are 80.82 % and 85.76 %. If the rise time of the input signal decreases then the filters will perform more accurately as it will be shown in section 6.5.

6.5. Accuracy of Trapezoidal Filter with Rise Time and Decay Time Variations

It is addressed in this section the process of determining the accuracy of the filters tested in sections 6.4.1 and 6.4.2 but in this case the rise and decay times change. The precision measured in those sections was performed with different fix rise and decay time constants. It is investigated here if the filters perform better or worse when the rise and decay time constant values increase or decrease.

This section contains the results performed with single exponential pulses in subsection 6.5.1 and bi-exponential pulses in subsection 6.5.2. The signals used to perform the tests are generated with Matlab because τ_d and τ_r changes have to be simulated.

6.5.1 Single Decay Exponential Pulse Test

It is explored here the precision of the trapezoidal filters implemented with recursive equations and transfer function when the input signal is a single decay pulse. Single exponential pulses are generated with Matlab and they have a very short rise time close to zero and decay time constant

values from 0 to $50 \mu s$. These two filters are illustrated in figure 5.4 and they are designed to extract the amplitude of single exponential pulses and that is why they are chosen for testing.

6.5.1.1 Recursive Algorithms Implementation

The trapezoidal filter generated by recursive equations which was implemented in section 5.2.1 of chapter 5 will be tested here.

Figure 6.3 shows in green the height of the exponential function when the decay time constant τ_d increases from 0 to $50 \mu s$. It is possible to see that the input function has a constant height value and this means that the amplitude of the pulse is independent of the value of τ_d .

The height of the trapezoid output is shown in blue and it is observed that the filter performs better when τ_d increases. The error area is large with values of τ_d between 0 and $20 \mu s$ than with values of τ_d between 20 and $50 \mu s$.

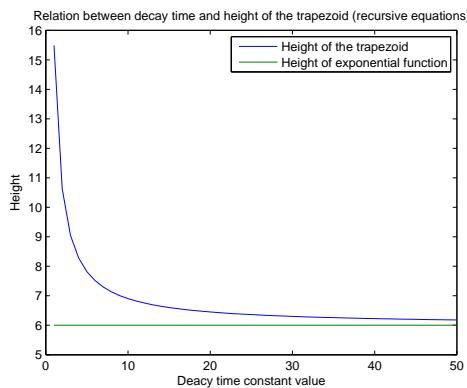


Figure 6.3: Variations in trapezoidal output when $\tau_r \approx 0$ and τ_d changes from 0 to $50 \mu s$ (Recursive Algorithms)

6.5.1.2 Transfer Function Implementation

The trapezoidal filters generated by transfer function single decay which was implemented in section 5.2.1 of chapter 5 will be tested here.

Figure 6.4 shows the accuracy of the filter when the value of τ_d increases. The filter behaves like the recursive algorithm implementation but it is observed here that the maximum error with values of τ_d close to zero is 9.7 instead of 15.5 in the recursive implementation which means that this filter is more precise with small values of τ_d .

Transfer Function Model Based

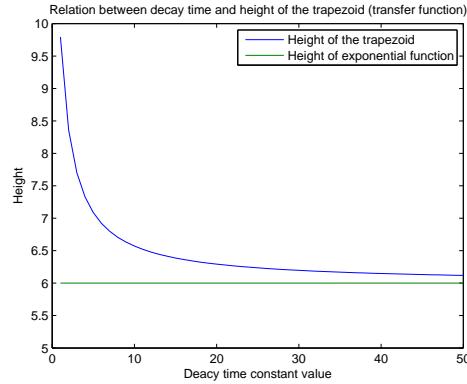


Figure 6.4: Variations in trapezoidal output when $\tau_r \approx 0$ and τ_d changes from 0 to 50 μs (Transfer functions)

6.5.1.3 Transfer Function Model Based

The trapezoidal filters generated by transfer function model based will be tested here. The transfer function implemented to deal with bi-exponential pulses has been used here to extract the amplitude of the Matlab generated single decay pulse.

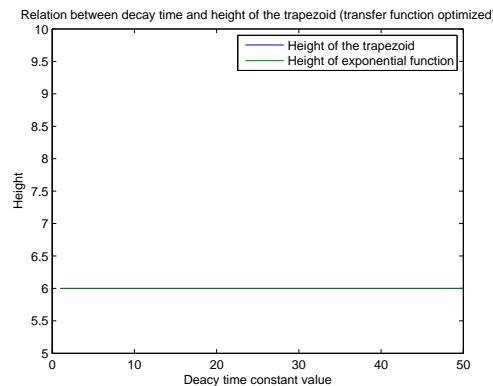


Figure 6.5: Variations in trapezoidal output when $\tau_r \approx 0$ and τ_d changes from 0 to 50 μs (model based transfer functions)

Figure 6.5 shows only a green line representing the height of the single exponential function because this lies over the blue line representing the height of the trapezoid output. Table 6.6 shows the total error of the filters which is estimated based on the area between the two curves.

Filter Type	Error (area between curves)
Trapezoidal filter recursive	36.45
Trapezoidal filter TF single	21.13
Trapezoidal filter TF model based	≈ 0

Table 6.6: Table with errors of trapezoidal filters when decay time constant changes in single exponential pulse

It is observed in the results that the trapezoidal filter with the model based transfer function has a

total error of almost zero. If compared with the first two filters of the table then the trapezoidal filter obtained from the transfer function is more accurate than the one obtained from recursive equations. Also as observed in table 6.4 the second filter was a bit more accurate than the first one with fix τ_d so it is conclude that in general the transfer function implementation is more precise in estimating the amplitude of the pulse.

6.5.2 Bi-Exponential Pulse Test

It is explored in this section the precision of the trapezoidal filters when the input signal is a bi-exponential pulse.

Bi-exponential pulses are generated with Matlab and they have the following time constant values:

- Fix decay time constant value equal to $15 \mu s$ and rise time constant value from 0 to $10 \mu s$
- Fix rise time constant value equal to $0.001 \mu s$ and decay time constant values from 0 to $50 \mu s$
- Fix rise time constant value equal to $1 \mu s$ and decay time constant values from 0 to $50 \mu s$

The same filters tested in section 6.4.2 with fix τ_d values and illustrated in figure 5.8 will be tested here.

6.5.2.1 Recursive Algorithms Implementation

Figure 6.6 shows the relation between the rise time constant τ_r and the height of the trapezoid implemented with recursive equations and figure 6.7 shows the relation between the fall time constant τ_d and the height of the trapezoid.

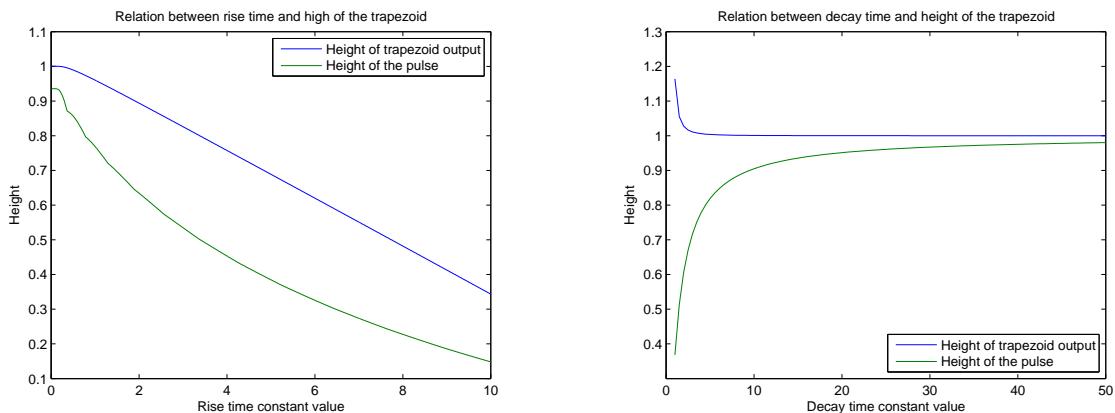


Figure 6.6: Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (Recursive Algorithms)

Figure 6.7: Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to $50 \mu s$ (Recursive Algorithms)

The error (area between the curves representing the height of the function and the height of the trapezoid output) when the rise time constant changes is 2.545 and 3.622 when the decay time constant changes. The range of the rise time constant which has been taken in consideration is from 0 to $10 \mu s$ because pulses have normally a very short rise time. The value of the error is proportional to the area where it is has been calculated from so even the value of the first error is smaller than the value of the second the accuracy of the first figure is worse than the second

TF Single Decay Implementation

because this error has been calculated from a smaller area.

6.5.2.2 TF Single Decay Implementation

Figures 6.8 and 6.9 show the relation between the rise constant τ_r and decay time constant τ_d and the height of the trapezoid implemented with transfer function.

