

Università degli Studi di Firenze



Scuola di Scienze Matematiche Fisiche e Naturali

Corso di Laurea in Fisica e Astrofisica

**Real-time extraction of energy and shape  
parameters of digitized signals from  
nuclear radiation detectors**

**Estrazione in real-time di energia e parametri  
di forma di segnali digitalizzati di rivelatori  
per fisica nucleare**

SUPERVISOR

Dott. Gabriele Pasquali

CANDIDATE

Pietro Ottanelli

Academic Year 2014/2015



# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>1</b>  |
| 1.1      | Physical context . . . . .  | 2         |
| 1.2      | Particle identification . . . . .                                   | 3         |
| 1.2.1    | Pulse Shape Analysis . . . . .                                      | 5         |
| 1.3      | Outline of the Thesis . . . . .                                     | 9         |
| <b>2</b> | <b>Energy Measurement</b>   | <b>11</b> |
| 2.1      | Obtaining the Energy information from a signal . . . . .            | 13        |
| 2.1.1    | Pole-Zero correction . . . . .                                      | 14        |
| 2.2      | The digital trapezoidal shaper with PZC . . . . .                   | 16        |
| 2.2.1    | Implementation of the trapezoidal filter . . . . .                  | 19        |
| 2.2.2    | Getting the Energy . . . . .  | 23        |
| <b>3</b> | <b>The Trigger system</b>   | <b>25</b> |
| 3.1      | Bipolar signal generation . . . . .                                 | 25        |
| 3.1.1    | Advantages of using a bipolar shaper . . . . .                      | 26        |
| 3.2      | Zero crossing detection . . . . .                                   | 29        |
| 3.3      | The trigger box . . . . .   | 32        |
| <b>4</b> | <b>Interpolation of digitized detector signals</b>                  | <b>35</b> |
| 4.1      | Interpolation and digital sampling . . . . .                        | 35        |
| 4.2      | Kernel spaces, kernel functions and interpolation process . . . . . | 36        |
| 4.3      | Choosing the right kernel . . . . .                                 | 38        |
| 4.4      | B-Spline kernels . . . . .  | 40        |
| 4.4.1    | Cubic B-Spline kernel . . . . .                                     | 41        |
| 4.4.2    | Approximation of the inverse B-Spline filter . . . . .              | 43        |
| 4.4.3    | Evaluation of the approximation error . . . . .                     | 44        |
| 4.4.4    | B-Splines interpolation: a graphical description . . . . .          | 46        |
| 4.5      | B-Spline interpolation and LTI filters . . . . .                    | 49        |
| 4.5.1    | Obtaining the derivative of the signal . . . . .                    | 49        |
| 4.5.2    | Correcting for the preamplifier decay time . . . . .                | 51        |
| 4.6      | Evaluating the maximum of the current signal . . . . .              | 54        |
| 4.7      | Dealing with noise and other problems . . . . .                     | 55        |
| 4.7.1    | Maximum detection . . . . .   | 55        |

|          |  |            |
|----------|--|------------|
| 4.7.2    | Signal reconstruction: the B-Spline problem . . . . .          | 57         |
| 4.7.3    | The effects of noise on the detection of the maximum . . . . . | 59         |
| <b>5</b> | <b>Experimental Setup</b>                                      | <b>63</b>  |
| 5.1      | The Si-Si-CsI Telescope . . . . .                              | 63         |
| 5.1.1    | The PACI preamplifiers . . . . .                               | 65         |
| 5.2      | Digitizing boards and firmware . . . . .                       | 67         |
| 5.2.1    | The DSP . . . . .  | 68         |
| 5.2.2    | The FPGA . . . . .   | 69         |
| 5.3      | General overview of the firmware . . . . .                     | 70         |
| 5.4      | Implementation of the interpolating filter . . . . .           | 72         |
| 5.4.1    | Inverse B-Spline filter . . . . .                              | 72         |
| 5.4.2    | Reconstruction filter . . . . .                                | 75         |
| 5.4.3    | Overall maximum detection . . . . .                            | 77         |
| 5.4.4    | Introduction of the moving average . . . . .                   | 78         |
| 5.5      | Acquisition and trigger logic . . . . .                        | 79         |
| 5.6      | Energy measurement: trapezoidal shapers settings . . . . .     | 80         |
| <b>6</b> | <b>Data Analysis and related remarks</b>                       | <b>81</b>  |
| 6.1      | Offline data analysis . . . . .                                | 82         |
| 6.2      | Results of the measure at LNS . . . . .                        | 82         |
| 6.2.1    | Comparison of online and offline data . . . . .                | 83         |
| 6.2.2    | $\Delta E$ -E matrix and energy calibration . . . . .          | 84         |
| 6.2.3    | $\Delta E$ -E plots . . . . .                                  | 88         |
| 6.2.4    | Energy-Imax PSA plots . . . . .                                | 88         |
| 6.3      | Simulation of the process . . . . .                            | 99         |
|          | <b>Concluding remarks and future developments</b>              | <b>105</b> |
|          | Firmware modifications . . . . .                               | 106        |
|          | <b>Greetings</b>   | <b>107</b> |



# Chapter 1

## Introduction

The work presented in this thesis is part of the research activities of the Experimental Nuclear Physics group, at INFN-Florence. For more than a decade, the group has studied the digitization and digital signal processing of signals from nuclear radiation detectors. Ten years ago a single-channel digitizing board was developed, equipped with a Digital Signal Processor (DSP) and a 12-bit ADC [1]. About 300 of these boards are currently used by the GARFIELD apparatus [2] at INFN Laboratori Nazionali di Legnaro (LNL). Recently, a new version of the board has been produced. The new digitizing board has two channels, each equipped with a 14-bit ADC. An important improvement is the replacement of the First In First Out (FIFO) memory, used in the previous boards to store the signal, with a powerful FPGA. At the beginning of this work a preliminary version of the firmware had been already implemented. It contained a FIFO and a simple acquisition logic, thus mimicking the physical components present in the previous boards. The first part of the work itself consisted in introducing a set of new features, namely:

- implementation of a trapezoidal shaper for energy measurements;
- implementation of an internal trigger system;
- creation of a state machine to directly access the EPCS (E2Prom memory device) to write the FPGA firmware without using the JTAG cable;
- implementation of a trigger management logic which allows for using complex trigger systems;
- use of a PLL on the FPGA in order to dynamically change the relative phase shift between the ADC clock and the FPGA clock (to avoid errors when reading samples from the ADC).

All these additions are of general purpose for a digitizer employed in nuclear physics (regardless of the detector to which it is connected). Some of them are briefly described in this thesis. However, the main addition, and the core argument of the current work, is the development of a set of linear filters devoted to real-time interpolation of the ADC samples. The process can be performed exactly by using an Infinite Impulse Response

(IIR) linear filter<sup>1</sup> [3, 4], but this makes it impossible to implement a real-time calculation. Our purpose was then to implement the interpolation via a Finite Impulse Response (FIR) filter without losing too much precision.

During the preliminar theoretical study the idea came out of using the output of the filter (the interpolating function coefficients) to apply additional processing to the reconstructed signal **in the continuous time domain**; namely we wanted to exploit this process in order to obtain an upsampled version of the detector current signal from the sampled charge signal. The current signal could then be used for pulse-shape identification of nuclear fragments (see sec. 6.2.4).

Two advantages of this approach are the reduction of the dead time, obtained by implementing in real-time some of the features that were previously extracted by the DSP each time an event occurred, and the reduction in data transfer bandwidth. The latter follows from the on-board extraction of information which should otherwise be extracted offline working on a portion of the sampled signal.

Apart from designing the firmware, we prepared and performed an experiment at Laboratori Nazionali del Sud (LNS) of INFN in Catania. The purpose of the experiment was to test the digitizers and their firmware in “real life” experimental conditions. The digitizing boards have been used for digitizing and processing signals from a silicon-silicon-cesium iodide  $\Delta E$ -E telescope of the FAZIA collaboration [5]. The processing algorithms designed in this work have been used for energy measurement and pulse shape identification of the nuclear fragments impinging on the telescope, also performing some “tuning” in order to improve their performance. Details about the experimental setup and the obtained results will be the subject of chapter 6.

## 1.1 Physical context

In the last few decades, nuclear physics has devoted many efforts to study the properties of nuclear matter. In particular, there is a great interest in those terms of the nuclear Equation of State (nEoS) that are linked to the isospin degree of freedom ( $N/Z$ ), where  $N$  and  $Z$  are respectively the number of neutrons and the number of protons in the nucleus. In more recent times, the possibility of performing reactions using beams of unstable nuclei (commonly called exotic nuclei) has opened a new physical window to study nuclear matter properties far from the so-called *valley of stability* in the nuclide chart. A detailed analysis of the fragments produced in nuclear reactions at energies around the Fermi energy (10 – 100 AMeV) is a powerful instrument to study nuclear matter properties. In fact, the distribution of the produced fragments (isotopic, angular, energetic) is strictly related to the nEoS. For this reason, the identification in charge and mass of the detected fragments has become a crucial aspect of nuclear physics. In the last years, the identification performance of the experimental apparatuses has been greatly increased, mainly because of:

- advances in detector technologies;

---

<sup>1</sup>More details are given in sec. 2.2

- advances in electronics;
- study of new identification techniques.

Two of the most used identification techniques are described in detail in sec. 1.2.

## 1.2 Particle identification

In the last decades, many different methods have been developed to identify nuclear fragments. The simplest method is probably the  $\Delta E - E$  correlation. Every charged particle interacts with matter, transferring its kinetical energy to the surrounding medium in different ways (e.g. electromagnetic interactions, with electrons and nuclei, and nuclear collisions). If the main contribution to the energy transfer comes from the interaction with the atomic electrons (as is the case in our incident energy regime), the specific energy loss  $\frac{dE}{dx}$  is given by the Bethe-Bloch formula:

$$-\frac{dE}{dx} = \frac{4\pi e^4 Z^2}{m_e v^2} N B \quad (1.1)$$

$$B = Z_{abs} \left[ \ln \frac{2m_e v^2}{I} - \ln \left( 1 - \frac{v^2}{c^2} \right) - \frac{v^2}{c^2} \right]$$

where  $Ze$  is the charge carried by the particle,  $v$  is its speed,  $m_e$  the rest mass of the electron,  $N$  the numeric density of atoms in the absorbing medium,  $Z_{abs}$  the atomic number of the absorber and  $I$  its average excitation/ionization potential [6]. Due to the logarithmic term,  $B$  is approximately independent on  $v$  for non-relativistic particles, and therefore dependent only on the absorber medium characteristics. In this approximation, we have for a nuclear fragment

$$B = K Z_{abs} \quad \text{with } K \text{ constant}$$

$$E = \frac{1}{2} m v^2 = \frac{1}{2} A m_u v^2$$

$$-\frac{dE}{dx} = \frac{4\pi e^4 Z^2}{m_e v^2} N K Z_{abs} = 2\pi e^4 \frac{m_u}{m_e} N K Z_{abs} \frac{A Z^2}{E} \quad (1.2)$$

where  $m_u$  is the atomic mass unit and  $A$  the atomic number of the nuclear fragment. With this expression it is possible to evaluate the energy lost by the particle along the track as a function of the penetration depth, whose representation is called Bragg curve (figure 1.1). The exact shape of the Bragg curve,  $f(x)$ , depends on the impinging particle properties  $(Z, A, E_0)$  as well as on the absorber properties  $(N, Z_{abs}, K)$ :

$$f(x) = f(x; Z, A, E_0, N, Z_{abs}, K) \quad (1.3)$$

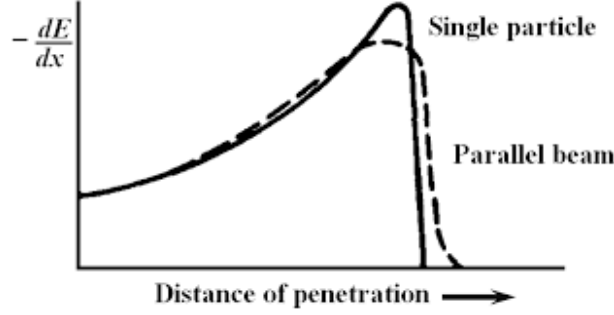


Figure 1.1: The Bragg curve for a single particle (solid line) and its average shape for a parallel beam of particles (dashed). Near the end of the path, the energy loss presents a peak (Bragg peak), since the effective charge of the impinging particle becomes  $< Z$  and finally goes to zero as the particles slows down. The differences between the two curves are due to the energy straggling process.

Suppose to have two layers of absorber (e.g. two silicon detectors) and a particle passing through the first one and being stopped in the second, the particle loses part of its energy in the first detector ( $\Delta E$ ), and the remainder in the other ( $E_{res}$ ). Such a configuration is called  $\Delta E - E$  telescope. If the two detectors have thickness  $t_1$  and  $t_2$ , then we have:

$$\begin{aligned}\Delta E &= \int_0^{t_1} f(x; Z, A, E_0, N_1, Z_{abs1}, K_1) dx \\ E_{res} &= \int_{t_1}^{t_1+t_2} f(x; Z, A, E_0, N_2, Z_{abs2}, K_2) dx\end{aligned}\quad (1.4)$$

where we used the indexes 1 and 2 to identify the two detectors. The condition that the particle is stopped in the second detector is expressed by  $\Delta E + E_{res} = E_0$ . Assuming that the energy loss in the first detector is small with respect to  $E_0$ , we obtain that  $\left| \frac{dE}{dx} \right|$  is constant in the first detector, so that

$$\Delta E = \left| \frac{dE}{dx} \right| t_1 \quad (1.5)$$

Assuming  $E(x) = E_0$  in the first detector we have that:

$$f(x; Z, A, E_0, N_1, Z_{abs1}, K_1) = 2\pi e^4 \frac{m_u}{m_e} N_1 K_1 Z_{abs1} \frac{AZ^2}{E_0} = \alpha_1 \frac{AZ^2}{E_0} \quad (1.6)$$

By plotting  $\Delta E$  vs  $E_0$ , it is then expected that points associated to a given isotope form an hyperbola on the correlation matrix. Each hyperbola has equation

$$y = \alpha_1 \frac{AZ^2}{x} \quad (1.7)$$

and so the various isotopes are separated. Also plotting  $\Delta E$  vs  $E_{res}$  produces different

curves for different isotopes. In this case we have:

$$\begin{aligned}\Delta E &= \alpha_1 \frac{AZ^2}{E_0} \\ E_{res} &= E_0 - \Delta E\end{aligned}\tag{1.8}$$

By extracting  $E_0$  from the first equation and substituting it in the second we obtain that points associated to a given isotope form an hyperbola on the correlation matrix with equation:

$$x = \alpha_1 \frac{AZ^2}{y} - y\tag{1.9}$$

An example of  $\Delta E - E_{res}$  correlation matrix is shown in fig. 1.2, where the  $\Delta E$  and  $E_{res}$  energy are measured by the silicon detectors of the telescope employed in our test at LNS.

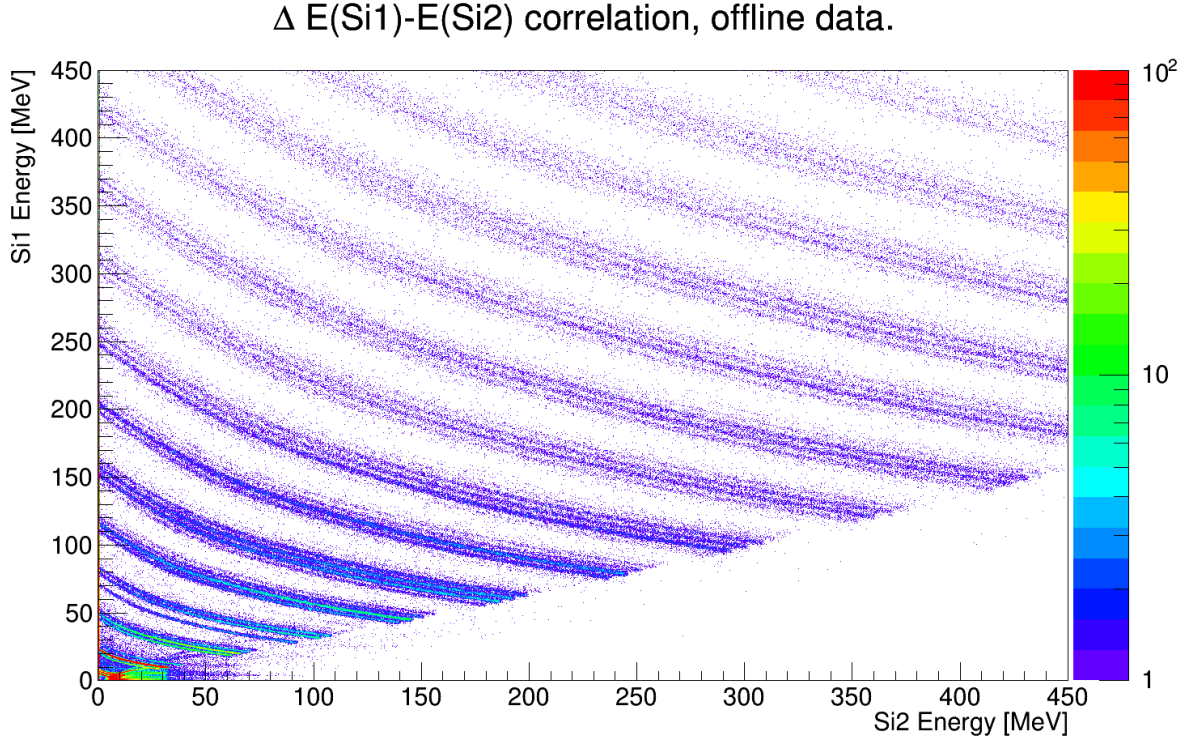


Figure 1.2: An example of  $\Delta E - E$  correlation, obtained with the data measured at LNS

### 1.2.1 Pulse Shape Analysis

$\Delta E - E$  identification has also some drawbacks. A minor problem is the need of two or more detectors with the associated electronic read-out chains. A major limitation is that in order for the particle to generate both informations ( $\Delta E$  and  $E$ ), the initial energy of the particle must be high enough for it to punch through the first detector; this

introduces an identification threshold in energy. The energy threshold could be lowered by using a thinner detector. However, a thinner detector would increase the effect on  $\Delta E - E$  resolution of the energy straggling [6] and the non uniformity of the detector thickness. Moreover, a thinner detector will have a bigger capacitance, thus increasing the noise contribution from the charge preamplifier [7].

In recent times, more advanced methods (generally known as Pulse Shape Analysis (PSA) methods) have been proposed. These methods have their ratio in the mechanism of radiation-matter interaction and of the charge/light collection. Like the name suggests, PSA methods use the analysis of the signal shape as a mean to identify particles, so they rely on a single detector; this partially solves the problem of the threshold<sup>2</sup>. In general, particle identification is performed by plotting matrices representing the correlation of two parameters ( $\Delta E$  and  $E$  in case of the  $\Delta E - E$  technique) and then identifying the particles using these plots. For PSA methods, the two variables are chosen as follows:

- one is an ordering parameter (e.g. the total charge);
- the other must be a shape-dependent variable (e.g. charge signal rise time).

When dealing with silicon detectors (like in this thesis) the two most used shape parameters are the rise-time of the charge signal and the maximum of the current signal ( $I_{MAX}$ ). Both depend on the so called *plasma time* and on the charge collection process [8]. Since the *plasma time* is associated, especially for light fragments, to the end of the particle's track, the particle must be stopped into the detector. By looking at fig. 1.1 and remembering that the number of electron-hole pairs produced by an impinging particle is proportional to its energy loss, we conclude that a particle produces a greater number of electron-hole pairs per unit length at the end of its path. This dense "plasma" of electrons and holes acts like a screen for the electric field, so at first only the external portion of it is affected by the external electric field and starts drifting towards the electrodes. The process continues until all charges have started drifting. The time needed to put all charges into motion is called *plasma time*. The *plasma time* depends on the dimensions of the cloud and on its density, and both these parameters are dependent on the characteristics of the impinging particle. The *plasma time* is of particular importance for tracks produced by fragments with  $Z \geq 3$ , when the density of electron-hole pairs is high enough to produce a plasma of electrons and holes all along the track.

Another effect is related to the collection process, i.e. the time needed for the charges to be collected by the electrode<sup>3</sup>,  $t_c$ . Both  $t_c$  and the *plasma time* depend on the electric field applied and, as this varies with the penetration depth, they will depend on the impinging particle range (which itself depends on the particle properties). By exploiting these considerations, as well as the fact that high  $Z$  particles have smaller range than low  $Z$  ones for a given incident energy, we can evaluate the performances that can be obtained in two possible configurations of the entrance window in the silicon detector. When the particle

---

<sup>2</sup>Since particles must travel a minimum path into the detector in order to be identified through PSA, an energy threshold for identification still exists, as we will show in chapter 6 [14].

<sup>3</sup>Due to this contribution fragments with low  $Z$  value ( $Z=1,2$ ) can be identified. For these light fragments, in fact, the *plasma time* is practically negligible.

impinges on the high electric field side (i.e. the junction side) we say that the detector is mounted in a “front” configuration. In this case:

- the *plasma time* will be reduced for high-Z particles (low range) and increased for low-Z particles (high range);
- $t_c$  will be smaller in the first case and higher in the second;

The latter assumption holds for silicon detectors with a n-bulk (as in our measure). In this case, the holes are collected by the junction electrode. Therefore, in “front” configuration, heavier fragments produce electron-hole pairs closer to the junction-side than light fragments (so  $t_c$  is smaller). On the other hand, if the particles imping on the low field side (“rear” configuration), the two effects are swapped.

Since high-Z fragments have longer *plasma time* than low-Z ones, entering from the low-field side produces better results than entering from the high-field side, as it can be seen in fig. 1.3, which is obtained using a fully depleted detector [9]. As apparent from the figure, at full depletion the PSA based on the maximum of the current signal gives no identification when particles enter the detector from the junction (front) side. In the last

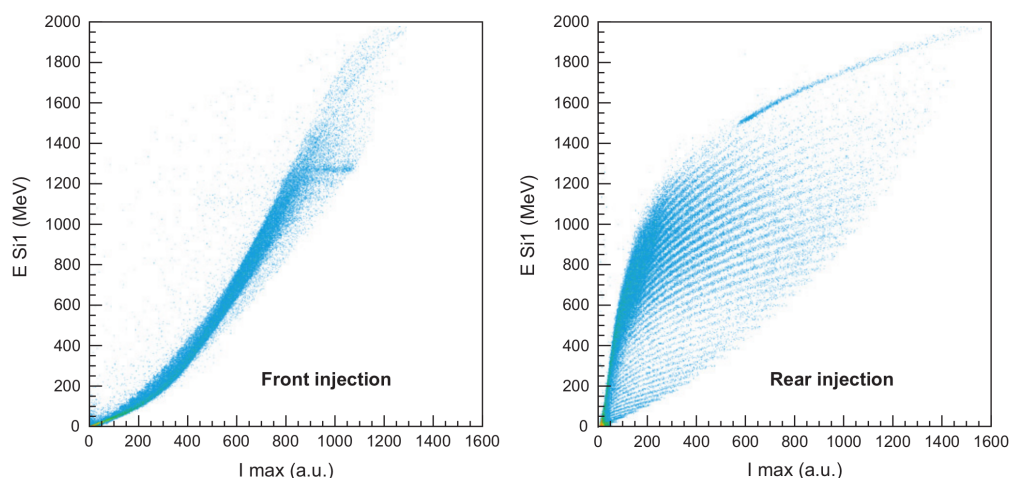


Figure 1.3: A comparison between PSA matrixes obtained with the depletion bias voltage applied using a front mounted silicon detector (left) and a rear mounted one (right). The increase of the performances is evident. This picture is taken from ref. [9].

years, the NUCL-EX experimental collaboration (which includes the Florence group) has made a wide use of PSA techniques, studying their performances and trying to make them better. The “Total Charge vs. Rise-Time” correlation is used in the GARFIELD apparatus [2], for which the current signal from the detector is presently not acquired.

The NUCL-EX collaboration is also involved in building the FAZIA telescope array [10]. The FAZIA front-end electronics acquires also the current signal (using the current output of a PACI preamplifier<sup>4</sup>) [11] and can then exploit both correlations. During the first

<sup>4</sup>This system, implemented with the first FAZIA prototype telescopes, and used in this thesis, is not used anymore. Instead, the current signal is obtained from the charge signal by differentiating it by means



beam tests of the FAZIA prototype telescopes, a comparison between the two identification methods has been made, and the result was that the “Total Charge vs. Maximum of the Current Signal” yields better results [12, 13].

For this reason, implementing an online method to obtain the current signal could be very useful for the GARFIELD apparatus, allowing for the production of Charge vs Max Current matrices and possibly enhancing the performance when it comes to particle identification.

Another important feature of the PSA identification with silicon detectors is its dependence on the bias voltage of the detector itself. In fig. 1.4 the two PSA correlations described here are shown for different values of the bias voltage applied to the detector. By lowering the bias, the electric field is lowered, increasing both  $t_c$  and the *plasma time* and their variation with the particle type, thus increasing the identification performances. This happens, however, at the cost of an increase in the energy threshold for identification [14].

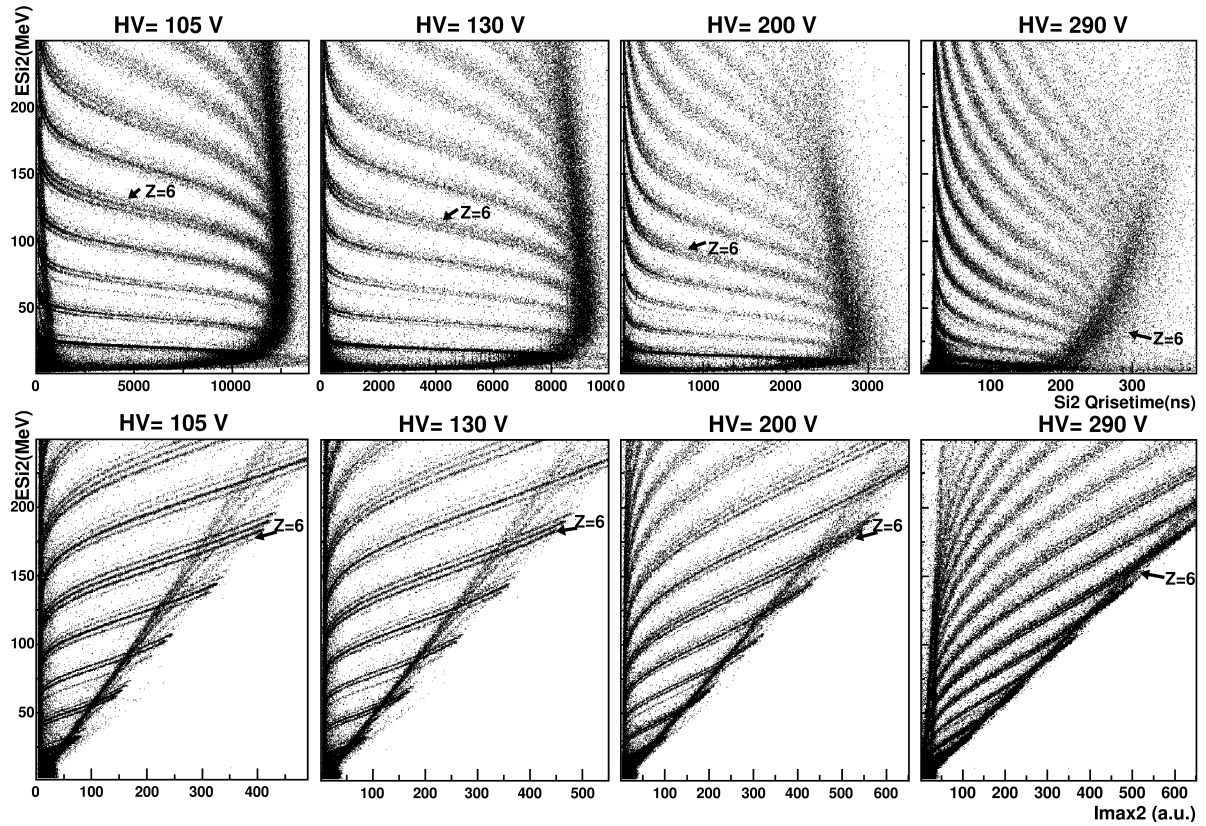


Figure 1.4: Some correlation matrices obtained with a rear-mounted  $500\mu\text{m}$  thick silicon detector with different bias voltages applied. Reducing the HV (thus the electric field) produces longer charge rise times and thus smaller maximum current are obtained. Better isotopic identification is obtained for lower bias voltage, though the energy threshold for identification increases [14].

of a series capacitor.



## 1.3 Outline of the Thesis

This thesis is subdivided into six chapters. Apart from the introductory chapter (this one) chapters 2 to 4 are dedicated to the basic principles of digital signal processing applied in this work, with some sections dedicated to their practical implementation. In particular:

- chapter 2 describes the energy measurement process by mean of a trapezoidal filter;
- chapter 3 focuses on the generation of trigger signals;
- chapter 4 is devoted to the interpolation procedure.

The remaining three chapters are dedicated to the description of the experiment performed at LNS. The experimental setup and the obtained results are presented respectively in chapters 5 and 6. In the end, our concluding remarks are reported, together with an overview of possible future developments.



## Chapter 2

# Energy Measurement

One of the main features of an apparatus for Nuclear Physics experiments is its ability to measure the energy of the detected nuclear fragments. A particle entering a detector loses its energy according to Bethe-Block's law<sup>1</sup>; the particle energy (or part of it) is deposited into the active area of the detector inducing some physical changes in it [6]. In different types of detectors, different types of information carriers are produced, e.g.:

- in ionization chambers the gas contained in the chamber is ionized, and electron-ion pairs are produced;
- in semiconductor detectors and inorganic scintillators, electron-hole pairs are produced;
- in organic scintillators, the molecules of the scintillating material are excited, and they subsequently decay to the ground state, emitting photons in the process.

In all cases, except for the organic scintillators (which experience a non-linear dependence due to quenching effects at high stopping power values), the average number of carriers produced is directly proportional to the total deposited energy. In the following, we will concentrate on semiconductor detectors. To measure the energy released by a particle in a silicon detector, the read-out circuit integrates over time the current induced by the moving charge carriers (electron and holes), producing a *charge signal*. The energy information is then carried by the charge signal and must be extracted from it. The extraction of the energy information from the signal is treated in this chapter.

The charge signal produced by the preamplifier is affected by noise which is to be taken into account. Noise sources are, for instance, the charge preamplifier itself (mainly its first amplifying stage), the input stage of the digitizer and the ADC (for sampled signals). Noise affects our charge measurement in two ways. First, it affects the evaluation of the maximum. For a given deposited energy, the extracted maximum will fluctuate due to the effect of noise. Moreover, in most cases, the signal arises over a DC level (the base-line) which must be subtracted in order to obtain the actual maximum amplitude of the signal. In order to achieve a better precision, the signal is often filtered in order to increase the

---

<sup>1</sup>Though at the end of the track, Bethe-Block formula is not valid anymore.

SNR<sup>2</sup> before checking for the maximum.

When dealing with digitized signal, the ADC contribution to noise must be taken into account. The ADC noise comes from different sources [15]:

- the quantization noise (approximation of a continuous value to a discrete one);
- the clock jitter noise (small phase changes from clock cycle to clock cycle);
- others (aperture jitter, non-linearities etc...).

This noise will be added to the one already present on the signal (due to preamplifier and input stage) and the variance of the different sources will add up, allowing for two extreme cases:

- the ADC noise dominates over the input stage noise
- viceversa, the input stage noise dominates.

The first case is usually undesired, because it implies that the digitizing process is reducing the SNR. However, it is sometimes unavoidable (e.g. when the dynamic range at the preamplifier output is much wider, in terms of voltage, than the input range of the ADC). To increase the SNR of the charge signal, a common choice is to apply a trapezoidal filtering to the signal, both for analog and digitized signals. The trapezoidal filtering reduces the noise level thus increasing the precision of the energy measurement. Further details can be found in ref. [16].

The trapezoidal shaper, as its name suggests, transforms a step signal in a trapezoidal signal. The trapezoidal shape is characterized by two parameters:

- the rise-time, e.g. the duration of the leading/trailing edges of the trapezoid;
- the flat-top, e.g. the duration of the flat portion of the trapezoid.

The effects of such a filtering on the noise can be analyzed using the model proposed by F. Goulding [17]. The model treats noise as composed by two components:

- DELTA noise, e.g. a sequence of delta-like impulses with random time distribution;
- STEP noise, e.g. a sequence of step impulses with random time distribution.

From the model one gets a mathematical method to evaluate how these components are transformed by the filtering. Using Goulding's method one finds that, in presence of pure white noise (i.e. delta noise only, no step noise), the best SNR is given by a triangular shaper, both in continuous-time domain [18] and in discrete-time domain [16]. A triangular shaper can be thought of as a trapezoidal shaper with a flat-top length of zero. On the other hand, using such a short value of flat-top introduces another problem,

---

<sup>2</sup>SNR stands for **S**ignal to **N**oise **R**atio and it is the ratio between the RMS amplitude of the signal and the standard deviation of the noise present on the signal. In fact, as we are interested in measuring the maximum amplitude of the signal, we will use this latter value instead of the RMS value in the following.

the “ballistic deficit” [16, 19]. In fact, the shaper produces a perfect triangular/trapezoidal shape only if applied to a perfect step signal. If the input is a realistic charge signal, with a finite rise-time, the output is different from a trapezoid, and it is possible for the trailing edge of the output to begin before the signal has reached its maximum. By properly setting the flat-top parameter, it is possible to ensure that even the slowest charge signal produces an output that reaches its real maximum before decreasing.

The preamplifier does not actually perform a mathematical integral of the current signal. Instead, the preamplifier response presents an exponential decay. If the decay time of the preamplifier is comparable with the charge signal rise-time, the decay will affect the result of the measurement, introducing a systematic contribution to the “ballistic deficit”. To eliminate this unwanted effect one has to perform additional filtering to the charge signal. This filtering is described in sec. 2.1.1 (continuous-time approach) and in sec. 2.2 (discrete-time approach).

## 2.1 Obtaining the Energy information from a signal

We remind that we call *charge signal* the signal produced by integrating of the current induced on the electrodes by the drifting charge carriers. As anticipated in the previous section, charge signals are usually produced by a charge preamplifier of the type depicted in fig. 2.1, which is not an ideal integrator; in fact, the RC loop which is present in the feedback branch of the operational amplifier has the effect of an exponential decay of the charge signal, with a decay constant  $\tau$  which is given by the product  $R_f C_f$ .

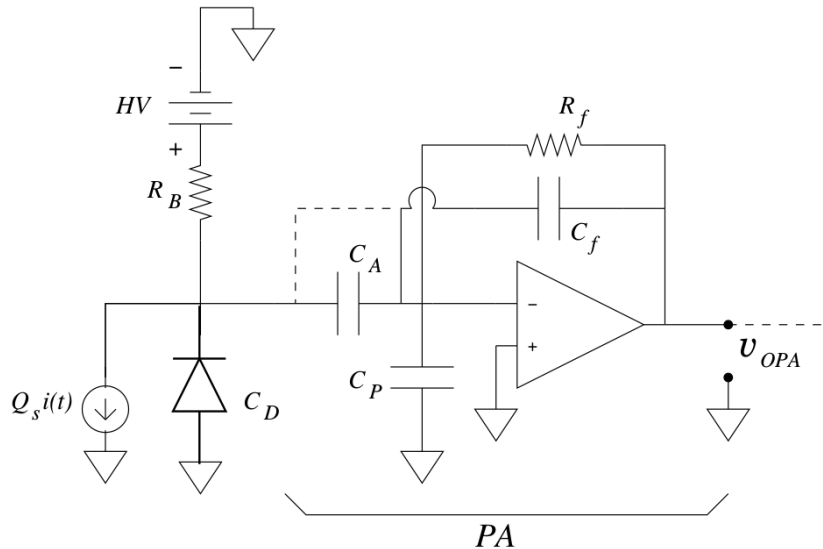


Figure 2.1: A scheme of a typical read-out chain for a silicon detector, implementing a charge preamplifier. Picture taken from [20].

We will now introduce the various filtering stages needed to perform the energy measurement described in continuous-time domain. In the following we exploit the properties of LTI filters [21]. LTI, which stands for **L**inear and **T**ime **I**nvariant, is a family of systems for which linearity and time unvariance properties apply. Denoted with  $F(x(t))$  the application of a LTI filter  $F$  to the signal  $x(t)$  we can write :

- $F(ax(t) + by(t)) = aF(x(t)) + bF(y(t))$
- $F(x(t)) = y(t) \rightarrow F(x(t + t')) = y(t + t')$

where the first property describes linearity and the second time invariance. Defined  $h(t) = F(\delta(t))$ , where  $\delta(t)$  is Dirac's delta, it is then possible to evaluate  $F(x(t))$  as

$$F(x(t)) = \int_{-\infty}^{\infty} h(t - T)x(T)dT = (h * x)(t) \quad (2.1)$$

where  $*$  denotes a convolution product. In fact, since

$$x(t) = \int_{-\infty}^{\infty} \delta(t - T)x(T)dT = \int_{-\infty}^{\infty} \delta(T)x(t - T)dT \quad (2.2)$$

so that, treating  $x(T)$  as a parameter, we obtain

$$F(x(t)) = \int_{-\infty}^{\infty} F(\delta(t - T))x(T)dT \quad (2.3)$$

from which comes eq. 2.1. Eq. 2.1 indicates a crucial property of LTI system: the filter behaviour is entirely described by its impulse response. For more information about this family of filters, see ref. [21].

### 2.1.1 Pole-Zero correction

The exponential decay of the charge signal can be described in terms of the response function of the preamplifier. In fact, as long as the preamplifier output stays inside the region where the preamplifier behaviour is linear, the preamplifier behaves as a LTI filter. In the continuous time domain, an ideal accumulator has response function:

$$\hat{h}_{int}(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \quad (2.4)$$

while the impulse response of the the charge preamplifier (exploiting an ideal operational amplifier) with RC feedback is given by:

$$h_{int}(t; \tau) = \begin{cases} 0 & t < 0 \\ e^{-t/\tau} & t \geq 0 \end{cases} \quad (2.5)$$

Let's assume that the current signal  $i(t)$  is constant over an interval  $\Delta t$ , zero outside this interval and with unit integral:

$$i(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{\Delta t} & 0 \leq t < \Delta t \\ 0 & t \geq \Delta t \end{cases} \quad (2.6)$$

The charge signal produced by the preamplifier,  $c(t)$ , is the convolution product of the current signal and the preamplifier impulse response. For the ideal integrator we obtain:

$$\hat{c}(t) = \begin{cases} 0 & t < 0 \\ \frac{t}{\Delta t} & 0 \leq t < \Delta t \\ 1 & t \geq \Delta t \end{cases} \quad (2.7)$$

For a charge preamplifier with RC feedback (real integrator), the result is:

$$c(t; \tau) = \begin{cases} 0 & t < 0 \\ \frac{\tau}{\Delta t} (1 - e^{-\frac{t}{\tau}}) & 0 \leq t < \Delta t \\ \frac{\tau}{\Delta t} e^{-\frac{t}{\tau}} (e^{\frac{\Delta t}{\tau}} - 1) & t \geq \Delta t \end{cases} \quad (2.8)$$

In the first case, the charge signal reaches a maximum that is equal to the integral of the current signal, in the second case we need to make some remarks. Due to the monotony of  $c(t; \tau)$  in the three intervals, the maximum is obtained when  $t = \Delta t$ :

$$\max(c(t; \tau)) = \frac{\tau}{\Delta t} (1 - e^{-\frac{\Delta t}{\tau}}) \quad (2.9)$$

If the decay constant of the preamplifier is much longer then the duration of the current signal ( $\tau \gg \Delta t$ ), it's possible to approximate the exponential to first order, obtaining:

$$\max(c(t; \tau)) = \frac{\tau}{\Delta t} (1 - 1 + \frac{\Delta t}{\tau}) = 1 \quad (2.10)$$

That is, under this condition, the maximum of the charge preamplifier output is equal to the integral of the current signal; otherwise, if this condition does not hold, the integral can not be reconstructed exactly. In order to avoid this effect, known as *ballistic deficit*, it is possible to apply to the charge signal some further filtering, to compensate for the exponential decay. The correction filter turns out to be a member of the LTI filters family, characterized by its impulse response function  $h_{PZC}(t)$  which is to be determined by requesting that:

$$(h_{PZC} * c)(t) = \hat{c}(t). \quad (2.11)$$

Using the expressions for  $c$  and  $\hat{c}$  and the properties of the convolution product it's easy to obtain

$$(h_{PZC} * h_{int})(t) = \hat{h}_{int}(t). \quad (2.12)$$

The solution of this equation is as follows

$$h_{PZC}(t) = \delta(t) + \frac{1}{\tau} \theta(t) \quad (2.13)$$

where  $\theta(t)$  is the Heavyside step function:

$$\theta(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} . \quad (2.14)$$

This particular processing transforms an exponential decay with time constant  $\tau$  into a Heavyside step function. We call it pole-zero **correction** (PZC), in order to avoid confusion with another kind of processing (known in electronics as pole-zero **cancellation**) which transforms an exponentially decaying function into a Dirac's delta. In short, the effect of PZC is to take the output produced by a charge preamplifier and transform it in the signal that an ideal accumulator would have produced. The output would then have an infinite length in time. However, since PZC is actually part of our trapezoidal shaping, the final output in response to an exponentially decaying signal would be a finite lenght trapezoid. This is the kind of shaping we will assume in what follows.