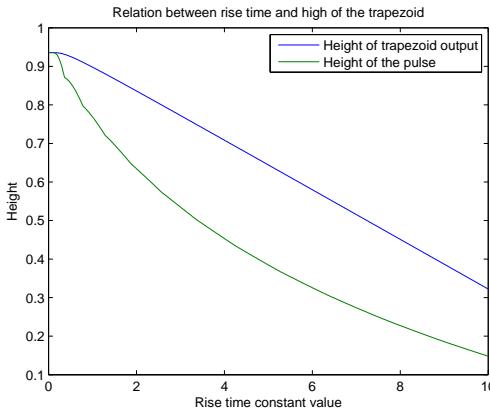


Figure 6.8: Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (TF single decay)

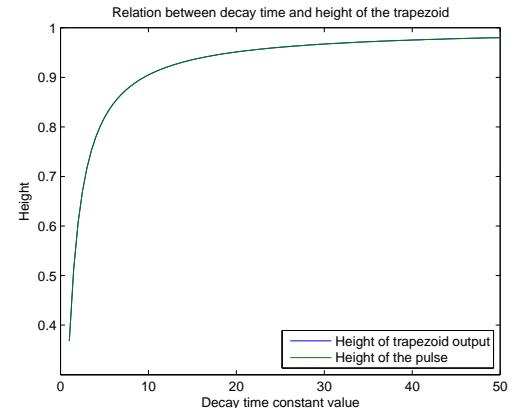


Figure 6.9: Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to $50 \mu s$ (TF single decay)

The filter behaves very similar to that of the one implemented with recursive equations where the error in rise time variations is quite small with a value of 2.102 but in this case the error with decay time variations is very small with a value of $1.7*e^{-13} \simeq 0$. Observing the figure it is possible to see that the green line lies over the blue line in the figure of the right due to the small error so only the green line is displayed.

Figures 6.10 and 6.11 illustrate the result of the test performed when the rise time constant has a value $1 \mu s$ instead of $0.001 \mu s$ while the decay time constant changes from 0 to $50 \mu s$.

The purpose of increasing the value of τ_r is to observe how the filter behaves.

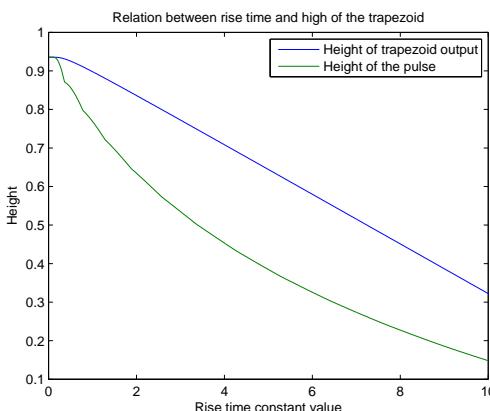


Figure 6.10: Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to $10 \mu s$ (TF single decay)

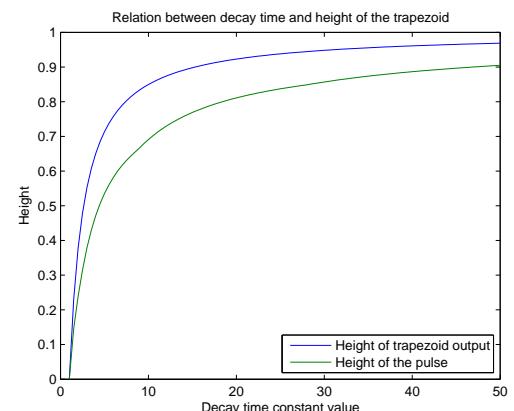


Figure 6.11: Variations in trapezoidal output when $\tau_r = 0.001 \mu s$ and τ_d changes from 0 to $50 \mu s$ (TF single decay)

It can be observed that the accuracy of the filter decays when the rise time increases. The error area of the relation between decay time constant and high of the trapezoid increases from $1.7 \cdot e^{-13}$ to 5.279 and with the relation between rise time constant and high of the trapezoid is 2.102.

6.5.2.3 TF Bi-exponential Implementation

Figures 6.12 and 6.13 show the relation between the rise and decay time constants and the height of the trapezoid implemented with bi-exponential transfer function.

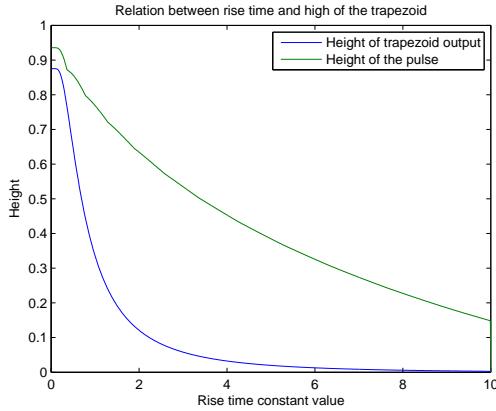


Figure 6.12: Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to 10 μs (Biexponential transfer function)

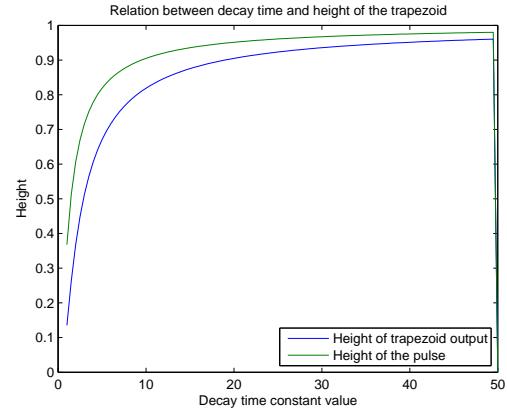


Figure 6.13: Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to 50 μs (Biexponential transfer function)

It can be observed in the figures that the filter behaves very similar to that of the ones implemented with the transfer function and with recursive equations. The error area is 3.28 for rise time variations and 2.86 for the decay time variations.

Figures 6.14 and 6.15 show the relation between the rise and decay time constants and the height of the trapezoid when the rise time constant value is set to 1 μs instead of 0.001 μs .

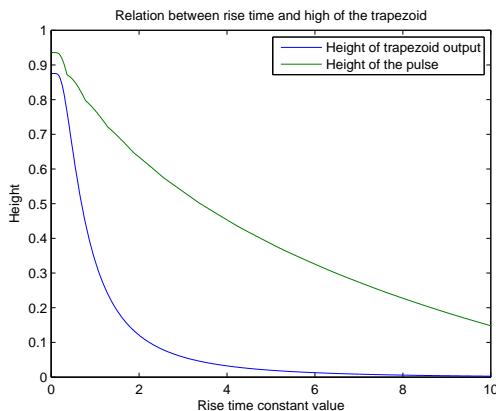


Figure 6.14: Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to 10 μs (Biexponential transfer function)

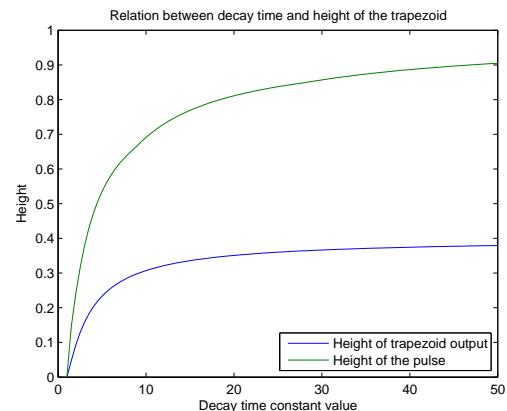


Figure 6.15: Variations in trapezoidal output when $\tau_r = 0.001 \mu s$ and τ_d changes from 0 to 50 μs (Biexponential transfer function)

The error area of the relation between decay time constant and height of the trapezoid increases

from 2.86 to 21.81 and the error area of the relation between rise time and high of the trapezoid is 3.28 so it is confirmed again that increasing the rise time affects the accuracy of the filter.

6.5.2.4 TF Bi-exponential Model Based Implementation

Figures 6.16 and 6.17 show the relation between the rise and the decay time and the height of the trapezoid implemented with bi-exponential transfer function model based.

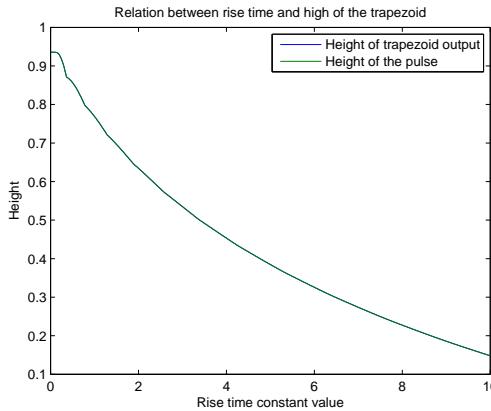


Figure 6.16: Variations in trapezoidal output when $\tau_d = 15 \mu s$ and τ_r changes from 0 to 10 μs (Model based transfer functions)

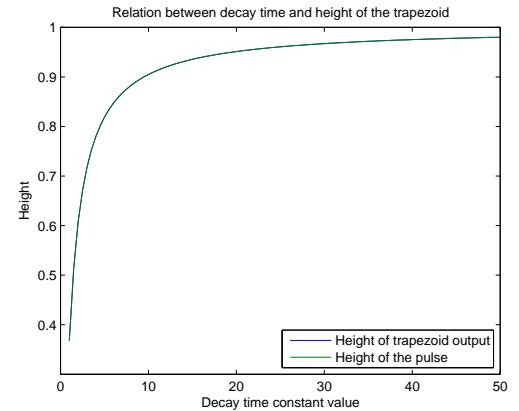


Figure 6.17: Variations in trapezoidal output when $\tau_r = 1 \mu s$ and τ_d changes from 0 to 50 μs (Model based transfer functions)

The error area of the relation between decay time constant and height of the trapezoid is $4.97 * e^{(-14)} \simeq 0$ and the error area of the relation between rise time and height of the trapezoid is $1.06 * e^{(-14)} \simeq 0$. The figures show only the green line representing the height of the pulse because these results are very close to zero.