## 2.2 The digital trapezoidal shaper with PZC

Up to now, all considerations have been made in the hypothesis of working with analog signals. In the following paragraphs these conclusions will be revisited in order to apply them to the discrete time domain. When dealing with discrete-time systems, the continuous time signal  $s(t)$  is substituted by the discrete time sequence  $s[n]$  with  $n \in Z$ , and operators are replaced by their discrete-time counterparts. Among them we have the derivative, which is substituted by a first backward difference, the integral, substituted by an accumulator, and the convolution integral, substituted by a convolution sum [21]. A summary of these operators is shown in table 2.1. The convolution sum has the same properties of the convolution integral; its commutativity, in particular, is exploited in the following. The properties of discrete-time LTI filters are equivalent to the continuous-time filter ones, and they can be obtained by substituting continuous-time operators with discrete-time ones and the Dirac's delta function with its discrete-time equivalent, the unity delta sequence:

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (2.15)$$

|             | Continuous   | Discrete  |
|-------------|--|---|
| derivation  | $s'(t) = \lim_{h \rightarrow 0} \frac{s(t+h)-s(t)}{h}$ | $s'[n] = s[n] - s[n-1]$                             |
| integration | $\int_A^B s(t)dt$                                      | $\sum_{n=A}^B s[n]$                                 |
| convolution | $(s * r)(t) = \int_{-\infty}^{\infty} s(x)r(t-x)dx$    | $(s * r)[n] = \sum_{m=-\infty}^{\infty} s[m]r[n-m]$ |

Table 2.1: The discrete time operators corresponding to some basic continuous time operators

In particular, defined  $y[n] = F(x[n])$  the result obtained by applying an LTI filter to a signal, in analogy with the continuous-time case, we get that:

- $F(ax[n] + by[n]) = aF(x[n]) + bF(y[n])$



- $F(x[n]) = y[n] \rightarrow F(x[n + M]) = y[n + M]$

where the first property expresses linearity and the second expresses time-invariance. By defining the filter impulse response  $F(\delta[n]) = h[n] = (h * \delta)[n]$ , we obtain that:

$$F(x[n]) = F\left(\sum_k x[k]\delta[n - k]\right) = \sum_k x[k]F(\delta[n - k]) = \sum_k x[k]h[n - k] = (x * h)[n] \quad (2.16)$$

This means that, as for continuous-time LTI filters, also discrete-time LTI filters are fully characterized by their impulse response and, exploiting the commutativity of the convolution sum, it follows that LTI filters (generally speaking, LTI systems) commute

$$F_1(F_2(x[n])) = F_2(F_1(x[n])) \quad (2.17)$$

Furthermore, discrete-time LTI filters can also be described in terms of constant coefficients finite difference equations [21], which have the general formula

$$y[m] + \sum_{n=1}^{N_1} B[n]y[m - n] = \sum_{n=0}^{N_2} A[n]x[m - n] \quad (2.18)$$

defining the  $m$ -th samples of the output sequence  $y[m]$  as a combination of the preceding samples of the output signal ( $y[m - n]$ ) and of the input samples. Finite difference equations are extremely useful when dealing with IIR filters, which are one of the two groups of LTI filters. In fact, since these filters are fully characterized by their impulse response  $h[n]$ , they have been divided in two subgroups:

- the FIR filters, with a response function that is zero outside a limited interval of indices;
- the IIR filters, that do not satisfy the condition to be FIR filters.

At variance with FIR filters, which can be easily calculated by convolution of the input signal with their impulse response, a difference equation is the only practical way to obtain a IIR system (a convolution sum would involve an infinite number of terms). An example of FIR filter is the first backward difference introduced in table 2.1 as a discrete-time derivation, while the discrete-time accumulator is a typical example of an IIR filter. Calculating the output of an accumulator via convolution yields

$$y[n] = (h * x)[n] = \sum_{m=-\infty}^n x[m] \quad (2.19)$$

i.e. a sum running over an infinite number of terms. However, eq. 2.19 can be rewritten in a simpler way, exploiting the previous output  $y[n - 1]$  of the calculation:

$$y[n] - y[n - 1] = x[n]. \quad (2.20)$$

This is a finite difference equation and it contributes an obvious simplification. Since the ADC digitizes the analog signal produced by the charge preamplifier, also the digitized signal will be affected by an exponential decay. Therefore, the discrete-time equivalent of the PZC filter introduced in eq. 2.13 must be applied to the signal to compensate the decay. The discrete-time PZC filter has an impulse response function:

$$h_{PZC}[n] = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ 1 - a & n > 0 \end{cases} \quad (2.21)$$

where  $a = e^{-\frac{\tau_{clk}}{\tau}}$  and  $\tau_{clk}$  is the sampling period. This filter can be also viewed as a cascade of two LTI filters. The first is a FIR filter, the digital equivalent of the pole-zero cancellation, with impulse response:

$$h_{pzc}[n] = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ -a & n = 1 \\ 0 & n > 1 \end{cases} \quad (2.22)$$

while the second is a simple accumulator.

The digital trapezoidal shaper is a FIR LTI filter that transforms a step signal (like the one produced by an ideal preamplifier) in a signal with a trapezoidal shape. This is achieved by cascading two LTI filter:

- a moving average filter;
- a delayed difference filter.

The moving average filter<sup>3</sup>, as the name suggests, performs an average of the signal over a fixed amount of samples,  $M$ . So, starting from a signal  $x[n]$  it generates a signal

$$y[n] = \frac{1}{M} \sum_{m=0}^{M-1} x[n - m] \quad (2.23)$$

and it has an impulse response function

$$h_{MovAv}[n; M] = \begin{cases} 0 & n < 0 \\ 1/M & 0 \leq n < M \\ 0 & n \geq M \end{cases} . \quad (2.24)$$

In short, the moving average consists in calculating the average value of the signal over a moving window containing  $M$  samples. The delayed difference filter simply subtracts to the signal a copy of itself, delayed by a fixed amount of samples. Its impulse response function is

$$h_{DelDiF}[n; M] = \begin{cases} 1 & n = 0 \\ -1 & n = M \\ 0 & otherwise \end{cases} . \quad (2.25)$$

---

<sup>3</sup>The moving average employed in this work is a “causal” moving average, i.e. the  $n$ -th output sample does not depend on input samples after the  $n$ -th input sample.

The moving average filter can be viewed as a cascade of a delayed difference, an accumulator and an attenuator. In fact

$$(h_{DelDif} * h_{accu})[n; M] = Mh_{MovAv}[n; M] \quad (2.26)$$

At this point, by using the properties of the LTI filters, we construct the complete trapezoidal filter with PZC as a cascade of:

1. a pole-zero cancellation filter;
2. two delayed differences (with different delays);
3. two accumulators.

Because of commutativity of the convolution product, the order in which these filters are applied should not count. However, this is not true when the filter is applied in real-time, to a signal which is not time-limited, on a system with a finite dynamic range. In this last case, the stability of the filter becomes a major concern, and the order in which the filters are applied to the signal becomes relevant (see sec. 2.2.1). The accumulators, in particular, represent a serious problem when applying a filter in real time to such a signal, because every deviation of the DC level from zero is integrated over time, giving ramp-like transients which can eventually saturate the digital range of the filter. To avoid digital saturation in intermediate stages, the sequence of the filters in the cascade has to be wisely chosen.

### 2.2.1 Implementation of the trapezoidal filter

We come now to the actual implementation of the trapezoidal filter. To avoid divergences and saturation, we have arranged the filter cascade so that each accumulator is preceded by a differentiation filter that suppresses the DC level, keeping it around zero. The signal fed to each accumulator will then have a DC level with zero average value. As sketched in the flow graph in fig. 2.2, some other elements were introduced just to increase the manageability of the system. One of these devices is the SHIFTER block (second rectangle from the beginning of the process in fig. 2.2), which shifts the digital values by a certain amount of bits to the left (thus introducing a gain) to guarantee that the range of the first accumulator is completely used. Analogous to the SHIFTER is the CUTTER block, which shifts the digital values to the right in order to avoid destabilization of the second accumulator<sup>4</sup>. Another improvement is a base-line restoration system implemented on both the accumulators which, at discrete intervals of time, adds or subtract one unit to the signal value, thus pushing the base line towards zero. This “decay” toward zero has a time scale much longer than the length of the signal, so it doesn’t affect its amplitude. In fig. 2.2, the three parameters introduced in previous sections have been renamed. For the Pole-Zero cancellation, instead of  $a$  we introduce a parameter called TAU, a rescaled version of  $a$ :

$$TAU = 2^{31}a = 2^{31}e^{-\frac{\tau_{clk}}{\tau}} \quad (2.27)$$

<sup>4</sup>By reducing the signal amplitude, also the DC fluctuations are reduced, increasing the stability of the accumulator

while the delays of the two delayed difference filters are called  $RT$  (for Rise-Time) and  $RTFT$  (Rise-Time plus Flat-Top). Figure 2.3 shows a schematic representation of the various intermediate signals that the filter generates starting from a perfect exponential signal with no noise, just to give a graphical hint on how the filter works. In these pictures all signals have been rescaled just for simplicity and the amplitude scale is not relevant; as a matter of fact the filter itself multiplies the signal by a gain factor whose value can be calculated by taking into account all contributions:

- the TAU parameter is a number in the interval  $[0,1]$ , and is represented in an integer 32-bit unsigned notation where 0x80000000 corresponds to value 1 (i.e. its value is multiplied by a factor of  $2^{31}$ );
- the left shift of the signal contributes with a factor  $2^{SHIFT}$  to the overall gain;
- after the shift, the 16 LSB's of the signal are discarded, and the remaining bits are aligned to the right, thus introducing a factor  $2^{-16}$ ;
- the right shift contributes a factor  $2^{-CUT}$ ;
- at last, the moving average is substituted by a sum over  $RT$  samples, thus giving a gain factor of  $RT$ .

With all these considerations it's possible to calculate the overall gain of the filter, which turns out to be:

$$G = RT * 2^{15+SHIFT-CUT} \quad (2.28)$$

To help the user in the setup phase, a multiplexer has been added so that it is possible, when setting up the digitizing channel, to acquire and view all the intermediate signals produced by the shaper. This is of particular importance when choosing the values of SHIFT and CUT parameters. The first is chosen by imposing that the range of the first accumulator is fully exploited at the same time avoiding saturation. The latter is needed to prevent saturation of the second accumulator (which has the same output range of the first) and is used also to reduce the DC level fluctuations of its input signal.

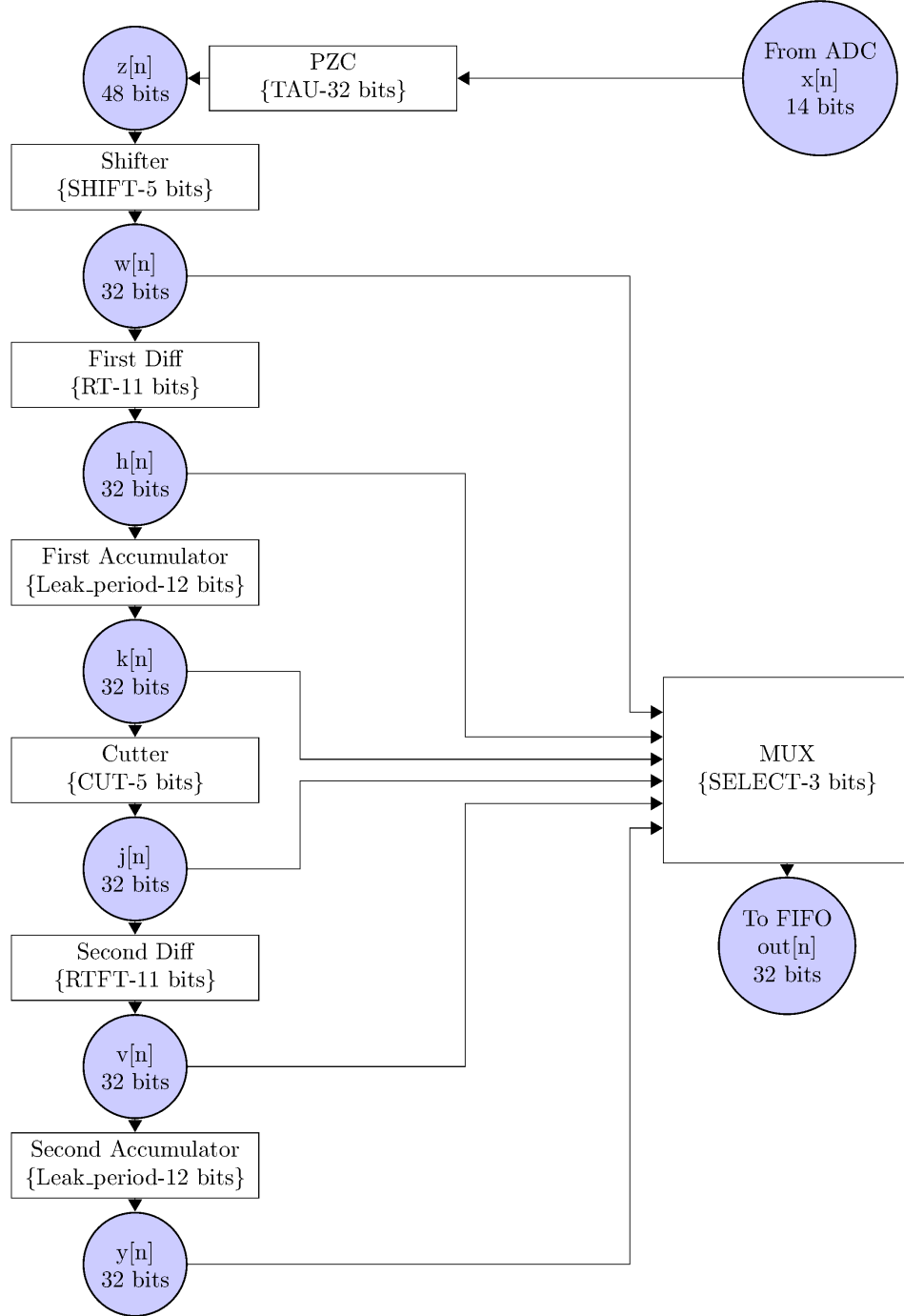


Figure 2.2: Block diagram of the trapezoidal filter with PZC

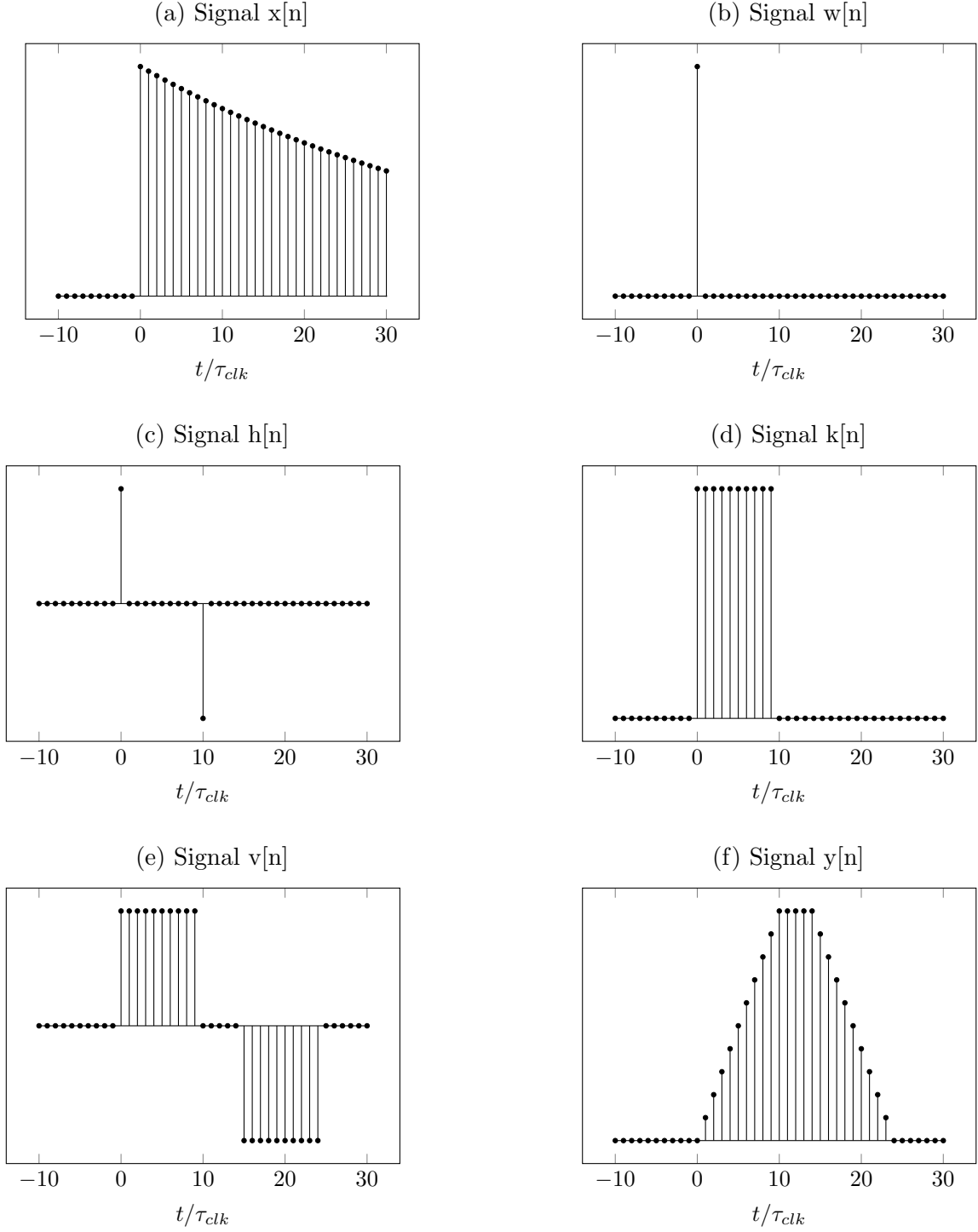


Figure 2.3: A schematic representation of the trapezoidal shaper algorithm, with the intermediate signals showed (see fig. 2.2). This example is obtained by using  $RT=10$  and  $RTFT=15$ .

### 2.2.2 Getting the Energy

To obtain a good measure of the energy deposited in the detector (which is coded into the amplitude of the trapezoidal shaper output), the shaper parameters introduced in sec. 2.2.1 must be chosen appropriately. This means that we must check that:

- the various shifting of the digital values do not spoil the energy information carried by the signal: this happens both if the signal saturates and if it is attenuated too much;
- the flat portion of the trapezoidal signal (flat top) is really flat;
- the flat-top value is proportional to the energy;
- the SNR of the output signal is better than the SNR of the input signal (otherwise the filter is useless).

Every parameter has an effect on the shaper output that must be taken into account. To avoid saturation/suppression of the signal we must change SHIFT and CUT parameters. The pole-zero correction constant (TAU) is responsible for the flatness of the flat top and for its value. An incorrect TAU will compensate the preamplifier decay only partially (generating a decaying flat top) or too much (generating an exponentially increasing flat top), also losing the proportionality. The SNR of the output signal, can be estimated starting from the input signal's SNR, using the simplified framework proposed by Goulding [17], and it depends on the parameters  $RT$  and  $RTFT$  [16].

If the parameters are well chosen, the measure of the flat-top amplitude yields better results than measuring the amplitude of the charge signal before filtering. The measure can be performed in many different ways, each with its pro and cons. We decided to use a peaking time measure, which consists in taking a single sample of the flat top of the trapezoidal signal. This sample is taken a fixed number of clock cycles after a triggering event occurred. This method is very easy to implement, but it is very sensitive to the shaper parameters value. In fact, we have to deal with a wide range of rise-time values and this can lead to two different problems:

- if the flat-top length is too short, slower signals will suffer from *ballistic deficit*;
- if the  $TAU$  parameter is wrongly set, the flat-top will not be flat.

To avoid *ballistic deficit* the solution is to use a long enough flat-top, though this will make the shaper output more sensitive to low frequency noise at the input [16]. A decaying flat top gives problems because of the time jitter of the trigger. The peaking time method, combined with a decaying flat, produces an underestimate of the signal amplitude. If the trigger system is immune from amplitude and rise-time walk, the underestimate would be the same fraction for all signals, so that it could be taken easily into account in a linear calibration procedure. Problem arise, however, when the trigger system is affected by rise-time walk, as in our case (see chapter. 3). The delay between the onset of the signal and the trigger occurrence would be longer for slower signals, which would be affected by a

larger underestimate of the amplitude, thus spoiling the linearity of the system. Another downside of the peaking time method is related to its SNR. The noise affecting the sample taken from the flat top is the same affecting each sample of the trapezoidal signal. Noise fluctuation could be decreased, at the cost of an increase in complexity, by taking the average value of a few samples of the flat-top instead of just one sample. The choice of the simplest method was driven mainly by the lack of time before the experiment at LNS that made necessary to prefer easy implementable methods over more complex ones, but we plan to add and test other methods in the future.



## Chapter 3

# The Trigger system

A digitizing channel for nuclear radiation detectors is in some respects similar to a digital oscilloscope: though some parts of the system are continuously running in real-time, it needs a trigger information in order to perform signal acquisition and processing. Moreover, the channel can contribute in building the global trigger of the experiment, whose role is to select for acquisition the collision events which are of particular physical interest. In order to respond to trigger information and to contribute to the global trigger, our digitizer has been equipped with a complete trigger system, including:

- a fast bipolar shaper that produces a bipolar signal used for internal trigger detection;
- a state machine that analyzes the output of the shaper and generate the internal trigger;
- a “trigger box” unit capable of complex triggering schemes.

All these features are described in this chapter.

As mentioned in sec. 3.1.1, using a bipolar shaper produces a trigger system which is intrinsically immune from amplitude walk. However, bipolar shaping does not solve the rise-time walk problem. The zero crossing of a bipolar shaper like the one implemented on our board has a systematic dependence on the rise-time of the input signal. When the input signals have a wide range of possible rise-times, like in our test at LNS (described in chapters 5 and 6) this effect could affect the amplitude estimation (as mentioned in sec. 2.2.2). A rise-time walk correction should be applied to get the proper time of occurrence of the signal. Since, in our measurement at LNS, we were not interested in using the trigger information for timing measurement, such a correction has not been applied in this work.

### 3.1 Bipolar signal generation

Each digital channel features a configurable trigger generation system. Its input is the digitized sequence from the on-board ADC. The trigger system can generate an internal

trigger which can be used to elaborate a more complex trigger (see section 3.3). In order to recognize the presence of a signal from the detector and to generate a logical signal indicating its presence, the trigger system first filters its input through a fast bipolar shaper (typical shaping times of about 200 ns). The fast bipolar shaper reduces noise fluctuations and eliminates the DC component from the signal, at the same time preserving as much as possible the rise-time of fast transients. A trigger detection logic analyzes the output of the bipolar shaper and generates the trigger. Fig. 3.1 outlines the basic concepts: the bipolar signal is represented in blue. When the first threshold (dashed red line) is trespassed (red circle), the trigger logic looks for a zero crossing (black circle). When a zero crossing occurs, a logic true signal is generated at the output of the trigger system.

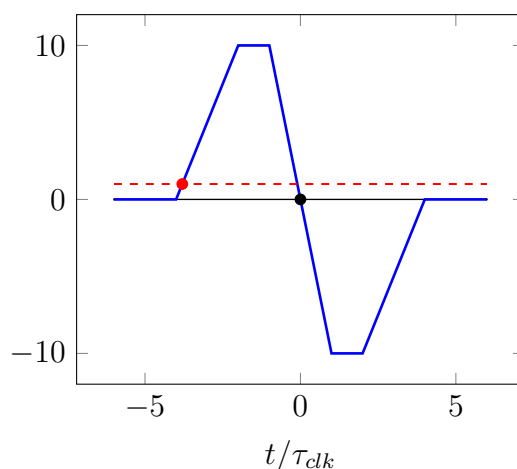


Figure 3.1: A simple graphical representation of the trigger detection logic. The signal generated by the bipolar shaper (blue line) is monitored continuously in order to detect the crossing of the threshold level (red dashed line). Once the threshold is passed (red circle), the system waits for the time in which the signal crosses the baseline (black circle). For a more accurate description of the algorithm see sec. 3.2.

### 3.1.1 Advantages of using a bipolar shaper

The trigger generation system might seem, at first, a bit complex. However, this solution is needed in order to have a trigger system which is immune to amplitude walk.

Amplitude walk is a problem which is intrinsically present when the crossing of a fixed amplitude threshold is used to generate the trigger. The best way to understand the phenomenon is to consider a signal with a linear rising edge of constant length but variable amplitude:

$$s(t) = \begin{cases} 0 & t < 0 \\ At & 0 \leq t < 1 \\ A & t \geq 1 \end{cases} \quad (3.1)$$

If we define a threshold value  $B$  and a threshold crossing time  $\hat{t}$  which will satisfy the condition  $s(\hat{t}) = B$  then two cases arise:

- if  $A < B$ ,  $\hat{t}$  is not defined. The signal does not generate any trigger and it is simply ignored;
- otherwise, the solution yields  $\hat{t} = B/A$ , which obviously depends on the amplitude of the signal.

To understand why using a bipolar shaper and a zero-crossing detection system grants immunity to amplitude walk, let's have a closer look at the shaper, basically a trapezoidal shaper without (Pole-Zero correction) combined with a delayed difference. Like in chapter 2, the trapezoidal shaper can be viewed as a cascade of two filters:

1. a sum, performed over  $RT$  samples (substituting the moving average and then introducing a gain factor  $RT$ );
2. a delayed difference (with delay  $RTFT$ ).

The pole-zero correction is not needed here; in fact, due to the fast shaping time used for this filter, the exponential decay of the charge signal has practically no effect on the shaper output. The output signal of the trapezoidal filter is then passed through another delayed difference with delay  $RTFT$ . These steps are represented in fig. 3.2. To demonstrate our assumption we will do our calculation in the continuous-time domain and ignore the gain factor  $RT$  introduced by the summation. Let  $x(t)$  be the shape of the input signal  $s(t) = Ax(t) + B$  where  $A$  and  $B$  are not fixed. After the application of the continuous-time equivalent of the moving average filter<sup>1</sup>, which is linear, we have the averaged signal  $\bar{s}(t) = A\bar{x}(t)$ . According to the description of the shaper, the final signal will be of the form:

$$s_{final}(t) = A(\bar{x}(t) - 2 * \bar{x}(t - RTFT) + \bar{x}(t - 2 * RTFT)) \quad (3.3)$$

At this point, some conditions must be imposed:

- the two delayed differences suppress the possible DC offset, so the final signal has a DC level equal to zero;
- since  $s(t)$  is the charge signal generated by integrating an unipolar current signal, if we neglect the preamplifier decay time it will be a monotonic function.

Here we assume that the current signal has positive polarity, so that the charge signal will be a monotonic increasing function. With these conditions let's require the signal shape to be a case-defined function:

$$x(t) = \begin{cases} 0 & t < 0 \\ f(t) & 0 \leq t < T_r \\ 1 & t \geq T_r \end{cases} \quad (3.4)$$

---

<sup>1</sup>According to tab. 2.1 the continuous-time equivalent of the moving average over  $M$  samples is a filter which operates on signals in the following way

$$y(t) = \frac{1}{M\tau_{clk}} \int_0^{M\tau_{clk}} x(t - z) dz \quad (3.2)$$

where  $f(T_r) = 1$  and  $f(t)$  has positive first derivative in  $(0, 1)$ . With these assumptions it's easy to demonstrate that  $s_{final}(t)$  is zero outside the interval  $(0, 2 * RTFT + T_r)$ .

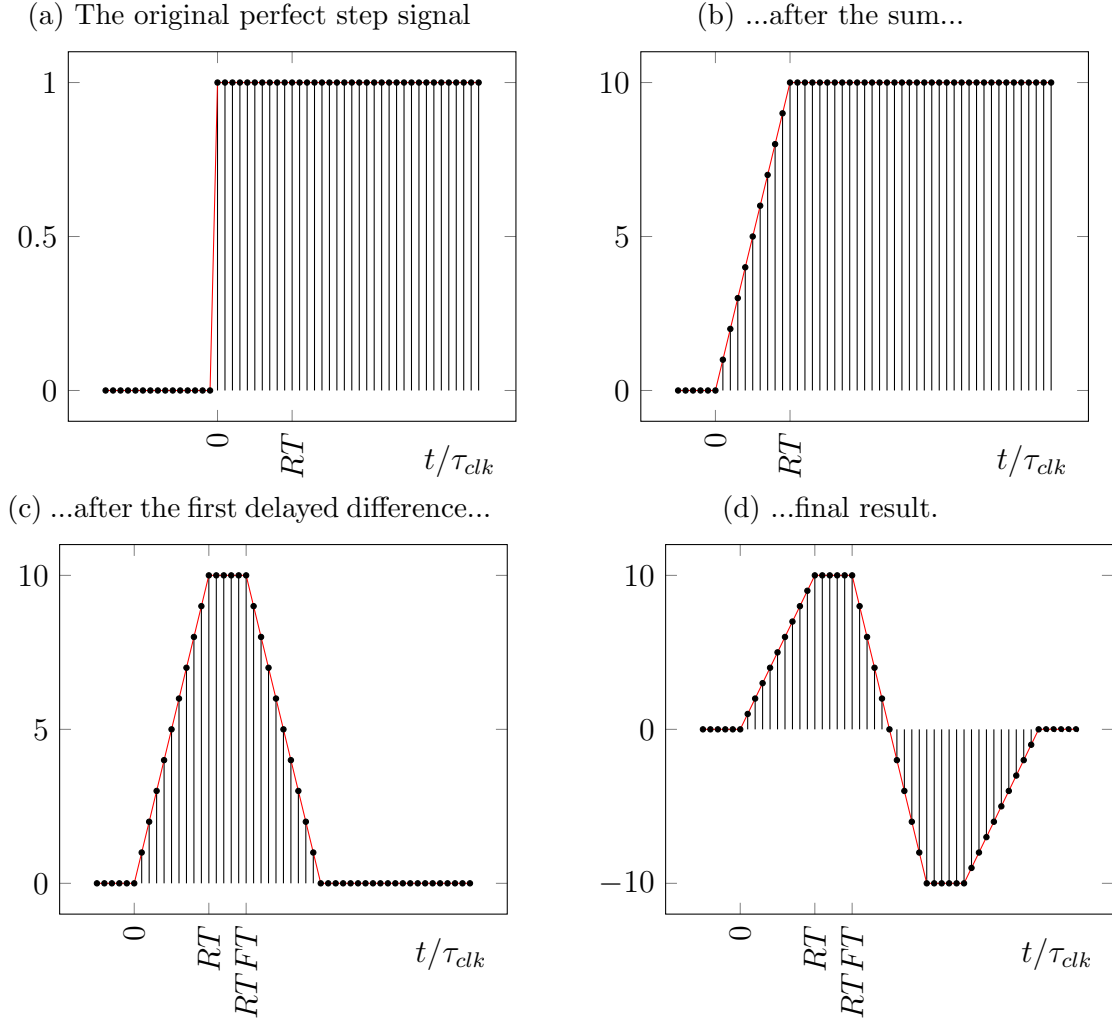


Figure 3.2: The steps of the construction of the bipolar signal. Picture (a) represents a perfect step signal which is fed to a sum filter to obtain the signal showed in (b). The signal goes through two identical delayed differentiation originating the signal in (c), taken after the first differentiation, and the one in (d). In this picture, the sum was performed over 10 samples at a time and the parameter  $RTFT$  (the delay of the two differentiations) was set to 15.

Inside the interval, a zero exists because:

1.  $s_{final}(t)$  is continuous;
2.  $s_{final}(\delta t) > 0$ ;
3.  $s_{final}(2 * RTFT + T_r - \delta t) < 0$ ;

The time  $t_0$  at which the signal crosses zero can be evaluated once the function  $f(t)$  is known. In any case, since the value of  $t_0$  is solution of the equation

$$A(\bar{x}(t_0) - 2 * \bar{x}(t_0 - RTFT) + \bar{x}(t_0 - 2 * RTFT)) = 0 \quad 0 < t_0 < 2 * RTFT + T_r \quad (3.5)$$

it follows that  $t_0$  doesn't depend on the amplitude of the original signal ( $A$ ), but only on its shape.

## 3.2 Zero crossing detection

Before describing the zero-crossing detection logic, we must confront ourselves with the real working conditions. In fact, we have neglected up to now some points that must be taken into account:

- the rise-time of the signal is not always the same
- as for the slow shaper, it is mandatory to guarantee the stability of the bipolar shaper.
- to avoid complications in the measurement of the zero-crossing time, we need signals, after filtering, with no DC offset;
- the actual signal is affected by noise, which will be present even after filtering;

The first item is responsible for the rise-time walk effect. In sec. 3.1.1 we have seen how the zero-crossing time of the bipolar shaper output depends only on the signal shape. Therefore, if the shape varies, the zero crossing time will vary. For this reason, the trigger implemented system is not immune from rise-time walk effects and, in general, from effects related to the signal shape.

To address the second issue we need to specify the actual implementation of the shaper. The system is actually a cascade of (in this order):

1. a delayed difference with delay  $RT$ ;
2. a delayed difference with delay  $RTFT$ ;
3. an integrator (which, in combination with the first delayed difference, makes up the sum filter);
4. a third delayed difference with delay  $RTFT$ , equal to the second one.

Since each differentiation suppresses the possible DC offset, we can be sure that the input signal of the accumulator has no offset. We then use the third differentiation to suppress the long period fluctuations of the integrator output. This will solve the problem of stability of the filter and, partially, the problem of the DC offset.

To completely solve the DC problem, we have implemented a baseline subtraction block within the zero crossing detection logic. The baseline value to be subtracted is evaluated by applying a moving average to the shaper output signal. The average is actually performed on a delayed copy of the signal, because it is important that samples pertaining to the leading edge of a signal are not included into the average. When the first threshold is trespassed by the **undelayed** signal, the current value of DC offset is kept in memory and used for baseline subtraction. Since the baseline is calculated on the delayed signal, it is not affected by the first samples of the leading edge.

Coming to the noise problem, the characteristics of the noise present on the shaped signal are fixed once the shaping parameters are chosen and so, apart from optimizing the filter rise-time parameter (which determines the length of the moving average), we can do nothing but set the threshold at a level which prevents the system from triggering on noise.

In order to illustrate the zero-crossing detection logic, we will make use of the flow-chart of the associated algorithm (see fig 3.3). The flow chart represents a state machine with two states (blue rectangles) as well as the operations performed by the state machine on clock transitions. The behaviour of the state machine depends on two internal variables, called TRIGD and CNTR, on two external constants (THS, TIME) and, obviously, on the bipolar shaper output (Sig). Furthermore, it manages the variable UPDATE. A “true” value of UPDATE enables evaluation of the DC offset of the shaper signal. When UPDATE is “false”, the system keeps in memory the last evaluated DC value. At the beginning, or after a reset, the machine stays in the IDLE state as long as the signal stays below the threshold value. When in IDLE, the DC offset is evaluated and continuously subtracted from the signal by the baseline subtraction block. After threshold trespassing, the machine stays in COUNTING state for a fixed amount of clock cycles (TIME) looking for the zero crossing. When (and if) the zero crossing happens, the internal variable TRIGD is asserted, in order to avoid repeated triggers, and a trigger event is generated. Then, the machine counts TIME clock cycles before getting back to IDLE state, going back to the starting conditions.

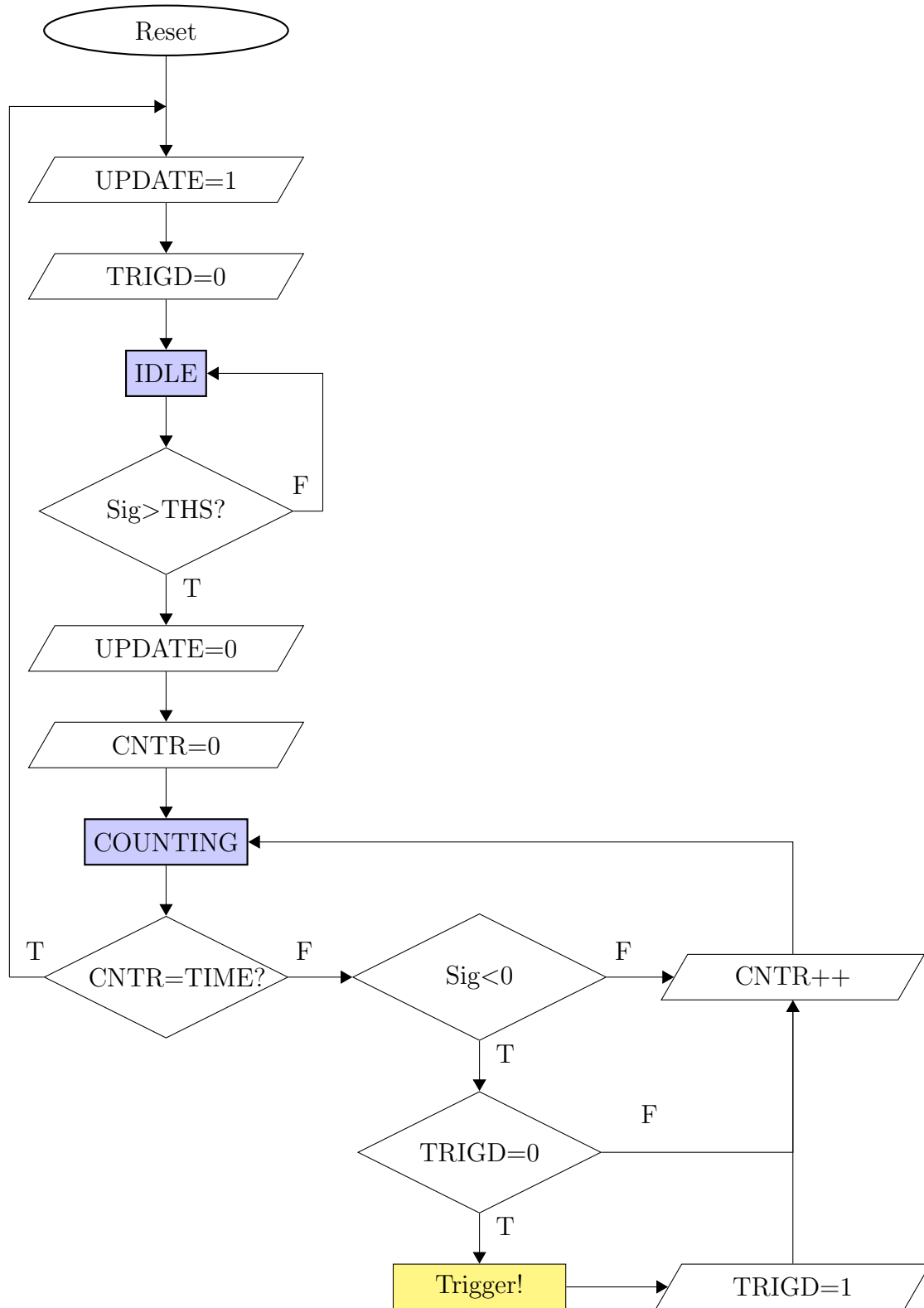


Figure 3.3: The flow-chart of the zero crossing detection algorithm.

### 3.3 The trigger box

The FPGA firmware has also been equipped with a configurable trigger logic, to be used for handling complex triggering schemes. The logic unit (*trigger box* from now on) can be represented by the logic diagram shown in figure 3.4. It can work in lots of different modes. This logic device has been designed years ago by P. Edelbruck of IPNO-Orsay for the FAZIA apparatus and we have adapted it to work with our boards. The behaviour of the trigger box is controlled by three registers: SOURCE, MODE and OUT.

The SOURCE register controls the demultiplexer (inside the red frame in fig. 3.4) and selects one of the inputs to be sent forward.

The OUT register controls the other demultiplexer (framed in blue in 3.4) and chooses one of the signals to be sent outside the board. The output of this multiplexer is connected to the right-angle 8-pin angular connector of the board (see fig. 5.5).

Last, but not least, the MODE registers controls all the other features. In facts it has:

- 2 bits used to enable/disable possible sources for local trigger (bit 0 enables the external source, bit 5 enables the output of the first demultiplexer);
- 1 bit to generate a forced trigger;
- 1 bit to rearm the trigger;
- 1 bit to enable/disable trigger vetoing.

Exploiting these three registers it is possible to obtain different triggering logic schemes; for instance:

- setting SOURCE=3 and MODE="110000", the system will simply trigger on the internal trigger (with veto);
- setting SOURCE=2 and MODE="010001", the system will trigger on the CFDIN trigger (with veto);

The CFDIN trigger (so called for historical reasons) comes from an 8-input ECL connector on the mother board. In our measurement at LNS we have used a third setup, which allows for complex trigger generation; by setting SOURCE=3, MODE="110000" and OUT=0, the following logic is applied:

1. the internal trigger signal (which is few nanoseconds long), is selected by the first multiplexer and then converted into a finite length rectangular pulse (100 ns duration);
2. the 100 ns long pulse is sent outside the board, via the front angular connector;
3. the board triggers on the CFDIN trigger source (trig\_ext pin is not used).

In this way, it is possible to use the rectangular pulse signals produced by the different channels to generate (using an external device) a trigger signal that is then injected in the CFDIN input pins of the mother board. A typical use of such a scheme is the generation of a "telescope" trigger built out of the single triggers coming from the individual detectors, as described in sec. 5.5.







# Chapter 4

## Interpolation of digitized detector signals

Given a function known only at discrete locations (samples), interpolation is a mathematical process that reconstructs its behaviour in the intervals in between samples. This process is useful, for example, when the known values are obtained by sampling an original continuous signal and one wants to reconstruct the original signal, or a close approximation of it, from its samples.

In this chapter we show how the well known Shannon's theorem [22] can be interpreted as a form of interpolation and how interpolation can be used to reconstruct those signals that do not fall under Shannon's theorem constraints. Furthermore, we present a method to apply interpolation in real-time using linear filters, based on some papers by M. Unser [3, 4]. We also illustrate our idea of applying linear operators to an interpolated signal by transforming the so-called *interpolation kernel* instead of the signal.

### 4.1 Interpolation and digital sampling

Sampling is a process through which a continuous function (e.g. an analog signal or a light intensity distribution) is measured at discretely spaced values of the independent variable(s). Considering analog signals, the independent variable is time ( $t$ ) and the sampling process associates a sequence of values  $x[n]$  to the continuous-time signal  $x(t)$  as follows

$$x[n] = x(n\tau_{clk}) \quad \forall n \in \mathbb{Z} \quad (4.1)$$

where  $\tau_{clk}$  is called sampling period and  $\nu_s = \tau_{clk}^{-1}$  is called sampling frequency.

A well-known result of the Sampling Theory, the Shannon's theorem<sup>1</sup>, affirms that, if the Fourier spectrum of the original continuous signal has no frequency components with

---

<sup>1</sup>More commonly known as Nyquist-Shannon sampling theorem.

$\nu > \nu_s/2$  (Nyquist condition), then the expression:

$$x_R(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin\left(\pi\left(\frac{t}{\tau_{clk}} - n\right)\right)}{\pi\left(\frac{t}{\tau_{clk}} - n\right)} \quad (4.2)$$

is a perfect reconstruction of the original function  $x(t)$ . On the other hand, if the signal does not fall under the Nyquist condition, the reconstruction is not perfect anymore. This expression can be rewritten in a more compact form by defining the function  $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ :

$$x_R(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}\left(\frac{t}{\tau_{clk}} - n\right) \quad (4.3)$$

The reconstruction can be described both in term of physical time units:

$$f(t) = \text{sinc}(t/\tau_{clk}) = \frac{\sin\left(\frac{\pi t}{\tau_{clk}}\right)}{\frac{\pi t}{\tau_{clk}}} \quad (4.4)$$

$$x_R(t) = \sum_{n=-\infty}^{\infty} x[n] f(t - n\tau_{clk}) \quad (4.5)$$

and exploiting adimensional normalized time units by defining  $y = t/\tau_{clk}$ :

$$\tilde{f}(y) = \text{sinc}(y) = \frac{\sin(\pi y)}{\pi y} \quad (4.6)$$

$$x_R(y) = \sum_{n=-\infty}^{\infty} x[n] \tilde{f}(y - n) \quad (4.7)$$

As we will show in section 4.2, Shannon's theorem is a particular case of interpolation, where the *sinc* function plays the role of the so-called kernel function.