6.5.2.5 Summary of the Results

Tables 6.7 and 6.8 show the error of the five types of trapezoidal filters implemented to have a better overview of the results and to be able to compare them. Table 6.7 shows the results of the tests performed to the trapezoidal filters with fix rise time constant τ_r equal to $0.001 \mu s$ while decay time constant τ_d changes from 0 to $50 \mu s$ and table 6.8 shows the results of the tests performed with fix rise time constant τ_r equal to $1 \mu s$ while decay time constant τ_{rd} changes from 0 to $50 \mu s$.

The first column contains the type of trapezoidal filter, the second columns contains the error area between the curves generated when the rise time constant changes, the third column contains the error area between the curves generated when the decay time constant changes and the forth contains the average value obtained from column two and three.

Filter type	τ_r changes error	τ_d changes error	Avg error
Trapezoidal filter recursive ($\tau_r=0.001 \mu s$)	2.524	3.622	3.073
Trapezoidal filter TF single ($\tau_r=0.001 \mu s$)	2.102	$17.05 * e^{(-13)}$	1.051
Trapezoidal filter TF biexponential ($\tau_r=0.001 \mu s$)	3.280	2.860	3.070
Trapezoidal filter TF biexponential model based ($\tau_r=0.001 \mu s$)	$\simeq 0$	$\simeq 0$	$\simeq 0$

Table 6.7: Table with test results from trapezoidal filters when $\tau_r=0.001 \mu s$

Filter type	τ_r changes error	τ_d changes error	Avg error
Trapezoidal filter recursive ($\tau_r=1 \mu s$)	2.524	8.220	5.372
Trapezoidal filter TF single ($\tau_r=1 \mu s$)	2.102	5.279	3.690
Trapezoidal filter TF biexponential ($\tau_r=1 \mu s$)	3.280	21.810	12.545
Trapezoidal filter TF biexponential model based ($\tau_r=1 \mu s$)	$\simeq 0$	$\simeq 0$	$\simeq 0$

Table 6.8: Table with test results from trapezoidal filters when $\tau_r=1 \mu s$

The trapezoidal filter TF biexponential and TF biexponential model based are implemented to deal with rise times as explained in 5.2.2.2 of chapter 5 and the TF single and recursive implementations are not implemented to deal with rise times. However if compared the average error of the third filter when $\tau_r=1 \mu s$ (12.545) with the errors obtaining by the first and second filters (5.372 and 3.690) it is possible to see that these two are more accurate in its measurements.

Observing the average error from table 6.8 it is possible to see that the accuracy of all the filters decreases when τ_r increases and the trapezoidal filter TF biexponential presents the worst accuracy (12.545).

It can be observed how the trapezoidal filter model based presents the best accuracy for rise and decay time changes with error values very close to zero.

The second filter with the best accuracy is the one implemented with the TF single function. This filter presents the second smallest error area in τ_r changes error (2.102), in τ_d changes error ($17.05 * e^{(-13)}$) and in total average error (1.051). The same applies when $\tau_r=1 \mu s$ where the total average error is 3.690. It can be concluded that this filter is very precise on estimating the amplitude of the pulse with rise time values close to zero.

The third filter with the best accuracy is the one implemented with recursive algorithms with average errors of 3.073 and 5.372.

These results validate the ones obtained in table 6.5 where the filters were tested with fix τ values.

6.6. Pile Up Reduction

The problem of pile up in pulses and how to minimize it will be explained in this section.

When two particles arrive at the detector within the width of the shaping amplifier output pulse, their respective pulses pile up forming an output pulse where the height is distorted. The pile-up pattern changes depending on the difference in time between two pulses. When the second pulse comes relatively late and rides on the falling tail of the first pulse, the rising edge and the height of the first pulse is not affected then the first event can be processed without problem. However, when the second pulse arrives relatively early and rides on the rising edge of the first pulse then both pulses form a single pulse with a slower rising edge.

It is proposed two ways to address this problem which occurs frequently when working with high frequencies.

6.6.1 Minimize Pile Up by Reduction of Flat Time in Trapezoidal Output

Pile up is illustrated in the top of figure 6.18 (colored red) where we can observe that there are three pulses. The second pulse rides on the falling tail of the first pulse and the third pulse is not distorted. The figure of the bottom left shows the output of the trapezoidal filter where the first trapezoid is composed by two flat tops corresponding to the first and the second pulses. We can conclude observing the figure that the output is also distorted by pile up.

Minimize Pile Up by $CR - (RC)^n$ filter

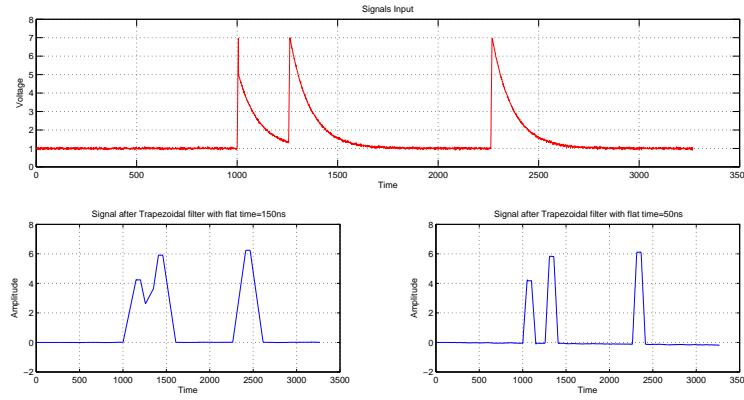


Figure 6.18: Trapezoidal outputs with different flat times

At the bottom bottom right the flat time has been reduced from 150 ns to 50 ns and the trapezoidal output has been restored.

By these observations we can conclude that the pile up in the output originated by two pulses can be reduced by decreasing the flat time in the trapezoid as much as possible.

6.6.2 Minimize Pile Up by $CR - (RC)^n$ filter

$CR - (RC)^n$ filters can be also used to handle the pile up pulses. The difference with the previous solution is that this filter is used to identify and count the pulses and not for measuring the energy because the height of the filter is independent to the height of the pulses.

The input signal used for testing is the $adc1_{16}$ from appendix E which has a rise time of $2\text{ }\mu\text{s}$ and a decay time of $20\text{ }\mu\text{s}$. Figure 6.19 illustrates in red the pile up problem having four exponential pulses close to each other. The figures colored in blue show the $CR - (RC)^n$ outputs for orders from $n=1$ to $n=4$. The crossing points with the upper threshold corresponding to the four pulses are represented in the figure 6.19 by the four upper red dots (P1,P3,P5 and P7). The crossing points with the lower threshold are represented by the four upper red dots (P2,P4,P6 and P8). It can be

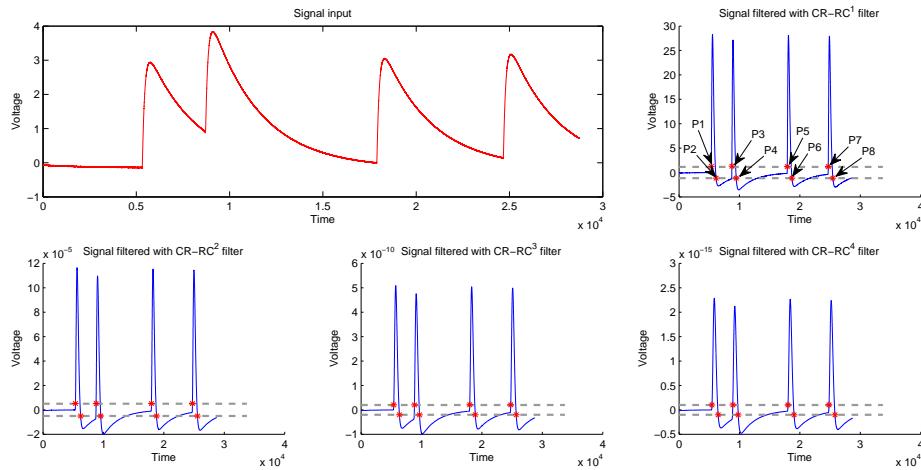


Figure 6.19: Input signals with pile-up and $CR - (RC)^n$ filter outputs

observed in the figure that all pulses are converted to bipolar pulses by the filter even though the input signal presents pile-up. It cannot be determined yet if those outputs generated by the filter are affected by the pile-up of the input signal. In section 6.6.3 the accuracy of the output will be tested in order to determine if this has been distorted by pile-up.

By these observations it can be concluded that the pile up can be corrected by the $CR - (RC)^n$ filter and all the pulses from signal $adc1_{16}$ can be recognized in the output.

6.6.3 Accuracy $CR - (RC)^n$ Filter for Pile-Up

Here it is measured the accuracy on reducing pile-up between two or more pulses with $CR - (RC)^n$ filters. The input signal is the $adc1_{16}$ from figure 6.19 which has a rise time of $2 \mu s$ and a decay time of $20 \mu s$.