## 4.2 Kernel spaces, kernel functions and interpolation process

From a general point of view, we can explain the interpolation process by introducing two basic concepts: the kernel space and the kernel function. In the following, we will treat only the interpolation of unidimensional signals (such as the current/charge signals produced by a detector) for simplicity. Given a kernel function  $f(x)$  and a shift value  $\Delta x$ , we can define the kernel space:

$$K(f(x), \Delta x) = \left\{ h(x) = \sum_{n=-\infty}^{+\infty} b_n f(x - n \cdot \Delta x) \right\} \quad (4.8)$$

where  $h(x)$  represents a generic member of the space and  $b_n$  are real coefficients<sup>2</sup>. We can then define a scaled version  $\tilde{f}(x)$  of the kernel function  $f(x)$  so that

$$K(f(x), \Delta x) = \left\{ h(x) = \sum_{n=-\infty}^{+\infty} b_n \tilde{f}(x/\Delta x - n) \quad \text{where } \tilde{f}\left(\frac{x}{\Delta x}\right) = f(x) \right\} \quad (4.9)$$

The function  $f(x)$  is the kernel function correlated to the kernel space  $K$ . By looking back at eq. 4.6 to 4.7, it is now apparent that the reconstructed function proposed by Shannon's theorem is nothing but a member of the Kernel space  $K\left(\text{sinc}\left(\frac{t}{\tau_{clk}}\right), \tau_{clk}\right)$ . In principle, every function can be used as a kernel function. As expected, the properties of kernel space member functions are related to the properties of the associated kernel function. This is particularly true when the action of linear operators over the member functions is considered. For instance, members functions inherit from the kernel function the continuity and derivability. Therefore, the desired properties of the interpolated function can be obtained through a wise choice of the kernel function.

If the Nyquist condition is satisfied, then the *sinc* kernel is the best possible kernel, perfectly reconstructing the original continuous signal. In principle, there is no reason to use a different kernel function. On the other hand, if the original function doesn't match the constraints of Shannon's theorem, the reconstruction cannot be exact, even using the *sinc* kernel. In this case, there is no reason to privilege the *sinc* kernel over other possibilities. As explained in sec. 4.3, other kernels can be tried out, basing on criteria of simplicity, best approximation, etc.

It is then useful to introduce the formalism used in advanced sampling theory and reported, e.g., by Unser [23]. It is a generalization of Shannon's theorem: the interpolation process is viewed as a projection of the original continuous signal  $x(t)$  onto the Kernel space associated to a specific kernel function. From this point of view, Shannon's theorem takes a much simpler form: if the original continuous signal is a member of the kernel space<sup>3</sup>  $K\left(\text{sinc}\left(\frac{t}{\tau_{clk}}\right), \tau_{clk}\right)$ , then its projection on the kernel space will be the signal itself.

The projection can be performed by defining a distance operator  $D(f(t), g(t))$  and then finding the member of the kernel space  $f_K(t)$  which minimizes  $D(x(t), f_K(t))$ . A possible choice for the operator  $D$  is:

$$D(f(t), g(t)) = \int_{-\infty}^{\infty} (f(t) - g(t))^2 dt \quad (4.10)$$

In a system exploiting a sampling ADC, the original continuous signal  $x(t)$  is substituted by a sequence of samples  $x[n]$ . Therefore, we can rely only on the information carried by the samples. We can then define a distance operator  $D(f(t), x[n])$  and find the member of

<sup>2</sup>By looking at eq. 4.8 it is evident that  $K$  is a vectorial space.

<sup>3</sup>It can be proven that the kernel space  $K\left(\text{sinc}\left(\frac{t}{\tau_{clk}}\right), \tau_{clk}\right)$  is a subspace of  $L^2$ , containing functions with bandwidth limited in  $[0, \nu_s]$ . Therefore, requesting the function to be a member of this space is totally equivalent to imposing the Nyquist condition [23].

the kernel space which has minimum distance from the sample sequence  $x[n]$ . A possible choice for the distance operator is

$$D(f(t), x[n]) = \sum_{n=-\infty}^{\infty} (f(n\tau_{clk}) - x[n])^2 \quad (4.11)$$

In discrete-time domain the most natural choice for  $\Delta x$  is the sampling period  $\tau_{clk}$ . This way we obtain

$$f(n\tau_{clk}) = \sum_{m=-\infty}^{\infty} b[m]k(n\tau_{clk} - m\tau_{clk}) = \sum_{m=-\infty}^{\infty} b[m]\tilde{k}(n - m) \quad (4.12)$$

where  $k$  is the kernel function and  $\tilde{k}$  its scaled version. With this assumption, the number of coefficients in the reconstructing function (the degrees of freedom of the system) is equal to the number of sampled values (the constraints), so it is possible to request that  $D(f(t), x[n])$  yields zero. In this case the minimization is performed by imposing that

$$D(f(t), x[n]) = \sum_{n=-\infty}^{\infty} \left( \left( \sum_{m=-\infty}^{\infty} b[m]\tilde{k}(n - m) \right) - x[n] \right)^2 = 0 \quad (4.13)$$

which means solving the linear system

$$x[n] - \sum_{m=-\infty}^{\infty} b_m \tilde{k}(n - m) = 0 \quad \forall n \in \mathbb{Z} \quad (4.14)$$

The solution of this system is a set of coefficients representing a function passing for all the samples, i.e. is nothing else than interpolation in disguise. However, though the interpolated signal is equal to the original signal on the samples, it can differ from it in the intervals in between samples. In other words, there is no guarantee that the original signal is exactly reconstructed. This is at variance with functions satisfying Shannon's theorem, for which reconstruction is exact, even in the intervals in between samples, when using the *sinc* kernel. The demonstration proposed by Unser and exposed in sec. 4.4.1 exploits condition 4.14.

### 4.3 Choosing the right kernel

The choice of the appropriate kernel is crucial to achieve good results. If the Nyquist condition is satisfied, the best kernel is obviously the *sinc*, since it reconstructs the signal exactly. Otherwise, the choice depends on few considerations. The most important constraint is the derivability/continuity properties desired for the interpolated function. However, there are two other items that must be kept in mind:

1. it is better to ensure that the system described in eq. 4.14 is easily solvable;
2. the function evaluation must be easy, from the computational point of view.

To achieve the first point, we adopt a scaled kernel function  $\tilde{k}(x)$  which is non-zero only for a few integer values of  $x$ . From this point of view, it would be very helpful if the kernel function could satisfy the following property:

$$\tilde{k}(n) = \beta \delta_{n,0} \quad \forall n \in \mathbb{Z} \quad (4.15)$$

where  $\delta_{n,m}$  is a Kronecker delta. In this very simple case the system in eq [4.14] has a trivial solution which is  $b[n] = x[n]/\beta$ . The *sinc* function used by Shannon's theorem has this property (with  $\beta=1$ ) and thus the coefficients of the interpolating function are equal to the samples  $x[n]$ . On the other hand, since the *sinc* function extends indefinitely in time, all the coefficients are to be considered, i.e. an infinite sum has to be calculated in order to evaluate the interpolated function at a given time. Even for a modern computer, an exact solution is not possible. Therefore it's common practice to use a kernel function which is non-zero only in a finite interval (usually the smallest compatible with the desired precision). Furthermore, this condition grants a simplification of the system in eq 4.14. An example of a good kernel function is the cubic convolution kernel proposed by Keys [24] whose expression is:

$$\tilde{k}_{Keys}(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}x^2 + 1 & 0 < |x| \leq 1 \\ -\frac{1}{2}|x|^3 + \frac{5}{2}x^2 - 4|x| + 2 & 1 < |x| \leq 2 \\ 0 & |x| > 2 \end{cases} \quad (4.16)$$

This kernel is very easy to deal with. In fact:

- the system in eq 4.14 is easily solved and yields  $b[n] = x[n]$ , like for the *sinc*;
- function reconstruction is easy because  $\tilde{k}(x) = 0$  for  $|x| > 2$ , so that, in order to evaluate  $f(x)$ , only four terms/coefficients are needed.

On the other hand, Key's kernel grants continuity only up to the first derivative. The properties of the two kernel functions described until now, can be viewed in fig. 4.1. By looking at the *sinc* kernel (left plot) it is apparent that, when using this kind of kernel, an infinite number of coefficients is needed to reconstruct the function at a given time. The cubic convolution kernel, shown on the right, is simpler to handle because of its limited domain. Anyway, it is important to remember that the *sinc* kernel is capable of reconstructing the signal exactly, at least if Shannon's hypothesis are met (which unfortunately is not the case for nuclear detector signals).

Other good kernels are the so called B-Spline functions described in the next section. They are also the backbone of the algorithms developed for this work.

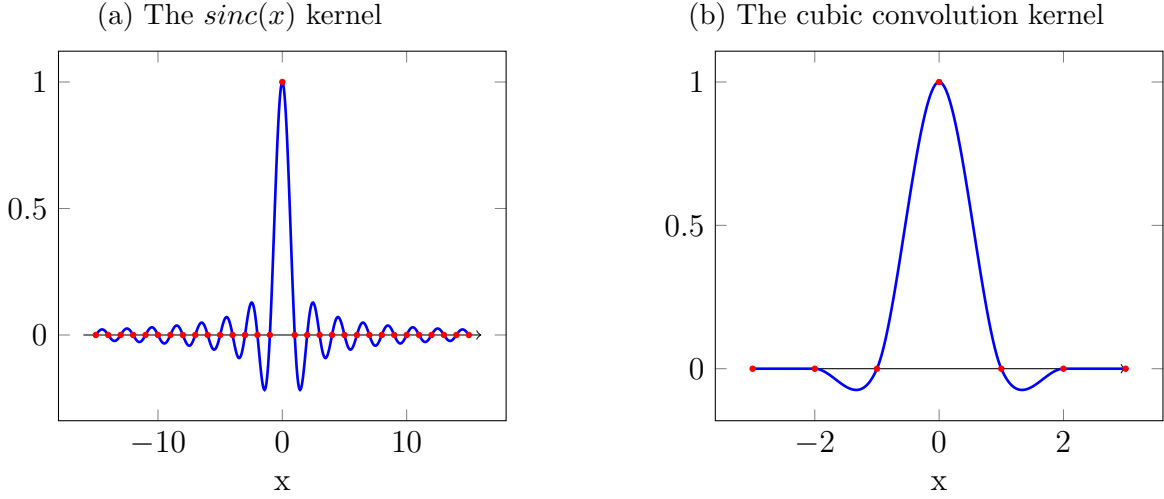


Figure 4.1: The  $\text{sinc}$  kernel (left) given by eq. 4.6 and the cubic convolution kernel (right) given by eq. 4.16.

## 4.4 B-Spline kernels

B-Splines (short form for Box-Splines) are a set of piecewise polynomial functions that are often used for interpolation due to their good properties. They are defined recursively by the following expression:<sup>4</sup>

$$B_0(x) = \begin{cases} 0 & |x| > 1/2 \\ 1 & |x| \leq 1/2 \end{cases} \quad (4.17)$$

$$B_n(x) = (B_{n-1} * B_0)(x) \quad n \neq 0. \quad (4.18)$$

Where  $*$  denotes a convolution product. The B-Splines functions of order 0–3 are plotted in fig. 4.2. Due to the properties of the convolution product it's easy to demonstrate that:

- $B_n(x)$  is continuous up to the  $(n - 1)$ -th order derivative.
- $B_n(x)$  is non-zero only in the interval  $(-\frac{n+1}{2}, \frac{n+1}{2})$ .
- $B_n(x)$  is a piecewise polynomial function in the intervals  $(k - \frac{1}{2}, k + \frac{1}{2})$  if  $n$  is even and  $(k, k + 1)$  if  $n$  is odd (with  $k$  integer).

<sup>4</sup>Here and in the following we will refer to this as the kernel used for interpolation; this functions are actually the **scaled version** of the Kernel, though we omit the  $\sim$  out of simplicity.



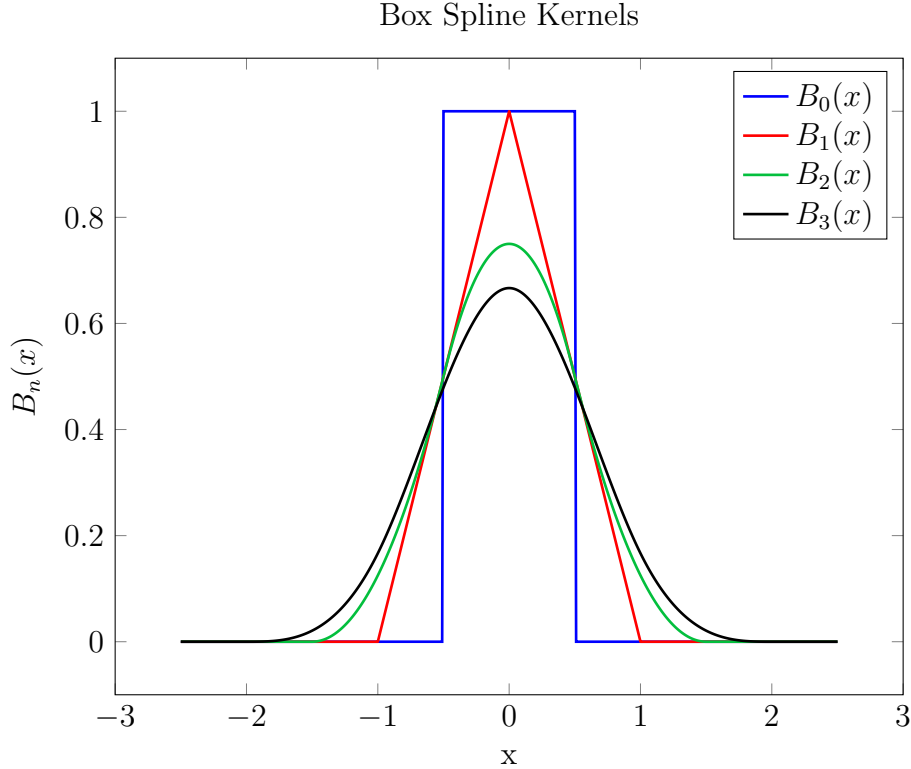


Figure 4.2: The first four B-Spline kernel functions (see legend) plotted in the interval  $(-2.5, 2.5)$ . The Cubic B-Spline kernel, used in our interpolation, is the one in black.

#### 4.4.1 Cubic B-Spline kernel

Based on past experience, when interpolating nuclear detector signals using a piecewise polynomial function, the order of the polynomial should be at least three. In fact, a signal from a charge preamplifier usually presents one inflection point. Since a quadratic polynomial has a constant second derivative, a piecewise second order polynomial function can have inflection points only at the junctions between one polynomial and the other. Third order polynomial have instead a non-constant second derivative, allowing for the presence of inflection points also in the intervals in between the junction points. In principle, it is possible to use higher order polynomials, but as the order goes up, the solution of the system in eq. 4.14 becomes more difficult. For this reason, we have adopted the cubic B-Spline ( $B_3$ ) kernel, which is continuous up to its second derivative. Its explicit expression, derived from equations 4.17 and 4.18 is as follows:

$$B_3(x) = \begin{cases} \frac{(2-|x|)^3}{6} & 1 < |x| \leq 2 \\ \frac{|x|^3}{2} - |x|^2 + \frac{2}{3} & |x| \leq 1 \\ 0 & |x| > 2 \end{cases} \quad (4.19)$$

The  $B_3(x)$  kernel has similar properties with respect to the cubic convolution kernel (which is also a piecewise polynomial) but it grants continuity up to the second order derivative

(Key's cubic convolution grants continuity up to the first order). Its main drawback is the complexity of the linear system which must be solved to get the coefficients, due to the fact that  $B_3(0) = 2/3$  and  $B_3(\pm 1) = 1/6$  and therefore condition 4.15 does not apply. The system thus assumes the following form [25]:

$$\frac{1}{6} \begin{pmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} b[0] \\ b[1] \\ b[2] \\ \vdots \\ b[N-3] \\ b[N-2] \\ b[N-1] \end{pmatrix} = \begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-3] \\ x[N-2] \\ x[N-1] \end{pmatrix} \quad (4.20)$$

whose solution is cumbersome. However, when it comes to signal processing, the problem of cubic B-Spline interpolation has been brilliantly solved by M. Unser [3, 4] who has demonstrated that it is sufficient to apply an IIR linear filter to the signal  $x[n]$  in order to obtain the coefficients of the cubic spline expansion.

We will now follow Unser's demonstration, obtaining the correct form of the IIR filter. First of all, we observe that a system with difference equation

$$x[n] = \frac{1}{6} (b[n-1] + 4b[n] + b[n+1]) \quad (4.21)$$

produces the original signal samples starting from the sequence  $b[n]$ , the sequence of the coefficients of the cubic B-Spline interpolation. This equation, which is a constant coefficients difference equation, implements a linear filter ( $F_{dir}$ ) with impulse response

$$h_{BSdir}[n] = \frac{1}{6} (\delta[n-1] + 4\delta[n] + \delta[n+1]) \quad (4.22)$$

Therefore, the filter  $F_{dir}$ , applied to the sequence of coefficients  $b[n]$ , produces the sequence  $x[n]$  as output. As said in chapter 2, discrete-time LTI filters are characterized by their impulse response function  $h[n]$ . We now make use of the  $Z$ -transform of a sequence, the discrete time equivalent of the Laplace transform. The  $Z$ -transform of a sequence  $h[n]$  is defined as [21]

$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n} \quad (4.23)$$

An LTI filter can be equivalently described either by  $H(z)$  or by  $h[n]$ . The sequence in eq. 4.22 has  $Z$ -transform:

$$H_{dir}(z) = \frac{z^{-1} + 4 + z}{6} \quad (4.24)$$

Considering the filter  $F_{inv}$  which performs the inverse operation of  $F_{dir}$ , actually obtaining the sequence  $b[n]$  starting from the samples  $x[n]$ , and calling  $h_{BSinv}$  its impulse response function, we have that:

$$\begin{aligned} x[n] &= F_{dir}(F_{inv}(x[n])) = \\ &= (h_{BSdir} * (h_{BSinv} * x))[n] = ((h_{BSdir} * h_{BSinv}) * x)[n]. \end{aligned} \quad (4.25)$$

By using the properties of the convolution sum, this equation leads to

$$(h_{BSdir} * h_{BSinv}) = \delta[n] \quad (4.26)$$

which, in terms of the  $Z$ -transforms, becomes  $H_{dir}(z)H_{inv}(z) = 1$  so that

$$H_{inv}(z) = \frac{6}{z^{-1} + 4 + z} \quad (4.27)$$

It is now possible to apply an inverse  $Z$ -transform to this equation, in order to obtain the impulse response function of the desired filter which is

$$h_{BSinv}[n] = \frac{-6\alpha}{1 - \alpha^2} \alpha^{|n|} = \sqrt{3}\alpha^{|n|} \quad (4.28)$$

with  $\alpha = \sqrt{3} - 2$  [3]. Unfortunately, the impulse response function in eq. 4.28 corresponds to an IIR filter which is infinite in both directions of time (and therefore non causal), and can thus be applied in practice only to finite length signals (otherwise, an infinite number of input samples would be needed in order to calculate each output sample).

#### 4.4.2 Approximation of the inverse B-Spline filter

A possible solution to the problem mentioned above is to truncate the impulse response function at a certain point, at the cost of introducing some approximation error. Fig. 4.3 represents the IIR impulse response (eq. 4.28) in the interval  $(-12, 12)$ . By looking at the picture, it is apparent that the convolution sum that represents the response of the filter,  $y[n]$ , to an input  $x[n]$ :

$$y[n] = \sum_{m=-\infty}^{\infty} h_{BSinv}[m]x[n - m] \quad (4.29)$$

is dominated by the terms with small values of  $|m|$ . This consideration, supported by the calculation that will be presented in sec. 4.4.3, suggests us to truncate the IIR impulse response. One then obtains a FIR filter, which can be applied also to infinite length signals (e.g. for real time filtering).