Table 6.9 contains the results of the test performed when the pulses present pile up because the time interval between them is very short. It shows the measurements and times offset of the output signals of figure 6.19.

The table is the same as table 6.2 but there are eight measurement points instead of four. The first column displays the filter order. The second column contains the crossing points corresponding to (P1,P3,P5 and P7). The third and fifth column displays the calculated time offset values of these measurements based on the following time positions for the maximum voltages of the input pulses: 86.008,112.72,186.624 and 240.688 μs . Fourth column represents the four lower red points of figure 6.19 (P2,P4,P6 and P8) and fifth column the peak estimation average time offset .

Filter Order	P1,P3,P5 and P7 Z-C t (μs)	Avg t Offset (μs)	P2,P4,P6 and P8 Z-C t (μs)	Avg t Offset (μs)	P E Avg t Offset (μs)
First order	-82.728 / -109.936 / -183.304 / -237.568	-3.126	89.064 / 115.712 / 189.560 / 243.728	3.006	- 0.060
Second order	-82.976 / -110.272 / -183.568 / -237.848	-2.844	89.816 / 116.464 / 190.312 / 244.496	3.762	0.459
Third order	-83.264 / -110.664 / -183.880 / -238.176	-2.514	90.768 / 117.408 / 191.264 / 245.440	4.710	1.098
Forth order	-83.640 / -111.136 / -184.272 / -238.592	-2.100	91.792 / 118.400 / 192.280 / 246.456	5.722	1.811

Table 6.9: Table with test results from $CR - (RC)^n$ filters with input signal $adc1_{16}$ where rise time = $2 \mu s$ and decay time = $20 \mu s$

It is observed that averaging the third column then a time offset of $-2.646 \mu s$ is obtained and averaging the fifth column then $4.300 \mu s$ is obtained so it is concluded based on the above results that the measurements of column two are $1.654 \mu s$ more accurate on average than the measurements of column four. Also it is confirmed that the first measurement of the $CR - (RC)^4$ filter is the most precise. If taking in consideration both measurements of each filter and averaging them then the following results are obtained:

- $3.066 \mu s$ time offset for $CR - (RC)^1$
- $3.303 \mu s$ time offset for $CR - (RC)^2$
- $3.614 \mu s$ time offset for $CR - (RC)^3$
- $3.911 \mu s$ time offset for $CR - (RC)^4$

Based on the above results and observing the peak estimation average time offset values it can be confirmed that the $CR - (RC)^1$ is the most precise with an average time offset of $-60 ns$.

Table 6.10 shows the peak estimation average time offset values from tables 6.2, 6.3 and 6.9.

Summary of the Results

Filter Order	P E Avg t Offset (μs) table 6.2	P E Avg t Offset (μs) table 6.3	P E Avg t Offset (μs) table 6.9
First order	-0.144	0.05	- 0.060
Second order	0.494	0.501	0.459
Third order	1.24	1.007	1.098
Forth order	1.726	1.672	1.811

Table 6.10: Table with Peak Estimation Average Time Offset from tables 6.2, 6.3 and 6.9

The input signals used in tables 6.2 and 6.3 are pulses without pile-up and the signal used in table 6.9 presents pile-up. The peak estimation values from the tables are very similar so it can be concluded that pile-up in the input signal $adc1_{16}$ does not distort the output generated by the $CR - (RC)^n$ filter.

6.7. Summary of the Results

Various tests have been conducted to explore the behavior and the accuracy of the different implementations of the $CR - (RC)^n$ and trapezoidal filters.

Observing the testing results of the $CR - (RC)^n$ filters we can conclude that:

- The $CR - (RC)^1$ is the most accurate filter on average within the $CR - (RC)^n$ filters for orders from 1 to 4 with an accuracy of 50 ns.
- The $CR - (RC)^1$ is the most accurate filter within the $CR - (RC)^n$ filters for orders from 1 to 4 when measuring pulses with pile up with an accuracy of 60 ns.

Observing the testing results of the trapezoidal filters the following can be concluded:

- The trapezoidal filter TF biexponential model based is the most accurate to extract the energy from the pulse with an accuracy of 100 %. An evaluation should be performed in order to determine whether the filter can be implemented in the FPGA or not.
- The trapezoid parameters should be:
 - As large as possible to increase the rise time
 - As short as possible to decrease the flat time

This will enhance the filtering and avoid pile up as explained in section 6.6.1.

The testing results of the noise added to the input signals show that:

- Values of $SNR = 20$ have no influence on the filter output
- Values of $SNR = 5$ decreases the accuracy of the filter by 7.17 %

The results of the tests provides an overview of which filters are the most accurate to be implemented in the FPGA. However the results do not provide information about the performance of the different filters once they are implemented in the FPGA.

Chapter 7

Conclusion and Future Work

7.1. Introduction

This chapter concludes the work conducted in this thesis. This thesis project demonstrated that it was possible to implement a data acquisition model that will be able to calculate the energy of pulse signals.

First this chapter evaluates in section 7.2 the goals presented in chapter 1 with respect to the achievements of the thesis project. Afterwards, section 7.3 presents the achieved results and evaluates the thesis goals. After this, future work is discussed in section 7.4, and finally concluding remarks are given in section 7.5.

7.2. Revisiting to the Goals of the Thesis

In order to evaluate the achievement of this thesis project goals were set. The goals of this thesis were presented in chapter 1, and shown again below:

Goal 1: To propose a data acquisition model for experimental physics applications that will be able to calculate the time and the energy of pulse signals generated by a pulse generator.

Goal 2: To evaluate which of the proposed solutions is more accurate on estimating the energy and the time of the pulse.

Goal 3: Improve personal skills in general for being able to conduct research by identifying transferable skills that may be reusable in the future.

7.3. Evaluation of the Achievement of the Goals

In the remaining part of this section, the achieved results are evaluated by revisiting the thesis goals. First, it is explained why these goals are considered to be achieved. Afterwards, the main contributions of the report are presented and related to the goals of this thesis project.

7.3.1 Goal 1: Data acquisition model

This goal is *successfully* achieved, by the analysis carried out in chapters 4 and 5.

These chapters presented the implementation of the filters used to extract the energy and the time of the pulse signals. At first the theory was explained in detail and then the filters were implemented in Matlab. The equations and transfer functions used to generate the filter outputs were referred to the theory.

7.3.2 Goal 2: Evaluate the proposed solutions

This goal is *successfully* achieved, by the analysis carried out in chapter 6.

This chapter presented the results of all the test performed to the filters implemented in chapters 4 and 5. The signals generated in chapter 2 were used to perform the tests and the results of these tests were used to evaluate the proposed solutions based on the accuracy of the filters.

There have been no considerations in the evaluations of the different filters about the hardware resources needed to implement the filter in the FPGA or the time used for each filter to calculate the parameters.

7.3.3 Goal 3: Improve transferable skills

This goal is considered to be *successfully* achieved due to the work carried out in this thesis project, and the transferable skills developed are discussed below. This thesis is the result of conducting research on a scientific topic for the first time by the author. This goal focuses on transferable skills acquired during the work of this thesis project, that will be useful for future development. The skills acquired are divided in three main parts, and it is discussed why they are considered to be transferable.

7.3.3.1 Acquiring New Information

Acquiring new information was a great challenge due to the lack of not much information exist in pulse shape and height analysis.

The first part of the thesis was focused on getting an overview of the existing literature and selecting the relevant parts for this thesis.

In order to choose the relevant literature properly a strategy was used. The index was read at first with the books and previous thesis in order to decide if it was worth to start reading them and the abstract was read at first in the case of research articles. The selected material was saved in a folder for later review and some material was discarded after the theory was implemented in Matlab because it was found out that it was wrong. This work was not useless because it provided knowledge about the topic.

The transferable skill developed is the skill of becoming a better and more effective researcher at deriving information.

7.3.3.2 Improving Digital Signal Processing Skills

Improving the DSP (Digital Signal Processing) skills was very important in order to understand the mathematics behind the Matlab implementation. The signals are generated by exponential equations and the filter outputs are generated by recursive algorithms and transfer functions. There is a mathematical process to obtain these algorithms and transfer functions that has to be well understood.

The knowledge of DSP was limited to one course during the bachelor education and two courses during the master education.

The transferable skill developed is the skill of obtaining the equations and transfer functions used to generate the filter outputs.

7.4. Future work

The discussion of future work is divided into two parts. First, future work that can be done in relation to the acquisition model implemented in this thesis. Afterwards the second section presents the future work that has to be done in order to complete the development of the digital pulse detection.

7.4.1 Digital Pulse Detection Model

A model based transfer function is implemented in this thesis where the peak detection function has been used to calculate the height of the pulse. Peak detection function can be used in the case that there is no noise.