In the present work, we have applied a cut to the response function by defining a threshold value  $A$  and approximating  $h[n] = 0$  if  $|h[n]| < A$ . Due to the fact that this filter is to be implemented on a FPGA, with coefficients expressed in finite precision fixed point arithmetics, a natural choice for the threshold value is the smallest representable value associated with the binary representation chosen for the coefficients. Because of the even parity of the response function and of the monotonic decrease of  $|h[n]|$  when moving away from  $n = 0$ , the cut produces a symmetric function whose expression is:

$$\tilde{h}_{BSinv}[n] = \begin{cases} h[n] & |n| < \tilde{n}(A) \\ 0 & otherwise \end{cases} \quad (4.30)$$

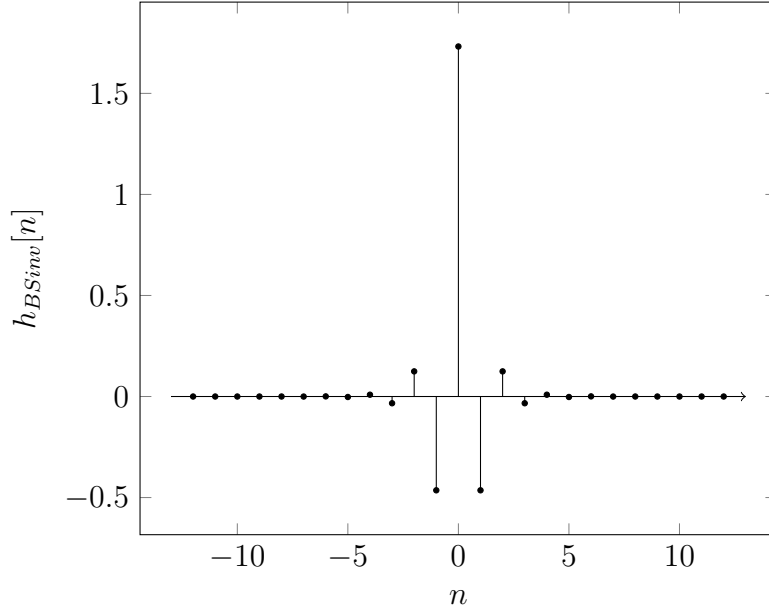


Figure 4.3: The impulse response function of the exact inverse B-Spline filter as given by eq. 4.28.

where  $\tilde{n}(A)$  is the first positive value of  $n$  for which  $|h[n]| < A$ . Additionally, we also ask for the discrete-time integral of the FIR response function to be equal to the integral of its IIR counterpart, which is actually normalized to unity. In fact:

$$\sqrt{3} \sum_{n=-\infty}^{\infty} \alpha^{|n|} = \sqrt{3} \left( 2 \sum_{n=0}^{\infty} \alpha^n - 1 \right) = 1 \quad (4.31)$$

To achieve this, we use the following form for the impulse response function:

$$\hat{h}_{BSinv}[n] = \begin{cases} h[n] & |n| < \tilde{n}(A) - 1 \\ C & |n| = \tilde{n}(A) - 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.32)$$

where  $C = \frac{1 - \sum_{n=-\tilde{n}(A)+2}^{\tilde{n}(A)-2} h[n]}{2}$  is the value needed in order to obtain a unit integral.

As mentioned before, the truncation of the impulse response function introduces an error in the evaluation of the coefficients; this error will then affect the reconstruction of the signal, as explained in the next section.

### 4.4.3 Evaluation of the approximation error

The error committed by approximating the impulse response function can be evaluated in a simple way. The aim of interpolation is to reconstruct the original continuous signal, with the constraint to generate, from a sequence of samples, a continuous function which

passes through all known points. We then ask ourselves how well the known points are reconstructed by the interpolation.

First let's have a more detailed look at the interpolation process. Given a signal  $s[n]$ , a linear filter is applied to it in order to get the sequence  $b[n]$  of the coefficients. From this sequence, the signal is reconstructed both at known points and in between them. To get the values in between known values, one needs also an upsampling stage following the linear filter. However, to obtain just the reconstructed values of the samples at known points, it is enough to apply the filter with the impulse response described in eq. 4.21 to the sequence of the coefficients.

Due to the linearity of the two filters, the operation is equivalent to applying a single filter with impulse response given by the convolution of the two impulse responses:

$$h_{tot}[n] = (h_{BSdir} * h_{BSinv})[n] \quad (4.33)$$

where  $h_{BSinv}$  can be either the exact filter response or the truncated one.

According to eq. 4.26, if the exact filter response is used the result is  $h_{tot}[n] = \delta[n]$ , i.e. the response of the identity system.

In the other case, since  $h_{BSdir}[n]$  is non-zero only for  $n$  in the interval  $[-1, 1]$ , we can simplify the expression of the convolution sum. Furthermore, due to the evenness of both response functions, it is enough to evaluate the convolution in the  $n \geq 0$  region. So, by explicitly writing the convolution sum we obtain:

$$\hat{h}_{tot}[n] = \sum_{k=-1}^1 h_{BSdir}[k] \hat{h}_{BSinv}[n-k] = \frac{1}{6} (\hat{h}_{BSinv}[n-1] + 4\hat{h}_{BSinv}[n] + \hat{h}_{BSinv}[n+1]) \quad (4.34)$$

with  $n \geq 0$ .

We notice the following:

- as long as  $n+1 < \tilde{n}(A) - 1$ ,  $h_{BSinv}$  and  $\hat{h}_{BSinv}$  assume the same values in the convolution sum, so the result will be the same;
- if  $n-1 > \tilde{n}(A) - 1$ ,  $\hat{h}_{BSinv}$  is always zero in the region of summation, so the result is  $\hat{h}_{tot}[n] = 0 = h_{tot}[n]$ .

So, we only need to evaluate  $\hat{h}_{BSinv}[n]$  in the interval  $[\tilde{n}(A) - 2, \tilde{n}(A)]$  and this can be done by explicitly writing the sum. The complete result is:

$$\hat{h}_{tot}[n] = \begin{cases} 0 & |n| > \tilde{n}(A) \\ \frac{1}{6}C & |n| = \tilde{n}(A) \\ \frac{1}{6}(\sqrt{3}\alpha^{\tilde{n}(A)-2} + 4C) & |n| = \tilde{n}(A) - 1 \\ \frac{1}{6}(\sqrt{3}\alpha^{\tilde{n}(A)-3} + 4\sqrt{3}\alpha^{\tilde{n}(A)-2} + C) & |n| = \tilde{n}(A) - 2 \\ \delta[n] & |n| < \tilde{n}(A) - 2 \end{cases} \quad (4.35)$$

If we now define the error  $e[n]$  as the difference between the reconstructed signal and the original signal we get:

$$e[n] = (\hat{h}_{tot} * s)[n] - s[n] = ((\hat{h}_{tot} - \delta) * s)[n] \quad (4.36)$$

where

$$\hat{h}_{tot}[n] - \delta[n] = \begin{cases} \frac{1}{6}C & |n| = \tilde{n}(A) \\ \frac{1}{6}(\sqrt{3}\alpha^{\tilde{n}(A)-2} + 4C) & |n| = \tilde{n}(A) - 1 \\ \frac{1}{6}(\sqrt{3}\alpha^{\tilde{n}(A)-3} + 4\sqrt{3}\alpha^{\tilde{n}(A)-2} + C) & |n| = \tilde{n}(A) - 2 \\ 0 & \text{otherwise} \end{cases} \quad (4.37)$$

In our implementation, the coefficients of the inverse B-Spline filter are multiplied by a factor  $2^{14}$  and represented in a 16 bit signed integer notation. So that the minimum representable number is 1, corresponding to an original value of the coefficient of  $2^{-14}$ . Assuming then  $A = 2^{-14}$ , the index  $\tilde{n}(A)$  can be found by solving

$$\sqrt{3}|\alpha|^t = A \implies t = \frac{\log\left(\frac{A}{\sqrt{3}}\right)}{\log(|\alpha|)} = 7.7856 \quad (4.38)$$

thus obtaining  $\tilde{n} = 8$  after approximating to the next integer value.

Knowing  $\tilde{n}$ , we can calculate the value of  $\hat{h}_{tot}[n] - \delta[n]$ , which turns out to be:

$$\hat{h}_{tot}[n] - \delta[n] = \begin{cases} -2.26 * 10^{-5} & |n| = 8 \\ 1.68 * 10^{-5} & |n| = 7 \\ 6.13 * 10^{-6} & |n| = 6 \\ 0 & \text{otherwise} \end{cases} \quad (4.39)$$

It is possible the sequence  $e[n]$  of the approximation errors just by convolution of  $\hat{h}_{tot}[n] - \delta[n]$  with the original signal. The maximum approximation error is then given by the maximum absolute value of  $e[n]$ , and will depend linearly on the amplitude of the input signal. Since we are dealing with step signals, it can be demonstrated that the worst case is that of a sudden step, having the maximum possible amplitude that, in the present case is  $2^{14}$  Least Significant Bits (LSBs). For such a signal, we have obtained that the maximum approximation error is about 0.4 LSB and, since our digitized signals have a noise with a standard deviation of few LSBs, this contribution is negligible and will then be neglected in the following.

#### 4.4.4 B-Splines interpolation: a graphical description

To better explain the interpolation process, we refer to fig. 4.4. The top panel (a) shows an analytically generated charge signal  $c(t)$ . The signal is generated starting from an analytical current signal of the form

$$i(t) = \frac{A}{N} \left( 1 - \left( 1 + \frac{t}{t_1} \right) e^{-\frac{t}{t_1}} \right) e^{-\frac{t}{t_2}} \quad (4.40)$$

where  $A$  represents the integral of the current signal and  $N$  is a normalization factor<sup>5</sup>. The current signal has been integrated analytically also introducing a decay time  $t_3$  to obtain

<sup>5</sup>This analytical shape is obtained by requesting that the current signal presents a rising edge, an exponential falling edge and imposing continuity up to the first derivative. We start from the analytical expression  $i(t) = f(t)e^{-t/t_2}$  for  $t \geq 0$  and  $i(t) = 0$  for  $t < 0$ . Then we assume that  $f(t) = 1 - (1 + t/t_1)e^{-t/t_1}$ . With these two assumptions one obtains eq. 4.40.

a charge signal  $c(t)$ . The charge signal is then sampled (with a simulated ideal ADC) and white gaussian noise with a standard deviation of 5 units is added to the sampled charge signal  $c[n]$  in order to simulate the ADC input stage noise contribution (which is supposed to be dominant).

The signal  $c[n]$  is passed through the approximated inverse B-Spline filter described in eq. 4.32 in order to obtain the sequence of coefficients  $b[n]$ . The original continuous charge signal is reconstructed by using the formula given in eq. 4.9:

$$s(t) = \sum_{n=-\infty}^{\infty} b[n] B_3(t/\tau_{clk} - n) \quad (4.41)$$

and plotted in panel (b) as a black line. In the picture, the single contributions to the sum are also shown (in blue).

In panel (c) the reconstructed signal  $s(t)$  is plotted over the original samples  $c[n]$ , just to show that the interpolation process yields the desired result: in fact, we have obtained a continuous signal which passes through all the known points (samples) of the original continuous signal  $c(t)$ .

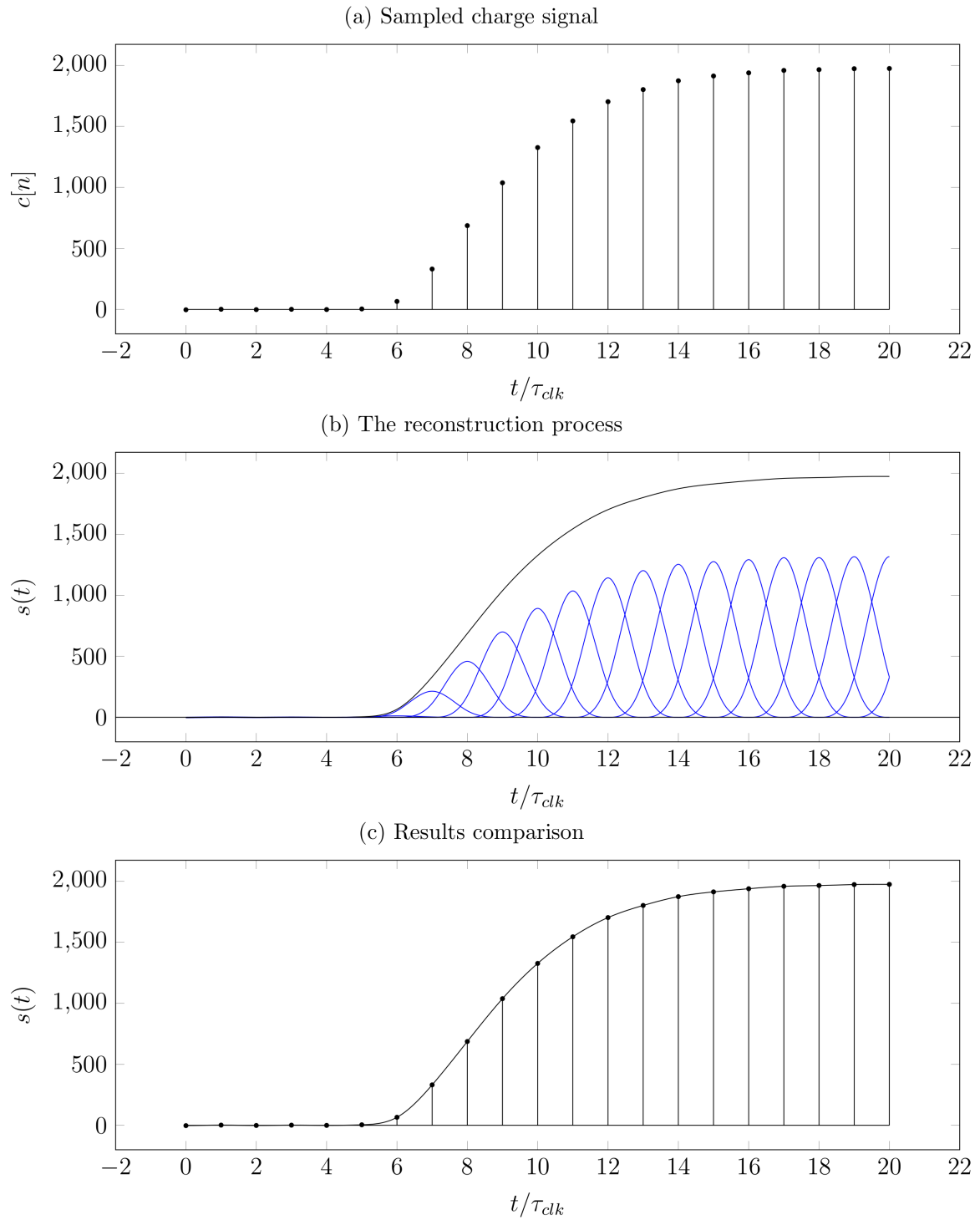


Figure 4.4: The interpolation process applied to a sampled charge signal



## 4.5 B-Spline interpolation and LTI filters

Having introduced the interpolation process and knowing how to obtain (in real time) the set of cubic B-Spline coefficients that represent a signal, we can finally come to the main aim of this work.

The sets of coefficients  $b[n]$  corresponding to the sampled signal  $s[n]$  is a representation of a continuous function:

$$f(t) = \sum_{n=-\infty}^{\infty} b[n]B_3(x - n) \quad (4.42)$$

where  $x = t/\tau_{clk}$ . The output  $g(t)$  of a LTI system with input  $f(t)$  is the convolution product of  $f(t)$  with the system's response  $h(t)$ , namely:

$$\begin{aligned} g(t) &= (h * f)(t) = \int_{-\infty}^{\infty} h(T)f(t - T)dT = \\ &= \sum_{n=-\infty}^{\infty} b[n] \int_{-\infty}^{\infty} h(T)B_3(x - n - T/\tau_{clk})dT = \sum_{n=-\infty}^{\infty} b[n]K(x - n) \end{aligned} \quad (4.43)$$

where the time invariance of the applied filter has been exploited and we have defined:

$$K(x) = (h * B_3)(x) = \int_{-\infty}^{\infty} h(T)B_3(x - T/\tau_{clk})dT \quad (4.44)$$

It follows that, once the coefficients  $b[n]$  are known, the filtered version of the signal  $f(t)$  can be obtained by simply changing the kernel function, i.e. there is no need to recalculate the coefficients. In fact, the new kernel function  $K(t)$  depends only on the filter to be applied and it can be evaluated analytically once and for all. Obviously, if simple interpolation is wanted, then  $h[n] = \delta[n]$  and  $K(x)$  is simply the cubic B-Spline kernel itself.

### 4.5.1 Obtaining the derivative of the signal

In nuclear physics experiments, the current signal produced by a detector is usually integrated by means of a charge preamplifier, and no detailed information about its shape is extracted. The information about the overall duration of the current pulse can then be recovered from the rise-time of the charge signal and used, e.g., for PSA. Sometimes, however, a more detailed knowledge of the shape of the current signal, e.g., its maximum value, would be a better PSA parameter [34]. In principle it should be enough to differentiate the charge signal  $c(t)$  to obtain the current signal  $i(t)$ .

In the particular case in which the charge signal is digitized, i.e. we have a sequence of samples  $c[n] = c(n\tau_{clk})$ , we could apply a “discrete derivative” (also known as “first backward difference”), to reconstruct the current from the charge signal

$$i_D[n] = c[n] - c[n - 1] = c(n\tau_{clk}) - c(n\tau_{clk} - \tau_{clk}) \quad (4.45)$$

$i_D$  represents the average first derivative of the signal in the interval  $((n - 1)\tau_{clk}, n\tau_{clk})$  multiplied by a factor  $\tau_{clk}$ . However, this averaging affects the reconstruction of the “real”

current signal especially when its first derivative varies significantly on a time scale much shorter than the sampling period (e.g. when  $\left| \tau_{clk} \frac{d^2 c(t)}{dt^2} \right| \gg 1$ ). In order to overcome this problem, we can exploit the cubic B-Spline interpolation presented in sections 4.4 and 4.5. Given the sampled charge signal  $c[n]$ , we can obtain the sequence of cubic B-Splines coefficients,  $b[n]$ , exploiting the linear filter described in section 4.4.1 and then have a continuous function which is an approximation of the original charge signal:

$$c_{rec}(t) = \sum_{n=-\infty}^{\infty} b[n] B_3(t/\tau_{clk} - n) \quad (4.46)$$

According to the idea explained above, it is then possible to obtain the derivative of  $c_{rec}(t)$  (in the continuous-time domain) as:

$$i_{rec}(t) = \sum_{n=-\infty}^{\infty} b[n] b3(t/\tau_{clk} - n) \quad (4.47)$$

where  $b3(x)$  is the derivative of the cubic B-Spline Kernel (see fig. 4.5):

$$b3(x) = \begin{cases} 2 + 2x + \frac{x^2}{2} & -2 \leq x < -1 \\ -\frac{3}{2}x^2 - 2x & -1 \leq x < 0 \\ \frac{3}{2}x^2 - 2x & 0 \leq x \leq 1 \\ -1 + 2x - \frac{x^2}{2} & 1 < x \leq 2 \\ 0 & |x| > 2 \end{cases} \quad (4.48)$$

Actually, the exact derivative of the charge signal in eq. 4.46 is

$$i_{rec}(t) = \frac{1}{\tau_{clk}} \sum_{n=-\infty}^{\infty} b[n] b3(t/\tau_{clk} - n) \quad (4.49)$$

though the term  $1/\tau_{clk}$  will be ignored from now on.

The signal  $i_{rec}(t)$  is continuous and so, in order to obtain from it a sampled current signal with the same sampling period as the original charge signal, we must evaluate it at the sampling points. The result is:

$$i_{rec}[n] = \sum_{m=-\infty}^{\infty} b[m] b3(n - m) = \frac{1}{2} \sum_{m=-\infty}^{\infty} b[m] (\delta[n - m + 1] - \delta[n - m - 1]) \quad (4.50)$$

By inspecting the result we notice that it can be interpreted as the result of an LTI filter, with FIR  $h[n] = \frac{1}{2}(\delta[n+1] - \delta[n-1])$ , applied to the sequence of coefficients  $b[n]$  associated with the charge signal. Even more important for our application is that, since we can calculate the continuous function at any time, we can also sample it at a frequency higher than the sampling frequency of the charge signal, thus obtaining an upsampled version of the current signal. The latter could give better results, especially when the aim of the experiment is to measure the maximum value of the current signal (see sec. 4.6).