However exponential signals always have noise superimpose on it because of:

1. Statistical processes [22] in detector (monoenergetic radiation does not produce always equal number of electrons and holes in semiconductor detector or ions in gas proportional detector or light photons in scintillator)
2. electronic noise (electronic components including detector)

What can it be done in the case of noise? There are two possibilities:

1. Take exponential signal and calculate its amplitude by using peak detection function in Matlab
2. Change the shape [28] of the signal into any other (trapezoidal, rectangle, Gauss,etc) and then calculate the amplitude (using the peak detection function)

In any case, the results when applying the same method for two or more pulses will fluctuate around the expected value in the case of no noise. Also it will be observed that the amount of fluctuation [13] depends on the shape of the signal that it is used. Of course the goal is to obtain this fluctuation is as small as possible.

The question is: what shape is optimal or for which shape the fluctuation is minimal? Future work could be done on exploring different shapes to obtain the energy from a pulse and evaluate the accuracy of these shapes.

The values set for the thresholds the $CR - (RC)^n$ filter may affect the accuracy of those. The values used for all the testings are the ones described in section 4.2.3 from chapter 4 and there have not been performed tests with different thresholds values. Future work could be done on testing the filters with different threshold values in order to determine if the accuracy is affected.

7.4.2 Implementation in the FPGA

The development process to implement the digital pulse detector in the FPGA is described in this section. Figure 7.1 illustrates this process where the board used as a example is the Zynq board

[36] which is a new kind of device which combines both FPGA fabric and a capable applications processor.

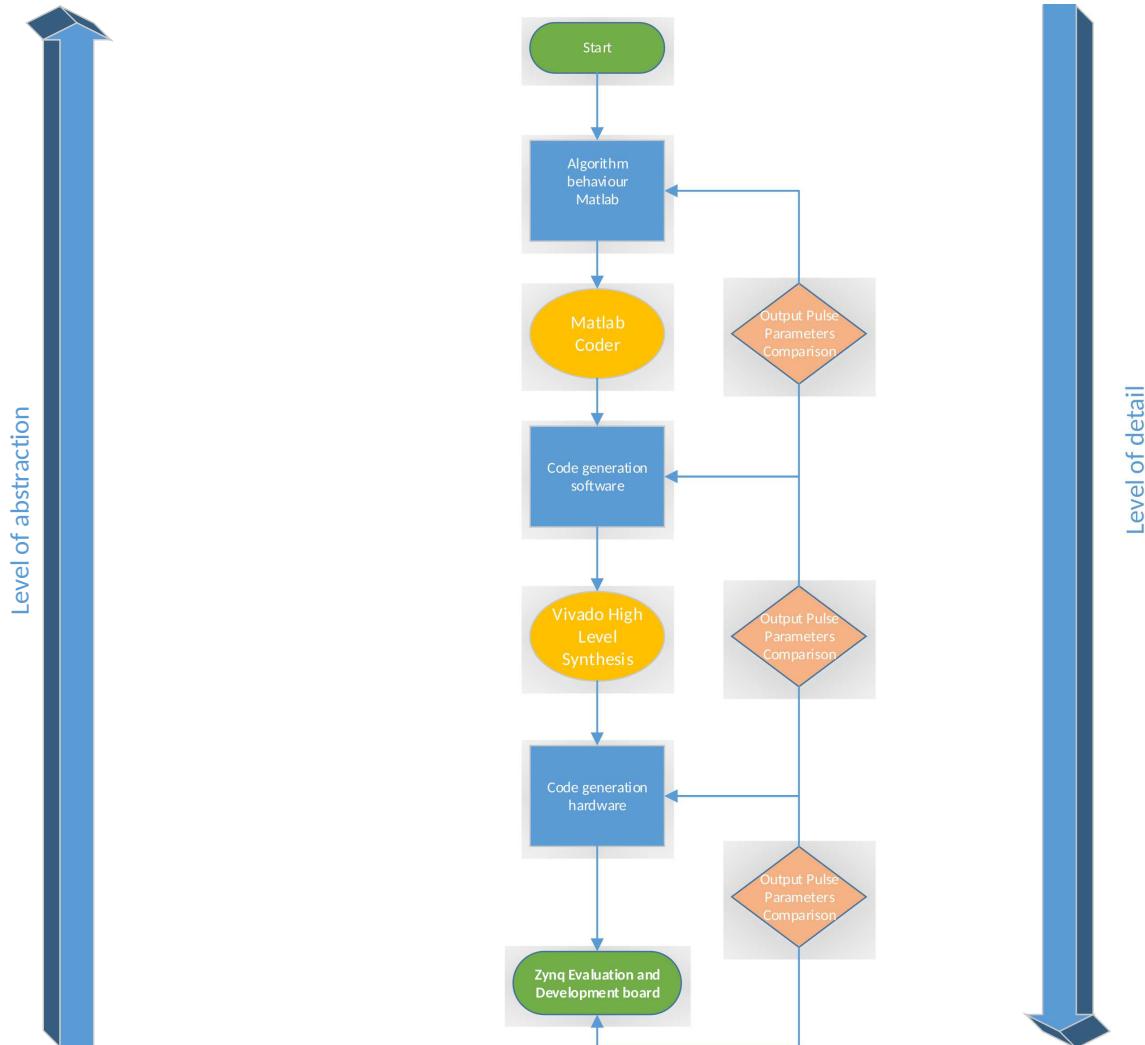


Figure 7.1: Process of the system [1]

The process is divided in three phases:

1. Algorithm behavior

In this phase the data acquisition model is written in Matlab, starting by the highest level of abstraction where all the details referred to timing and performance are hidden. The purpose is to find out if it is possible to implement the filters and observe how the system behaves. The output matrix generated by Matlab will be stored as a golden reference.

2. Software code generation

The second phase covers the implementation of the system in C++. The Matlab coder is used to transform the Matlab functions into C++ executable functions which can be loaded with the same parameters as done in Matlab. Matlab coder [36] is a tool from MathWorks which enables the generation of synthesisable HDL code (both VHDL and Verilog) from MATLAB functions and Simulink models.

Final Remarks

The output matrix generated by the execution of the C++ functions is compared with the golden reference in order to verify that the software behaves as expected.

3. Hardware code generation

The third phase covers the design of the data acquisition system in terms of the different hardware blocks where the system will be implemented in. The C++ code is used by Vivado High Level Synthesis (HLS) to generate the VHDL code. Vivado HLS [36] is a tool which converts designs written in C, C++ or SystemC code into RTL design files (VHDL/Verilog or SystemC) for implementation in Xilinx devices.

The output parameters of the hardware implementation will be compared with the golden reference.

Finally the .bit file generated by Vivado HLS will be used to download the hardware code into the Zynq board. Software code is written in Xilinx Software Development Kit (SDK) to access the hardware and load the data to the functions generated by Vivado HLS. SDK [36] is a C/C++ code editor.

The output matrix from the Zynq board is compared with the golden reference to verify the results.

The first phase has been completed during this thesis project. The filters have been tested and they are ready to be implemented in the board on phases two and three which have not been implemented in this thesis.

It is described below two areas where future work has to be done:

- An evaluation should be done about the number of computations required for each filter to be implemented in hardware. Based on this evaluation and the results obtained in chapter 6 decisions have to be taken as to which filter to choose.
- An evaluation should be done analyze which components have to be implemented in hardware and which components have to be implemented in software in order to perform more efficiently.

7.5. Final Remarks

The goals of this thesis project have been successfully met as argued above. Decisions have been taken on which filter is the best option to be implemented in hardware based in the results obtained from the tests performed.

To conclude this thesis project, it is the hope for the future that phases two and three of the development process can be successfully implemented and the digital pulse detector can be correctly installed in the Physics Department at Aarhus University.

Bibliography

- [1] José María Górriz Artigot. Karhunen-loeve transform (klt) with soc design and high-level synthesis. *Department of Computing Engineering, The University of Aarhus, Denmark*, 2015. [cited at p. 72]
- [2] The Particle Physics Department at the Rutherford Appleton Laboratory. Silicon strip detectors in the atlas experiment, 2015. [Online; accessed 11-May-2015]. [cited at p. 2]
- [3] Salah Awadalla. *Solid-State Radiation Detectors: Technology and Applications*. CRC Press, 2015. [cited at p. 10]
- [4] M. Bogovac. Signal processing and electronics for nuclear spectrometry. *INTERNATIONAL ATOMIC ENERGY AGENCY, VIENNA*, 2007. [cited at p. 1, 5]
- [5] M. Bogovac. Digital pulse shaping. *Implementation of a Trapezoidal filter in a FPGA by using MATLAB and Xilinx design tools*, 2013. [cited at p. 11]
- [6] Soo Hyun Byun. Radioisotopes and radiation methodology. *Department of Medical Physics and Applied Radiation Sciences, McMaster University, Canada*, 2014. [cited at p. 31]
- [7] FAST ComTec. Nuclear pulse generators, 2014. [Online; accessed 5-April-2015]. [cited at p. 19]
- [8] Supriya Das. Pulse and pulse processing. *Centre for Astroparticle Physics and Space Science, Bose Institute, India*, 2011. [cited at p. 10]
- [9] G Ripamonti , A Geraci. Towards real-time digital pulse processing based on least mean squares algorithms. *Politecnico di Milano, Dipartimento di Elettronica e Informazione, Italy*, 1997. [cited at p. 10]
- [10] Robert Grzywacz. Digital signal processing from practitioner point of view. *University of Tennessee*, 2009. [cited at p. 10]
- [11] Cosimo Imperialea , Alessio Imperialeb. On nuclear spectrometry pulses digital shaping and processing. *Pacific Western University, Technological Division, Los Angeles, USA*, 2000. [cited at p. 11, 39]
- [12] DIMITRIS G. MANOLAKIS , VINAY K. INGLE. *Applied Digital Signal Processing*. Northeastern University, Boston, USA, 2011. [cited at p. 10]

Bibliography

- [13] JORDANOV V.T , KNOLL G. F , HUBER A.C , PANTAZIS J.A. *Digital techniques for real-time pulse shaping in radiation measurements*. Nucl. Instr. Meth., 1994. [cited at p. 71]
- [14] Neil E. Jacobsen. *NMR Spectroscopy Explained*. John Wiley and Sons, 2007. [cited at p. 11]
- [15] Glen F Knoll. *Radiation Detection and Measurement*. John Wiley and Sons Inc., 2000. [cited at p. 11, 27, 28, 32, 52]
- [16] Valentin T Jordanov, Glenn F Knoll. Digital synthesis of pulse shapes in real time for high resolution radiation spectroscopy. *Department of Nuclear Engineering, The University of Michigan*, 1994. [cited at p. 11, 36, 37]
- [17] Alexander Rasborg Knudsen. High level synthesis. *Reading course at Department of Engineering*, 2015. [cited at p. 1]
- [18] Charnes, A ; Frome, E. L.; Yu, P. L. The equivalence of generalized least squares and maximum likelihood estimates in the exponential family. *Journal of the American Statistical Association*, 1976. [cited at p. 46]
- [19] Matthew Mahoney. *DSP-Based Testing of Analog and Mixed-Signal Circuits*. Wiley-IEEE Computer Society Press, 1987. [cited at p. 6]
- [20] MATLAB. *version 8.3.0.532 (R2014a)*. The MathWorks Inc., Natick, Massachusetts, 2014. [cited at p. 45, 46]
- [21] M. Nakhostin. Recursive algorithms for real-time digital pulse shaping. *TRANSACTIONS ON NUCLEAR SCIENCE*, 2011. [cited at p. 10, 32]
- [22] P. V. NICHOLSON. *Nuclear Electronics*. John Wiley and Sons, 1974. [cited at p. 11, 71]
- [23] CAEN Costruzioni Apparecchiature Elettroniche Nucleari. Caen company profile, 2015. [Online; accessed 25-May-2015]. [cited at p. 35]
- [24] CAEN Costruzioni Apparecchiature Elettroniche Nucleari. Wp2081 digital pulse processing in nuclear physics, 2015. [Online; accessed 20-Juni-2015]. [cited at p. iii, 3, 23]
- [25] Alan V Oppenheim. *Discrete Time Signal Processing*. Prentice Hall, 1999. [cited at p. 27, 47]
- [26] Bernard Picinbono. *Principles of Signals and Systems: Deterministic Signals*. Artech House, 1988. [cited at p. 10, 48]
- [27] Vinay K. Ingle , John G. Proakis. *Digital Signal Processing Using MATLAB*. Northeastern University, Boston, USA, 2012. [cited at p. 10]
- [28] V. RADEKA. *Trapezoidal filtering of signals from large germanium detectors at high rates*. Nucl. Instr. Meth., 1972. [cited at p. 11, 71]
- [29] S.S. Kapoor , V.S. Ramamurthy. *Nuclear Radiation Detectors*. New Age International, 1986. [cited at p. 1, 11]
- [30] R. Redus. Digital pulse processors, theory of operation. *Chief Scientist Amptek Inc*, 2009. [cited at p. 10]
- [31] Angelo Rivetti. *CMOS: Front-End Electronics for Radiation Sensors*. CRC Press, 2015. [cited at p. 10]

Bibliography

- [32] M. Sampietro. A digital system for optimum resolution and x ray spectroscopy. *Rev. Sci. Instrum*, 1995. [cited at p. 10]
- [33] , P Singhai , P; Roy, A; Dhara. Digital pulse processing techniques for high resolution amplitude measurement of radiation detector. *Journal of Heritage Institute of Technology*, 2012. [cited at p. 4]
- [34] Helmuth Spieler. Pulse processing and analysis. *Physics Division Lawrence Berkeley National Laboratory, USA*, 2009. [cited at p. 10]
- [35] S D Stearns. *Digital Signal Analysis*. NJ Rochelle Park, 1975. [cited at p. 32]
- [36] Louise H. Crockett, Ross A. Elliot, Martin A. Enderwitz, Robert W. Stewart. *The Zynq Book*. Strathclyde Academic Media, 2014. [cited at p. 72, 73]
- [37] Milan STORK. Median filters and applications. *University of West Bohemia*, 2010. [cited at p. 27]
- [38] Terpconnect. Spectrum signals and noise, 2015. [Online; accessed 18-Juni-2015]. [cited at p. 26]
- [39] ZHANG Huaiqiang,GE Liangquan,TANG Bin,LIU Tingli. Optimal choice of trapezoidal shaping parameters in digital nuclear spectrometer system. *Engineering Research Center of Nuclear Technology Application*, 2013. [cited at p. 35]
- [40] wavemetrics. Igor pro 6, 2011. [Online; accessed 17-April-2015]. [cited at p. 46]
- [41] Wikipedia. Exponential smoothing, 2004. [Online; accessed 23-April-2015]. [cited at p. 13, 14, 15, 17, 23, 27, 45, 46]
- [42] Steven W.Smith. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Newnes, 2003. [cited at p. 10]
- [43] Bo Yu. Nuclear science symposium and medical imaging conference record. *Anaheim, California, USA*, 2012. [cited at p. 10]

Appendix **A**

Development and System Specification

A.1. System Specification

The system specification is used to make an exact set of requirements based on the needs of the system to be developed.

A.1.1 Requirements

The requirements are divided into functional, non-functional, behavioral and performance requirements. For the digital pulse detector only the functional requirements have been considered and they are illustrated in SysML drawings and not described in text form.

The requirements are identified by a label CR-xxx where xxx represents a number. Each requirement consists of several sub-requirements; these requirements are labeled as follows:

FR - XXX.Z: Functional requirement

Where XXX makes reference to the requirement number and Z is the unique ID for the sub requirement. Please note, there can be several derived requirements so the ID number of a requirement may have several digits, for example FR-001.1.2.1.

A.1.1.1 Functional Requirements

Define a function of a system and its components. A function is described as a set of inputs, the behavior and outputs.