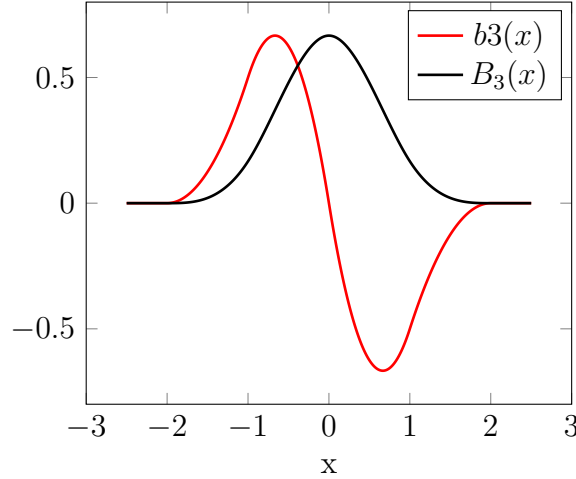


Figure 4.5: The cubic B-Spline kernel (in black) and it's first derivative (in red)

### 4.5.2 Correcting for the preamplifier decay time

In our experimental setup, the charge preamplifier produces an exponentially decaying output signal. The effect of the decay constant on the energy measurement has been described already in sec. 2.1. However, the decay introduces some unwanted effects also when evaluating the current, as the digital current signal can't be obtained anymore as the derivative of the charge signal. As suggested in section 2.1, the effects of the decay constant can be corrected for by using a pole-zero correction filter. This filter is characterized (in the continuous-time domain) by the impulse response function.

$$h_{PZC}(t; \tau) = \delta(t) + \frac{1}{\tau}\theta(t) \quad (4.51)$$

Applying this response function to the cubic B-Spline kernel (actually to the scaled kernel function) yields the following result:

$$\begin{aligned} B3_{PZC}(x; \tau, \tau_{clk}) &= \int_{-\infty}^{\infty} h_{PZC}(\eta; \tau) B_3(x - \eta/\tau_{clk}) d\eta = \\ &= B_3(x) + \frac{1}{\tau} \int_{-\infty}^{\infty} B_3(\eta/\tau_{clk}) \theta(x\tau_{clk} - \eta) d\eta \end{aligned} \quad (4.52)$$

where  $x = t/\tau_{clk}$  is the time measured in adimensional units. By exploiting the limited domain of the integrand functions, the integral can be simplified as follows:

$$B3_{PZC}(x; \tau, \tau_{clk}) = B_3(x) + \frac{1}{\tau} \int_{-\infty}^{x\tau_{clk}} B_3(\eta/\tau_{clk}) d\eta = B_3(x) + \frac{\tau_{clk}}{\tau} \int_{-\infty}^x B_3(\xi) d\xi \quad (4.53)$$

Using this new kernel to reconstruct the signal with the cubic B-Spline coefficients  $b[n]$ , one gets as a result the charge signal already corrected for the preamplifier decay. In order to obtain the current signal, we must use the kernel obtained by derivation from eq. 4.53

$$B3_{PZC;DER}(x; \tau_{clk}/\tau) = b3(x) + \frac{\tau_{clk}}{\tau} B_3(x) \quad (4.54)$$

where we have again ignored the multiplicative factor  $1/\tau_{clk}$ . This kernel has a shape that depends on the ratio  $\tau_{clk}/\tau$ , so it must be calculated according to the preamplifier decay time and the sampling period. Different shapes of the kernel obtained with different values of  $\frac{\tau_{clk}}{\tau}$  are plotted in fig. 4.6. This filter has been implemented inside the FPGA firmware. More details about the implementation are given in sec. 5.4. The reconstruction process

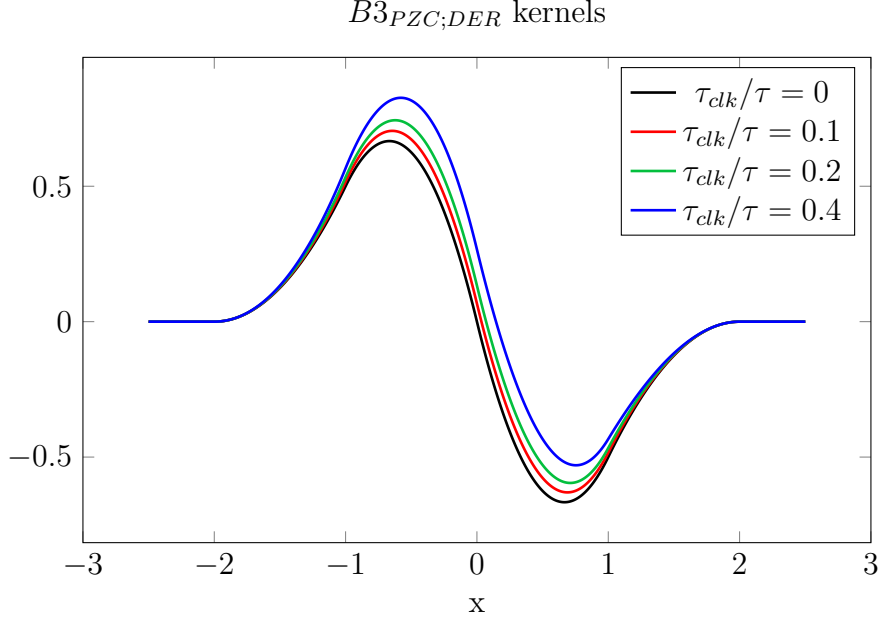


Figure 4.6: The  $B3_{PZC;DER}$  kernel for some values of  $\tau_{clk}/\tau$ .

is shown in picture 4.7. The method used for the simulation is the same as described in sec. 4.4.4. The only difference is that, instead of the  $B_3(x)$  kernel, we have used the  $B3_{PZC;DER}(x)$ , thus obtaining the derivative of the charge signal (including PZC). Again, panel (b) shows the reconstructed signal (in black) and the single contributions to the sum. In panel (c), the reconstructed signal is plotted (in black) with the original current signal, given by eq. 4.40, in red. A simpler process, with respect to our interpolation, is the first backward difference (discrete derivative) with discrete-time PZC. Such a process, applied to the sampled charge signal, would give the signal reported as black dots in panel c). It is apparent that differentiating via interpolation we obtain a better reconstruction of the current signal maximum, with respect to differentiating in the discrete time domain without interpolation.

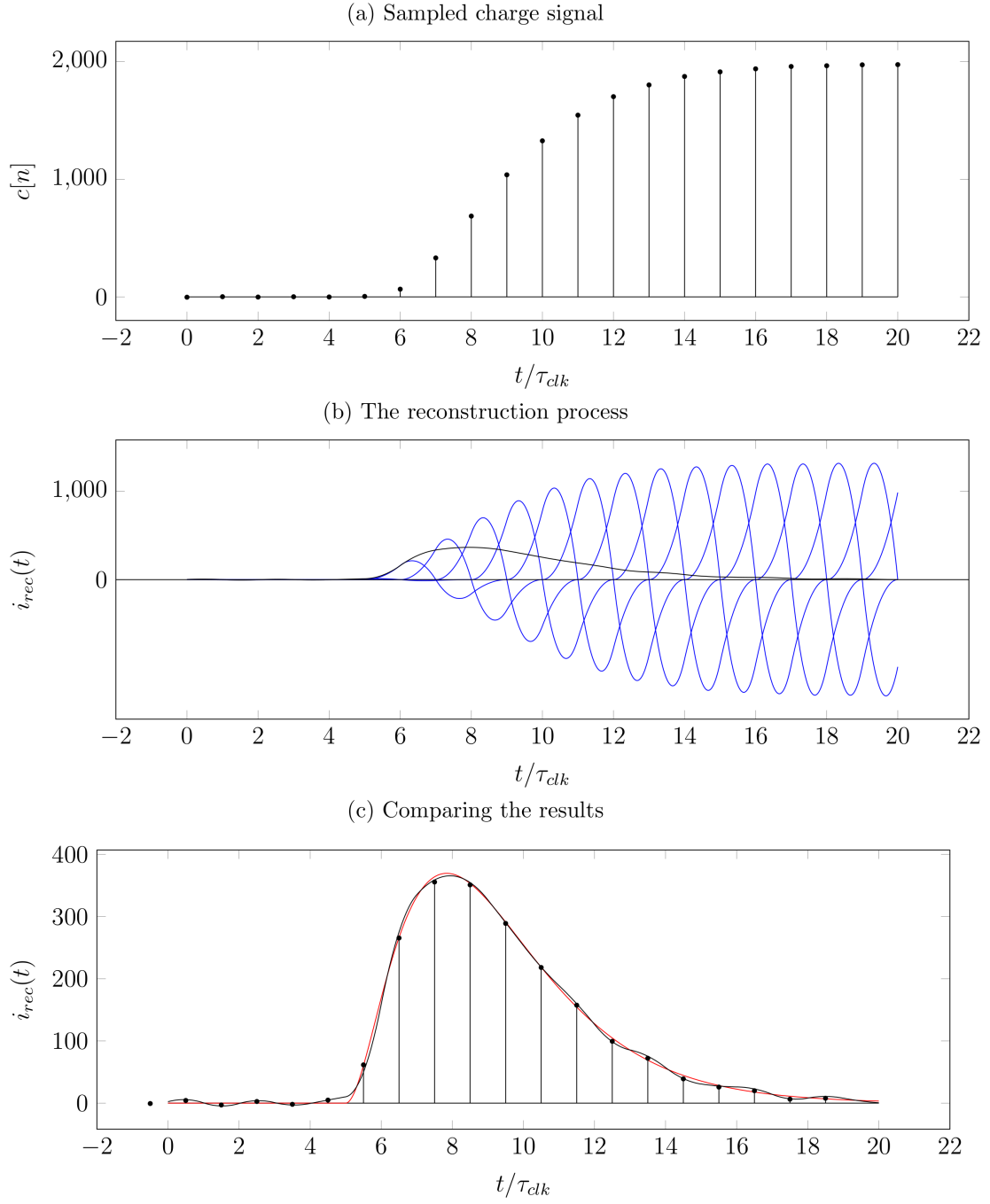


Figure 4.7: The derivation process of a charge signal, as performed using B-Splines derivative with PZC, compared to a first backward difference (with PZC).

## 4.6 Evaluating the maximum of the current signal

The current signal is obtained by applying the  $B3_{PZC;DER}$  kernel to the sequence of coefficients  $b[n]$  to reconstruct the signal. For real-time PSA applications, one then needs to develop a real-time algorithm in order to detect the maximum of the reconstructed signal. Usually, to get the best identification performance, it is not enough to reconstruct the current signal with a sampling period equal to that of the charge signal, so we need to sample the continuous signal given by:

$$i_{rec}(t) = \sum_{n=-\infty}^{\infty} b[n] B3_{PZC;DER}(t/\tau_{clk} - n) \quad (4.55)$$

at a higher frequency. By introducing an upsampling factor of  $M$ , this means evaluating  $i_{rec}$  at points  $t = n\tau_{clk}/M$ , thus producing  $M - 1$  new samples for each sample of the charge signal, one obtains the following set of samples:

$$i_{rec}[n] = \sum_{m=-\infty}^{\infty} b[m] B3_{PZC;DER}\left(\frac{n}{M} - m\right) \quad (4.56)$$

If we consider the group of  $M$  samples of  $i_{rec}$  falling between a sample (included) and the next sample (not included) of the charge signal (in particular  $c[n]$  and  $c[n + 1]$ ) we have that:

$$i_{rec}[Mn + m] = \sum_{k=n-2}^{n+1} b[k] B3_{PZC;DER}\left(n + \frac{m}{M} - k\right) = (b * h_m)[n] \quad (4.57)$$

where

$$h_m[n] = B3_{PZC;DER}\left(n + \frac{m}{M}\right) \quad 0 \leq m < M \quad (4.58)$$

This last equation is the basis for a real time implementation of the process using a set of linear filters. With all these considerations, we can now describe the implementation of the algorithm:

- the sampled charge signal  $c[n]$  is passed through a LTI filter with impulse response equal to  $\hat{h}_{BSinv}[n]$  (eq. 4.28) and a sequence  $b[n]$  of approximated cubic B-Spline coefficients is obtained;
- the sequence  $b[n]$  is fed in parallel to  $M$  different LTI filters with impulse responses equal to  $h_m[n]$  (eq. 4.58), each filter with a different  $m$ , from zero to  $M - 1$ ;
- for each sample of the original charge signal  $c[n]$ ,  $M$  samples of the reconstructed current signal are generated.

With the signal upsampled by a factor of  $M$ , the problem of managing the data arises. In fact, for each clock period of the FPGA device,  $M$  samples are produced. To treat them in real time one at a time, part of the device should be able to run at  $M$  times the standard clock (120 MHz). In practice, the problem can be solved by using a pipelined algorithm, as shown later on in sec. 5.4.

## 4.7 Dealing with noise and other problems

Since the maximum of the reconstructed signal is used as a shape parameter in order to identify different nuclei, we need to achieve a good precision so to be able to identify also different isotopes of the same element. The precision of the measure is, in principle, limited by three concurrent independent factors:

- the error committed when taking the maximum amongst a finite set of sampled values;
- the accuracy of the signal reconstruction;
- the presence of noise over the charge signal: this noise will be filtered and it will still be present on the final current signal.

In the rest of the chapter we will discuss these points.

### 4.7.1 Maximum detection

In our case, the maximum of the signal is extracted among a set of samples, instead of being detected on an analog continuous signal. Since the information about its behaviour in between samples is not available anymore, most of the times the actual maximum is lost because it falls in between two samples. Therefore, there will be two samples, one before and one after the occurrence of the maximum, both smaller than the actual maximum value. Not considering the possible influence of noise fluctuations, one of these two values will be selected as the maximum value by the processing system. With this hypothesis, it is possible to obtain the distribution of the measured maximum value using a simple model. We assume that:

- the analog signal has a parabolic shape in the region around the maximum;
- the noise is negligible;
- the sampling process is ideal, with sampling period  $\tau_{clk}$ ;
- the parabolic shape is characterized by a second derivative  $-\frac{K}{\tau_{clk}^2}$ , where  $K$  is a positive parameter;
- the phase shift between the signal and the sampling clock is random (uniformly distributed in  $[-\tau_{clk}/2, \tau_{clk}/2]$ );

If we call  $M$  the maximum of the analog signal and assume that  $t = 0$  when the maximum occurs, we can approximate the signal shape around the maximum using a second order polynomial, actually:

$$s(t) \approx M - \frac{K}{2} \left( \frac{t}{\tau_{clk}} \right)^2. \quad (4.59)$$

Due to the fact that the two samples closest to the maximum are separated by a time interval  $\tau_{clk}$ , if we define  $s_a = s(t_a)$  and  $s_b = s(t_a - \tau_{clk})$  the samples immediately after and before the maximum, we get:

$$t_a \in [0, \tau_{clk}]$$

$$m(t_a) = \max(s(t_a), s(t_a - \tau_{clk})) = \begin{cases} M - \frac{K}{2\tau_{clk}^2}(t_a - t_s)^2 & t_a \in [t_s/2, t_s] \\ M - \frac{K}{2\tau_{clk}^2}t_a^2 & t_a \in [0, t_s/2] \end{cases} \quad (4.60)$$

Where  $m$  is the bigger of the two samples  $s_a$  and  $s_b$ . We can now derive the distribution of the maximum value obtained ( $\hat{M}$ ). By definition:

$$P(\hat{M})d\hat{M} = P(m(t_a) \in (\hat{M}, \hat{M} + d\hat{M})) \quad (4.61)$$

If we take into account the interval  $I$  given by:

$$I = \{t_a \mid m(t_a) \in (\hat{M}, \hat{M} + d\hat{M})\} \quad (4.62)$$

the probability will be  $P(\hat{M})d\hat{M} = \frac{dI}{\tau_{clk}}$  (where  $dI$  is the width of the interval  $I$ ), due to the uniform distribution of  $t_a$  in  $[0, \tau_{clk}]$ . As it can be seen in fig. 4.8, due to the symmetry of the function  $m(t_a)$  around  $t = \tau_{clk}/2$ , the interval  $I$  is the union of two identical intervals  $I_1$  and  $I_2$  so that:

$$P(\hat{M})d\hat{M} = \frac{2dI_1}{\tau_{clk}} \quad (4.63)$$

$dI_1$  can be evaluated as:

$$dI_1 = d\hat{M} \frac{1}{\left| \frac{dm}{dt_a} \right|_{t_a(\hat{M})}} = d\hat{M} \frac{\tau_{clk}}{\sqrt{2K(M - \hat{M})}} \quad (4.64)$$

and we obtain:

$$P(\hat{M})d\hat{M} = \begin{cases} \frac{1}{\sqrt{\frac{K}{2}(M - \hat{M})}} d\hat{M} & M - \frac{K}{8} < m < M \\ 0 & \text{otherwise} \end{cases} \quad (4.65)$$

Now it is easy to evaluate the expectation values of  $\hat{M}$  and  $\hat{M}^2$ , which turn out to be:

$$E\{\hat{M}\} = M - \frac{K}{24} \quad (4.66)$$

$$E\{\hat{M}^2\} = M^2 - M \frac{K}{12} + \frac{K^2}{320} \quad (4.67)$$

$$(4.68)$$

The standard deviation of the distribution is given by:

$$\sigma_{\hat{M}} = \sqrt{(E\{(\hat{M} - E\{\hat{M}\})^2\})} = \sqrt{E\{\hat{M}^2\} - (E\{\hat{M}\})^2} = \frac{K}{\sqrt{720}} \quad (4.69)$$



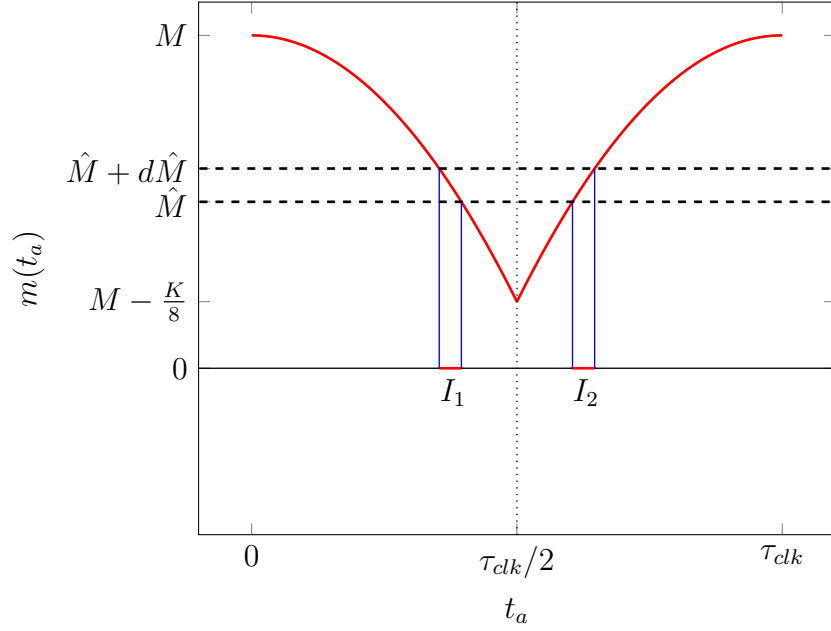


Figure 4.8: A graphical outline of the model described in sec. 4.7.1

Looking back at our assumptions, if  $s(t)$  is our generic signal, then

$$K = - \left( \frac{d^2 s(t)}{dt^2} \right)_{t=t_{max}} \cdot \tau_{clk}^2, \quad (4.70)$$

is a quantity related to the signal's first derivative rate of change on the time scale of the sampling period. In order to minimize  $K$ , and therefore  $\sigma_{\hat{M}}$ , one can use higher sampling frequencies (thus reducing  $\tau_{clk}$ ), or slow down the signal, reducing  $\frac{d^2 s(t)}{dt^2}$ . It is important to remember that we are not sampling an analog signal. What we are doing is evaluating the shape of an analog signal and then sampling it, thus the sampling frequency can be chosen arbitrarily (in principle) and it is not dependent on the charge signal sampling clock. In practice, in order to implement the process on a FPGA, the frequency at which the reconstructed signal is sampled must be an integer multiple of the charge signal sampling frequency. We have used an upsampling factor of 10, thus sampling the reconstructed signal at a frequency ten times higher than the charge signal's one. This result implies that if the signal is fast (with respect to its sampling period) the reconstructed value will be affected by an additional contribution to noise fluctuations due to the sampling process, so it is better to work with slower signals and to increase the reconstructed signal sampling frequency (i.e. increase either the upsampling factor or the charge signal sampling frequency) so that the error introduced in eq. 4.69 is negligible with respect to other contributions.

#### 4.7.2 Signal reconstruction: the B-Spline problem

The constraint used to derive the interpolation coefficients, forces the reconstructed function to pass through the samples. It does not introduce any constraints to the behaviour of

this function in the intervals between samples. Furthermore, requiring the second derivative to be continue, introduces a problem when dealing with rapid variations of the signal. For example, if the signal presents a sudden step (as shown in fig. 4.9), the reconstructed signal presents an undershoot and an overshoot (with possible ringing). The undershoot is needed to connect the flat part preceding the step with the rising edge, at the same time maintaining the continuity of the second derivative. The overshoot happens at the end of the rising edge for the same reason. This is a typical issue when applying interpolation and it comes from the simple fact that it is impossible to reproduce a discontinuity (such as a sudden step) when we are at the same time requesting continuity up to the second order derivative. This can be seen as a sort of Gibbs effect for interpolation [21].

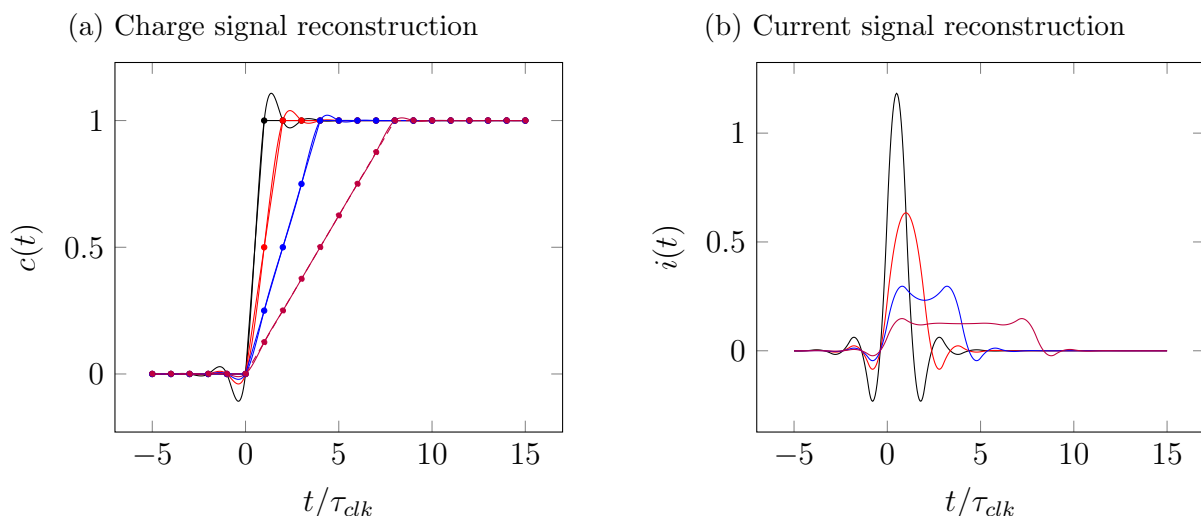


Figure 4.9: A graphical overview of the overshoot problem (see sec. 4.7.2)

In our work the current signal is obtained by differentiating the interpolated charge signal: the undershoot (overshoot) is differentiated as well, giving a negative (positive) lobe before (after) the current signal. This last assumption can be verified with a simple simulation, reported in fig. 4.9. Four different charge signals, with different rise-times and linear rising edges, are reconstructed using the filtering techniques described in sections 4.4.2 (simple interpolation) and 4.5.2 (current signal reconstruction). In the left panel, the samples are plotted together with the interpolated charge signals, showing a reduction in the undershoot/overshoot amplitude for slower signals. In the right panel, the corresponding current signals are plotted. Since the charge signals rising edge are linear, the exact current signals should appear as rectangular pulses. Again, the overall shape of the reconstructed current signal becomes more rectangle-like for slower signals, indicating a better reconstruction. As for the previous problem, this one can be reduced by working with charge signals which are slower with respect to the sampling frequency, thus reducing the entity of the undershoot and the overshoot (eventually eliminating it). Again, this can be obtained either by slowing down the signals and by increasing the sampling ADC frequency

### 4.7.3 The effects of noise on the detection of the maximum

The filtering applied to the signal also modifies its SNR. The actual change in SNR will be approached through a numerical simulation in sec. 6.3. The presence of noise can be particularly disturbing when applying cubic B-Spline interpolation to the signal. As reported in sec. 4.7.2, the constraint of continuity of the second derivative produces an oscillating signal, as can be seen, for example, in fig. 4.7 panel c. In addition, the differentiation process itself tends to reduce the SNR of the signal. To understand this last statement, let's consider a simplified model. Assuming that:

- the signals (both charge and current) must fit into the ADC range  $R$ ;
- the sampling process adds a component of white noise to the signal;
- the overall noise is dominated by the ADC input stage, i.e. it is white, with a constant spectral density  $\omega_n$  in the frequency interval  $[0, \nu_s/2]$ ;

With these hypotheses the standard deviation of the noise present on the signal is  $\sigma_n = \frac{1}{\sqrt{2}}\sqrt{\nu_s\omega_n}$ . Let's take, as an example, the case in which the charge signal has maximum amplitude and the shorter risetime compatible with the antialiasing low-pass filter. Since in this case both the charge signal and the associated current signal have their maximum values, we assume that the gain of the respective digitizers (assuming to digitize the two signals separately) is adjusted so that both the charge and current signals completely fill the ADC range.