Appendix A. Development and System Specification

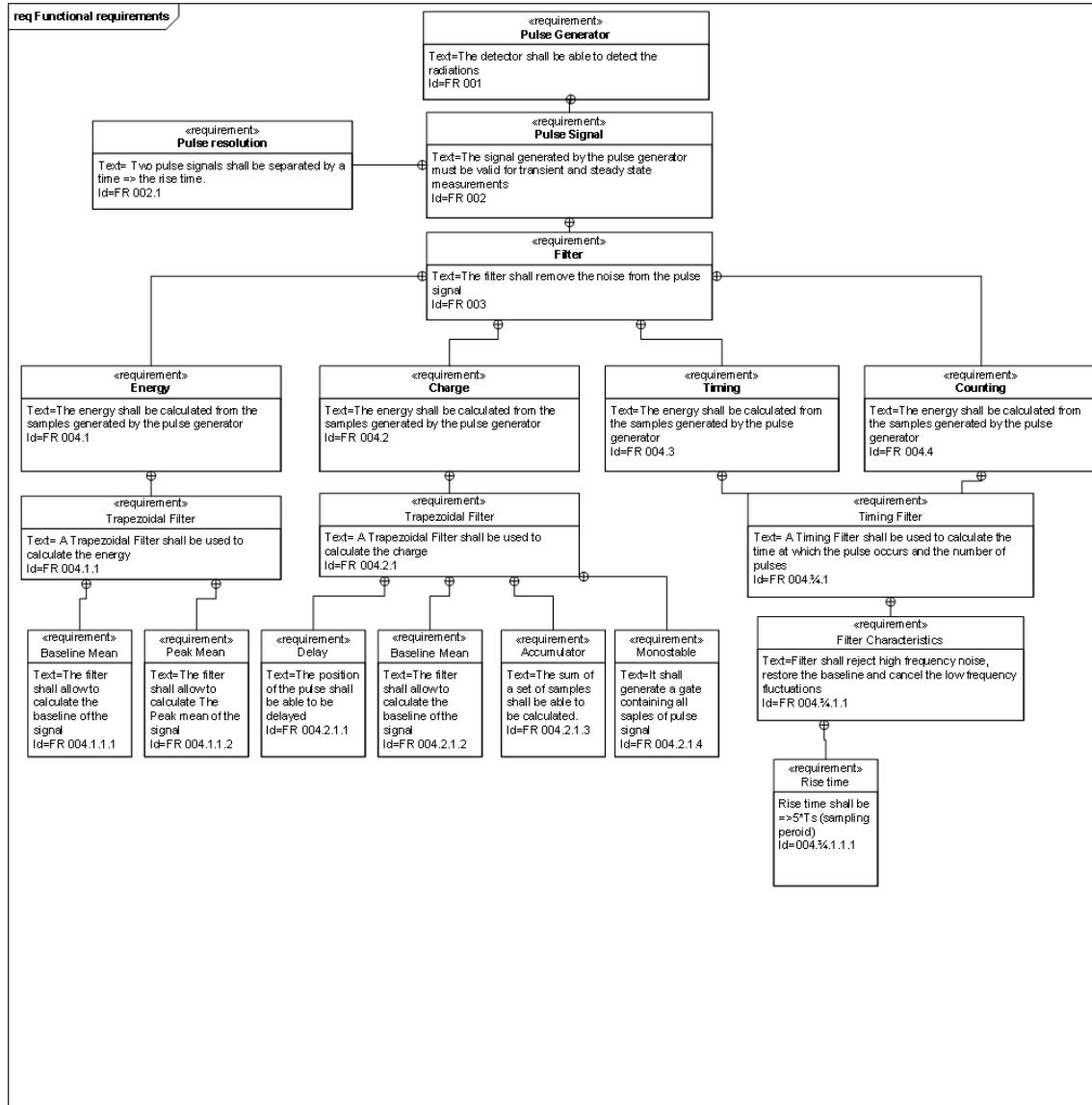


Figure A.1: Functional Requirements Diagram

A.2. Design Constraints Analysis

By making the Design Constraint Analysis, it is easier to ensure that the requirements are met within certain constraints, which ensure an overall reliability and quality of the product.

The constraints defined here, are boundaries which limit the conditions the product can be designed from. These can range anything from technology, materials or time taken to develop the system.

Constraints can be seen as non-functional requirements, with the difference that requirements have to be verified while constraints don't need verification.

1. The overall performance of the developed system is very important, since it needs to interface with the other components, thus meeting strict requirements.

Design Constraints Analysis

Issue	Critical	Very Important	Important	Less Important	Notes
Performance		X			1
Usage			X		2
Reliability			X		3
Easy serviceable		X			4
R&D cost effective				X	5
Production cost effective				X	6

Figure A.2: Design Constraints Table

2. The product should be easy to use and information to user has to be easily accessible, in accordance with the requirements.
3. The reliability of the system is important, however since the system might not be necessarily brought to a final physical state during the process of this project, the inner working and the analysis of the product contribute more in the project.
4. Since the product is mostly software and hardware based, it is very important that the code is commented properly, thus it is very important.
5. Tools provided by Physics department. It is intended to be used at Physics department.
6. Components provided by Physics department. Not intended to go into mass production

Appendix B

Parameters of the Pulse

This appendix contains a description of the pulse parameters used during the development of the thesis.

- Rise time (PRT): it is time taken for the signal leading edge to rise from 10 to 90 % of the maximum value. In the radiation detection, it can be considered the time that the charge generated by the radiation interaction in the material takes to travel through the detector material and induce the charge on the contact electrodes. Thus, the rise-time provides information about how is the interaction within a nuclear detector.
- Pulse Fall Time (PFT): it is the interval between 10 % and 90 % of the amplitude on the trailing edge.
- Shaping time: it is the value at which the pulse reaches its maximum.
- Peaking time: it is the time the signal takes to raise from zero the maximum value.
- Pulse Amplitude (PA): it is the difference between the maximum and the minimum value of the pulse.
- Pulse Width (PW): it is the interval between leading and trailing edge medians
- Pulse Integral (PI):it is the sum of the differences between the value of pulse and the minimum value of pulse during the sampling time

Figure B.1 shows a pulse with some of the parameters mentioned above.

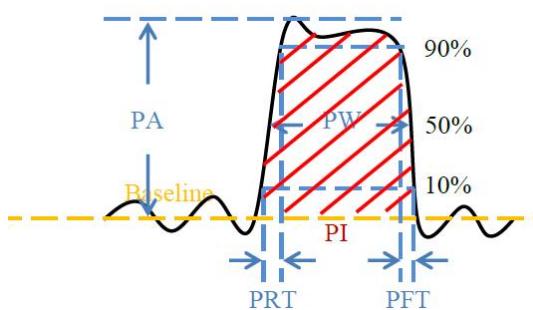


Figure B.1: Parameters of a pulse

Appendix C

Accuracy of the Implemented Filters

C.1. $CR - (RC)^n$ Filters

The tables bellow show the results of the tests performed to different pulse signals which were provided by Kim Bjerge. The rise time and decay time of those signals can be found in appendix E and they are displayed at the bottom of each table.

The tables contain the filter order together with the time offset in μs of P1 corresponding to the crossing with the upper threshold and the offset of the filter in μs of P2 corresponding to the crossing with the lower threshold.

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.628	3.625
Second order	3.580	4.528
Third order	3.452	5.523
Forth order	2.985	6.324

Table C.1: Table with test results from $CR - (RC)^n$ filters with $adc1_{10}$ bi-exponential pulse

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.257	3.800
Second order	3.250	4.625
Third order	3.728	5.625
Forth order	2.830	6.457

Table C.2: Table with test results from $CR - (RC)^n$ filters with $PE1$ bi-exponential pulse

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.405	3.300
Second order	3.630	4.640
Third order	3.658	5.500
Forth order	2.725	6.512

Table C.3: Table with test results from $CR - (RC)^n$ filters with $PE6$ bi-exponential pulse

Appendix C. Accuracy of the Implemented Filters

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.552	3.435
Second order	3.756	4.765
Third order	3.875	5.675
Forth order	2.879	6.123

Table C.4: Table with test results from $CR - (RC)^n$ filters with PE15 bi-exponential pulse

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.455	3.612
Second order	3.480	4.415
Third order	3.324	5.518
Forth order	2.995	6.113

Table C.5: Table with test results from $CR - (RC)^n$ filters with PE21 bi-exponential pulse

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.319	3.712
Second order	3.217	4.625
Third order	3.276	5.413
Forth order	2.870	6.213

Table C.6: Table with test results from $CR - (RC)^n$ filters with PE26 bi-exponential pulse

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.528	3.560
Second order	3.456	4.487
Third order	3.376	5.765
Forth order	2.970	6.456

Table C.7: Table with test results from $CR - (RC)^n$ filters with PE31 bi-exponential pulse

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.426	3.625
Second order	3.675	4.453
Third order	3.564	5.512
Forth order	2.985	6.245

Table C.8: Table with test results from $CR - (RC)^n$ filters with PE41 bi-exponential pulse

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.768	3.567
Second order	3.654	4.456
Third order	3.534	5.534
Forth order	3.004	6.122

Table C.9: Table with test results from $CR - (RC)^n$ filters with PE50 bi-exponential pulse

Trapezoidal Filter

Filter Order	P1 Time Offset (μs)	P2 Time Offset (μs)
First order	3.458	3.555
Second order	3.380	4.123
Third order	3.225	5.101
Forth order	2.889	6.005

Table C.10: Table with test results from $CR - (RC)^n$ filters with PE71 bi-exponential pulse

C.2. Trapezoidal Filter

The tables bellow show the results of the tests performed with the implemented trapezoidal filters. The second column of the table contains the energy measured by each implementation of the filter and the third column contains the accuracy of those values based on the energy calculated from the input signal. The signals used in the measurements are the same signals used with the $CR(RC)^n$ filters and they are also displayed at the bottom of the tables.

The energy of the pulse is: 216.2369 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	277.74	71.55
Trapezoidal filter TF single	271.72	74.34
Trapezoidal filter TF biexponential	297.54	62.4
Trapezoidal filter TF biexponential model based	214.05	99.1

Table C.11: Table with accuracy percentage of trapezoidal filter with $adc1_{10}$ bi-exponential pulse

The energy of the pulse is: 210.0866 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	225.75	92.54
Trapezoidal filter TF single	220.18	95.19
Trapezoidal filter TF biexponential	199.21	94.82
Trapezoidal filter TF biexponential model based	205.03	97.6

Table C.12: Table with accuracy percentage of trapezoidal filter with PE1 bi-exponential pulse

The energy of the pulse is: 477.8932 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	556.34	83.58
Trapezoidal filter TF single	532.13	88.64
Trapezoidal filter TF biexponential	375.6	78.59
Trapezoidal filter TF biexponential model based	471.19	98.6

Table C.13: Table with accuracy percentage of trapezoidal filter with PE6 bi-exponential pulse

The energy of the pulse is: 813.7465 Joules

Appendix C. Accuracy of the Implemented Filters

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	950.23	83.22
Trapezoidal filter TF single	941.39	84.31
Trapezoidal filter TF biexponential	657.31	80.77
Trapezoidal filter TF biexponential model based	799.09	98.2

Table C.14: Table with accuracy percentage of trapezoidal filter with *PE15* bi-exponential pulse

The energy of the pulse is: 178.1797 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	206.63	84.03
Trapezoidal filter TF single	204.25	85.36
Trapezoidal filter TF biexponential	144.78	81.25
Trapezoidal filter TF biexponential model based	176.21	98.9

Table C.15: Table with accuracy percentage of trapezoidal filter with *PE21* bi-exponential pulse

The energy of the pulse is: 469.0825 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	552.46	82.22
Trapezoidal filter TF single	530.74	86.85
Trapezoidal filter TF biexponential	374.68	79.87
Trapezoidal filter TF biexponential model based	454.06	96.8

Table C.16: Table with accuracy percentage of trapezoidal filter with *PE26* bi-exponential pulse

The energy of the pulse is: 801.3338 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	949.25	81.54
Trapezoidal filter TF single	940.19	82.67
Trapezoidal filter TF biexponential	656.55	81.93
Trapezoidal filter TF biexponential model based	780.49	97.4

Table C.17: Table with accuracy percentage of trapezoidal filter with *PE31* bi-exponential pulse

The energy of the pulse is: 164.7015 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	195.38	81.36
Trapezoidal filter TF single	169.99	96.78
Trapezoidal filter TF biexponential	246.58	50.28
Trapezoidal filter TF biexponential model based	163.38	99.2

Table C.18: Table with accuracy percentage of trapezoidal filter with *PE41* bi-exponential pulse

The energy of the pulse is: 402.1277 Joules

Trapezoidal Filter

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	504.84	74.45
Trapezoidal filter TF single	479.87	80.66
Trapezoidal filter TF biexponential	336.96	83.79
Trapezoidal filter TF biexponential model based	393.87	97.7

Table C.19: Table with accuracy percentage of trapezoidal filter with $PE50$ bi-exponential pulse

The energy of the pulse is: 661.3635 Joules

Filter Type	Energy (Joules)	Accuracy in %
Trapezoidal filter recursive	834.95	73.75
Trapezoidal filter TF single	774.65	82.87
Trapezoidal filter TF biexponential	874.46	67.77
Trapezoidal filter TF biexponential model based	650.77	98.4

Table C.20: Table with accuracy percentage of trapezoidal filter with $PE71$ bi-exponential pulse

Appendix **D**

Approach in the Implementation of the Transfer Function of the Trapezoidal Filter

The appendix shows a new approach to reduce the transfer function implemented with single exponential pulse from section 5.2.2.1.

The quantized output series of data $y(t)$ from figure 5.3 can be represented in Z-domain:

$$H(z) = \frac{A}{n_a} \frac{z}{(z-1)^2 z^{n_a+n_b}} (1 - z^{n_a} - z^{n_b} + z^{(n_a-n_b)}) \quad (\text{D.1})$$

The single decay input signal $x(t)$ from equation 2.10 is defined by:

$$x(t) = A e^{(\frac{-t}{\tau_d})} \quad (\text{D.2})$$

Taking the z-transform of the input $x(t)$ it is obtained:

$$X(z) = A \frac{z}{z - e^{(\frac{-Ts}{\tau_d})}} \quad (\text{D.3})$$

where Ts represents the sample period. Ts is calculated in equation 6.1.

Using those equations the transfer function can be written:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{A}{n_a} \frac{z}{(z-1)^2 (z^{n_a+n_b})} (1 - z^{n_a} - z^{n_b} + z^{(n_a+n_b)})}{A \frac{z}{z - e^{(\frac{-Ts}{\tau_d})}}} \quad (\text{D.4})$$

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-Ts}{\tau_d})})(1 - z^{n_a} - z^{n_b} + z^{(n_a+n_b)})}{(z-1)^2 (z^{n_a+n_b})} \quad (\text{D.5})$$

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-Ts}{\tau_d})})(z^{(n_a+n_b)})(1 - z^{n_a} - z^{n_b} + z^{(-n_a-n_b)})}{(z-1)^2 (z^{n_a+n_b})} \quad (\text{D.6})$$

This can be factorized in terms of z

$$H(z) = \frac{1}{n_a} \frac{(z - e^{(\frac{-Ts}{\tau_d})})(z^{(n_a+n_b)})(1 - z^{n_a})(1 - z^{n_b})}{(z-1)^2 (z^{n_a+n_b})} \quad (\text{D.7})$$

If dividing numerator and denominator by $(z^{(n_a+n_b)}) * z^2$

Appendix D. Approach in the Implementation of the Transfer Function of the Trapezoidal Filter

$$H(z) = \frac{z^{-1}}{n_a} \frac{(1 - e^{(\frac{-Ts}{\tau_d})} z^{-1})(1 - z^{-n_a})(1 - z^{-n_b})}{(1 - z^{-1})^2} \quad (\text{D.8})$$

It can be observed that equation D.8 is equal to equation 5.22.

Appendix E

Pulses generated with Pulse Generator BNC

The tables show all the pulses which have been provided by kim Bjerge from Aarhus University. Some of the pulses were discarded because there was missing a part of the pulse and not the entire pulse was present in the txt file or because there were pulses with two or more peaks. All pulses except the last five have been generated in repetitive mode by the random tail pulse generator BNC model BD-2.

The first column has the number of the folder where the signals are saved. Each folder contains five signals of the same type. The second and the third column contain the rise time and the fall time in microseconds. Finally the last columns shows the name of .txt file saved in the folder of the first column.

Appendix E. Pulses generated with Pulse Generator BNC

Rise Time (in microseconds)	Fall Time (in microseconds)	Text File Name
1	0.1	5 '1.R01F5/PE1.txt'
1	0.1	5 '1.R01F5/PE2.txt'
1	0.1	5 '1.R01F5/PE3.txt'
1	0.1	5 '1.R01F5/PE4.txt'
1	0.1	5 '1.R01F5/PE5.txt'
1	0.1	5 'adc1 ₁₂ .txt'
2	0.1	10 '2.R01F10/PE6.txt'
2	0.1	10 '2.R01F10/PE7.txt'
2	0.1	10 '2.R01F10/PE9.txt'
2	0.1	10 '2.R01F10/PE10.txt'
3	0.1	20 '3.R01F20/PE11.txt'
3	0.1	20 '3.R01F20/PE12.txt'
3	0.1	20 '3.R01F20/PE13.txt'
3	0.1	20 '3.R01F20/PE14.txt'
3	0.1	20 '3.R01F20/PE15.txt'
4	0.1	50 '4.R01F50/PE20.txt'
4	0.1	50 'adc1 ₁₃ .txt'
5	0.2	5 '5.R02F5/PE21.txt'
5	0.2	5 '5.R02F5/PE22.txt'
5	0.2	5 '5.R02F5/PE23.txt'
5	0.2	5 '5.R02F5/PE24.txt'
5	0.2	5 '5.R02F5/PE25.txt'
6	0.2	10 '6.R02F10/PE26.txt'
6	0.2	10 '6.R02F10/PE27.txt'
6	0.2	10 '6.R02F10/PE28.txt'
6	0.2	10 '6.R02F10/PE29.txt'
6	0.2	10 '6.R02F10/PE30.txt'
7	0.2	20 '7.R02F20/PE31.txt'
7	0.2	20 '7.R02F20/PE32.txt'
7	0.2	20 '7.R02F20/PE34.txt'
7	0.2	20 '7.R02F20/PE35.txt'
7	0.2	100 'adc1 ₁₁ .txt'
9	0.5	5 '9.R05F5/PE41.txt'
9	0.5	5 '9.R05F5/PE42.txt'
9	0.5	5 '9.R05F5/PE43.txt'
9	0.5	5 '9.R05F5/PE45.txt'

Rise Time (in microseconds)	Fall Time (in microseconds)	Text File Name
10	0.5	'10.R05F10/PE47.txt'
10	0.5	'10.R05F10/PE48.txt'
10	0.5	'10.R05F10/PE50.txt'
11	0.5	'11.R05F20/PE51.txt'
11	0.5	'11.R05F20/PE53.txt'
11	0.5	'11.R05F20/PE54.txt'
12	0.5	'12.R05F50/PE60.txt'
13	1	'13.R1F5/PE61.txt'
13	1	'13.R1F5/PE64.txt'
14	1	'14.R1F10/PE66.txt'
14	1	'14.R1F10/PE67.txt'
14	1	'14.R1F10/PE68.txt'
14	1	'14.R1F10/PE69.txt'
14	1	'14.R1F10/PE70.txt'
15	1	'15.R1F20/PE71.txt'
15	1	'15.R1F20/PE74.txt'
15	1	'15.R1F20/PE75.txt'
16	1	'16.R1F50/PE79.txt'
17	2	'adc1 ₁₄ .txt'
17	2	'adc1 ₁₅ .txt'
17	2	'adc1 ₁₆ .txt'
17	2	'adc1 ₁₇ .txt'
17	5	'17.R5F5/PE82.txt'
17	5	'17.R5F5/PE83.txt'
17	5	'17.R5F5/PE84.txt'
17	5	'17.R5F5/PE85.txt'
18	5	'18.R5F10/PE86.txt'
18	5	'18.R5F10/PE90.txt'
18	5	'adc1 ₁₀ .txt'
19	5	'19.R5F20/PE90.txt'
19	5	'19.R5F20/PE91.txt'
19	5	'19.R5F20/PE92.txt'
19	5	'19.R5F20/PE94.txt'
20	5	'20.R5F50/PE99.txt'
20	5	'20.R5F50/PE100.txt'
21	5	'21.RandomR05F50/PE101.txt'
21	5	'21.RandomR05F50/PE102.txt'
21	5	'21.RandomR05F50/PE103.txt'
21	5	'21.RandomR05F50/PE104.txt'
21	5	'21.RandomR05F50/PE105.txt'