Applying a first backward difference filter to the charge signal, we obtain a reconstruction of the current signal. This process is obviously different by the one used in this thesis, though performing a similar operation. The present choice has the advantage of requiring simpler calculations: further studies will be made in order to better understand the limits of the actually used algorithm. The first backward difference filter is characterized by a transfer function:

$$A(\omega) = 1 - e^{-j\omega} \quad (4.71)$$

where  $\omega$  is the frequency in the discrete-time domain, whose expression in terms of the continuous-time domain frequency  $\Omega$  is :  $\omega = 2\pi\frac{\Omega}{\nu_s}$ . Evaluating the variance of noise fluctuations after this filter yields:

$$\begin{aligned} \sigma_f^2 &= \int_0^{\frac{\nu_s}{2}} |A(\omega)|^2 \omega_n d\Omega = 2\omega_n \int_0^{\frac{\nu_s}{2}} (1 - \cos\omega) d\Omega = \\ &= \frac{\nu_s}{\pi} \omega_n \int_0^{\pi} (1 - \cos\omega) d\omega = \nu_s \omega_n \end{aligned} \quad (4.72)$$

and so  $\sigma_f = \sqrt{\nu_s\omega_n} = \sqrt{2}\sigma_n$ . Therefore, a reconstructed current signal is noisier than a current signal directly sampled by its own ADC. Furthermore, by considering an event for which we obtain the maximum energy and the minimum rise-time (e.g. maximum current) we observe that:

- the maximum current obtained from the sampled current signal ( $I_{real}$ ) has value  $R$ ;

- the maximum obtained by the reconstructed signal ( $I_{rec}$ ) is smaller than  $R$ ;

so that the SNR for the two signals will be:

$$\begin{aligned} SNR_{real} &= \frac{R}{\sigma_n} \\ SNR_{rec} &< \frac{R}{\sqrt{2}\sigma_n} \end{aligned} \quad (4.73)$$

and the ratio between the two SNR is given by  $r = \frac{SNR_{real}}{SNR_{rec}} > \sqrt{2}$ , indicating a worsening of the SNR in the derivation process.

The actual value of the current maximum depends on the exact shape of the charge signal, so it is not known in general unless some analytical form of the leading edge is assumed. As an example, assuming that the rising edge of the charge signal is linear and contains  $N$  samples, then the reconstructed current signal will be a finite step of amplitude  $R/N$ , and the SNR will be worse by a factor  $r = N\sqrt{2}$ .

This model can be extended to evaluate the effects of the moving average, a LTI filter commonly used to reduce noise, on the SNR of the reconstructed current signal. Assuming that the charge signal has a  $N$ -samples long linear rising edge, we can start by applying the moving average (over  $M$  samples) and the first backward difference to the charge signal  $c[n]$  (assuming it has amplitude  $R$ ). Since the moving average can be viewed as a delayed difference (with delay  $M$ ) followed by an accumulator and a multiplication by  $\frac{1}{M}$ , then the result of the two filters can be evaluated by using the equation

$$y[n] = \frac{1}{M}(c[n] - c[n - M]) \quad (4.74)$$

which yields two different results depending on  $N$  and  $M$ :

$$y[n] = \begin{cases} \frac{R}{M} \frac{n}{N} & 0 \leq n \leq N \\ \frac{R}{M} & N < n \leq M \\ \frac{R}{M} - \frac{R}{M} \frac{n-M}{N} & M < n \leq M + N \\ 0 & otherwise \end{cases} \quad N \leq M \quad (4.75)$$

$$y[n] = \begin{cases} \frac{R}{M} \frac{n}{N} & 0 \leq n \leq M \\ \frac{R}{N} & M < n \leq N \\ \frac{R}{N} - \frac{R}{M} \frac{n-N}{N} & N < n \leq M + N \\ 0 & otherwise \end{cases} \quad N > M \quad (4.76)$$

In the first case the maximum of the reconstructed current signal is  $\frac{R}{M}$ , while in the second case it is  $\frac{R}{N}$ .

Assuming, as before, white noise with spectral density  $\omega_n$ , consider again the cascading of the moving average and the first backward difference. According to eq. 4.74, the transfer function of the whole filter is:

$$A(\omega) = \frac{1}{M}(1 - e^{-jM\omega}) \quad (4.77)$$

By substituting this expression in eq. 4.72 we obtain:

$$\sigma_f^2 = \frac{\nu_s \omega_n}{\pi M^2} \int_0^\pi (1 - \cos M\omega) d\omega = \frac{1}{M^2} \nu_s \omega_n = \frac{2}{M^2} \sigma_n^2 \quad (4.78)$$

By evaluating the SNR of the reconstructed current signal we then obtain:

$$SNR_{rec,MA} = \frac{MR}{\sqrt{2}\sigma_n \max(N, M)} \quad (4.79)$$

Finally, the ratio between the SNR of the sampled current signal and the SNR of the reconstructed current signal SNR becomes:

$$r = \frac{SNR_{real}}{SNR_{rec,MA}} = \begin{cases} \sqrt{2} & M \geq N \\ \sqrt{2} \frac{N}{M} & M < N \end{cases} < \sqrt{2}N \quad (4.80)$$

The SNR of the reconstructed signal is in both cases worse than the one of the real signal, but it is better than the result obtained without moving average. So it is useful to apply a moving average as a way to reduce the effect of noise on the reconstructed signal. Furthermore, since the moving average is a low-pass filter, it will slow down the signal transients, thus reducing the effects described in sections 4.7.1 and 4.7.2. The positive effects of the introduction of the moving average are evident when looking at fig. 4.10. The figure is obtained in the same way of fig. 4.7, and the only difference is the introduction of the moving average. Panel c) shows a comparison between:

- the output of the interpolating algorithm (black line);
- the output of a first backward difference (with PZC) applied to the averaged signal (black dots);
- the real current signal filtered with a continuous-time equivalent of the moving average (red line).

The interpolated current signal is closer now to the original signal than in fig. 4.7, showing also less oscillations.

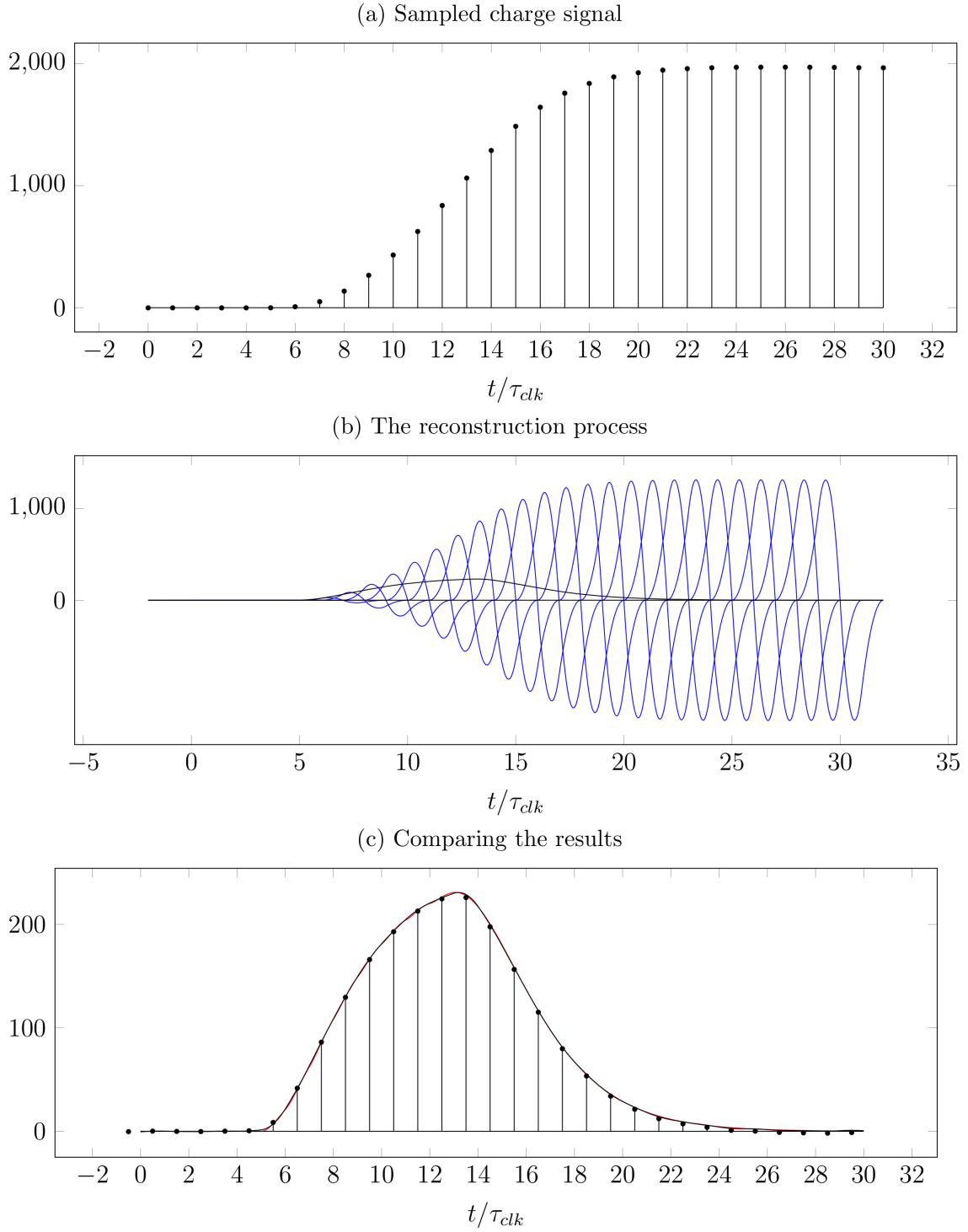


Figure 4.10: The derivation process of a charge signal using the moving average, as performed using B-Splines derivative with PZC, compared to a first backward difference (with PZC).

# Chapter 5

## Experimental Setup

In order to evaluate the performance of the digitizing board (equipped with the firmware designed in this work) a test under beam has been performed. In fact, the detection of nuclear fragments produced in nuclear reaction is the only way to test the PSA performance of a detector coupled with its Front End Electronics (FEE). We have taken advantage of a beam time allotted to the FAZIA collaboration by the Program Advisory Committee of LNS to perform the FAZIAsym experiment, in December 2015. On a side of the main experimental apparatus, we have mounted a three stage telescope (Si-Si-CsI(Tl)). The telescope is one of the prototypes built during the R&D phase of FAZIA and it is known to give good PSA performance, as published in a few papers in the last few years [9, 12, 13, 14]. The three stages of the telescopes are read out by three PACI preamplifiers, whose signals are then digitized and processed by the new GARFIELD digitizer boards. In this chapter, the experimental setup will be presented in some detail. The data were collected using beams of  $^{40}\text{Ca}$  and  $^{48}\text{Ca}$  at 35A MeV with various targets (mainly  $^{40}\text{Ca}$  and  $^{48}\text{Ca}$ , but also  $^{12}\text{C}$ ,  $^{112}\text{Sn}$  and  $^{197}\text{Au}$ )<sup>1</sup>.

During the experiment, the firmware has been changed to correct some mistakes and to add the moving average filter prior to interpolation (see sec. 4.7.3) because the first online results were not satisfactory.

### 5.1 The Si-Si-CsI Telescope

The main elements of the experimental apparatus are the FAZIA prototype telescope, the PACI preamplifiers and the digitizing boards. In fig. 5.1, a picture of the prototype telescope, mounted in the CICLOPE vacuum chamber of LNS and ready for measurement, is shown. The telescope axis pointed at the target, forming a polar angle of  $14.41^\circ$  with the beam axis. The telescope was placed in the horizontal plane containing the beam axis.

The telescope is equipped with two silicon detectors and a scintillator, a CsI(Tl), mounted as shown in fig. 5.2.

---

<sup>1</sup>For our purpose, the type of the reaction is of no importance, our only need was to produce fragments and to identify them.

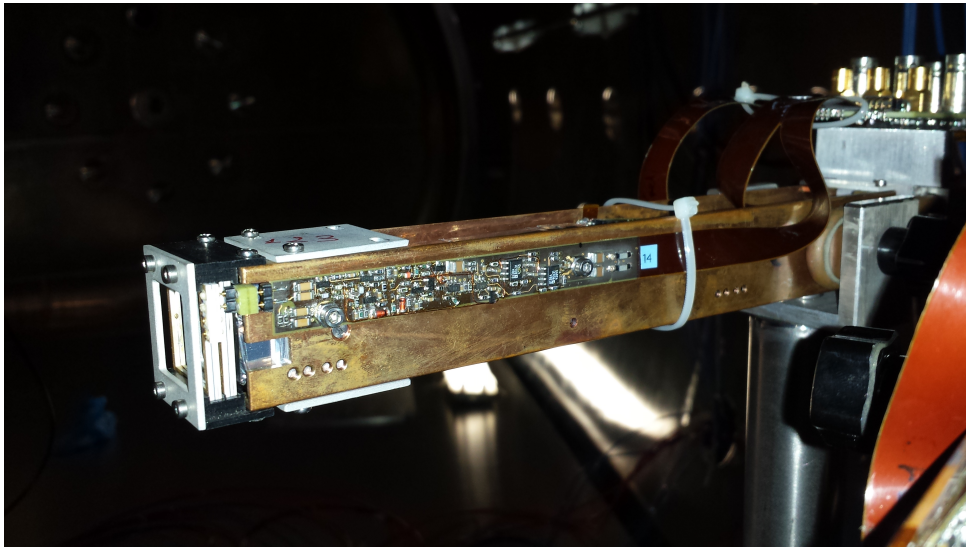


Figure 5.1: The telescope used for the test. The two silicon detectors are visible on the leftmost side of the picture (white frames) while the CsI is hidden by the copper frame. In the center, one of the PACI preamplifier (#14, connected to Si1) and, on the right, the telescope support and the board used for HV inputs, pulser inputs and signal outputs.

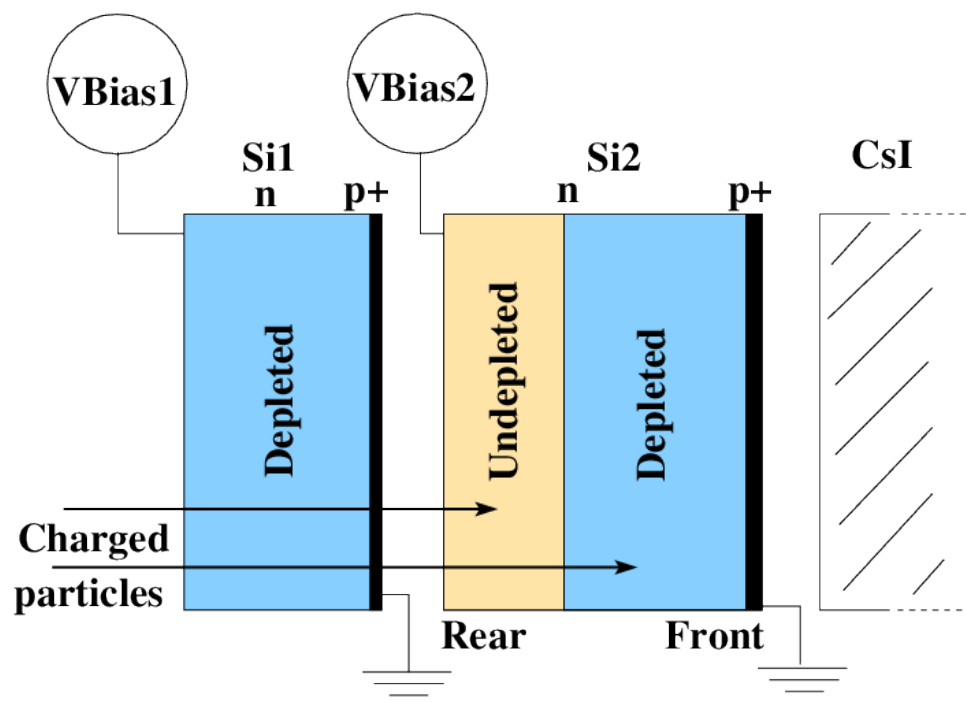


Figure 5.2: A schematic view of the FAZIA telescope, with its three detection stages (Si1, Si2 and CsI) from [14].



The characteristics of the three telescope stages are summarized in Table 5.1.

Table 5.1: The properties of the telescope stages

| Technical specifications    | Si1                   | Si2                   | CsI                   |
|-----------------------------|-----------------------|-----------------------|-----------------------|
| Thickness                   | 310 $\mu m$           | 510 $\mu m$           | 10 cm+PhotoDiode      |
| Bulk type                   | n                     | n                     |                       |
| Depletion Voltage           | 150 V                 | 290 V                 |                       |
| Applied Voltage             | 150 V                 | 220 V-270 V           | 30 V                  |
| Inverse Current @ Dep. Vol. | 4 nA                  | 17 nA                 |                       |
| Active Area                 | 20 $\times$ 20 $mm^2$ | 20 $\times$ 20 $mm^2$ | 20 $\times$ 20 $mm^2$ |
| Mounted                     | Rear                  | Rear                  |                       |
| Digitizer:bit/rate          | 14/120MHz             | 14/120MHz             | 14/120MHz             |

In addition, a 18 mm  $\times$  18 mm square collimator has been mounted in front of the telescope, to screen the silicon detector borders, whose response can be different from the rest of the detector active area. The rear mounting of the silicon enhances the performance of PSA (sec. 1.2.1) because the particle enters the detector from the low electric field side, thus exalting the effects of the plasma time. For the whole duration of the measurement, Si1 has been biased at his nominal depletion voltage (150V), while for Si2 different voltages have been tested (see chapter 6), to exploit also the enhancement in PSA obtained by underdepleting the silicon detector (see fig. 1.4) [14]. In fig. 5.2 the position of the undepleted region of Si2 is shown: the particle goes through Si1, which measures the  $\Delta E$  energy, then is stopped into Si2, either in the undepleted or in the depleted detector region, depending on its energy. Particles of longer range in Si ( $>820 \mu m$ ) are stopped in the CsI scintillator. The role of the CsI is to increase the dynamic range in energy for particle identification. Particles stopped in CsI can be identified, e.g., using the Si2-CsI  $\Delta E$ - $E$  correlation. For our purpose, this detector is used only as a veto to exclude from analysis the punch-through events, i.e. the events where the particle passes both silicon detectors and is then stopped in the CsI. In the following, only nuclear fragments stopped in Si1 or in Si2 will be considered.

### 5.1.1 The PACI preamplifiers

Each one of the three detectors is read out by a PACI preamplifier [11], which also delivers the bias voltage to it. PACI stands for “Pre-Amplifier for Charge and I-current”. It was developed at IPNO-Orsay a few years ago and it was adopted by the FAZIA collaboration. The PACI (visible in fig. 5.1) has four inputs (preamplifier voltages, detector bias voltage, pulser) and two differential outputs, one for the charge signal and the other for the current signal, as shown in the schematic of fig. 5.3.

The feedback capacitors of the PACI preamplifiers have different values in order to optimize the dynamic range in energy for each stage: in particular Si1 has a PACI with a range of 3.5 GeV, Si2 with 2.0 GeV.

The signals produced by the PACI are delivered to the digitizing boards and digitized without further processing, apart from the gain applied by the input stage of the board

## CHAPTER 5. EXPERIMENTAL SETUP

(described in sec. 5.2) and the low-pass antialiasing filtering.

The PACI has two differential outputs (charge and current) designed to couple with the FAZIA dedicated FEE, which has differential inputs. To double the dynamic range for

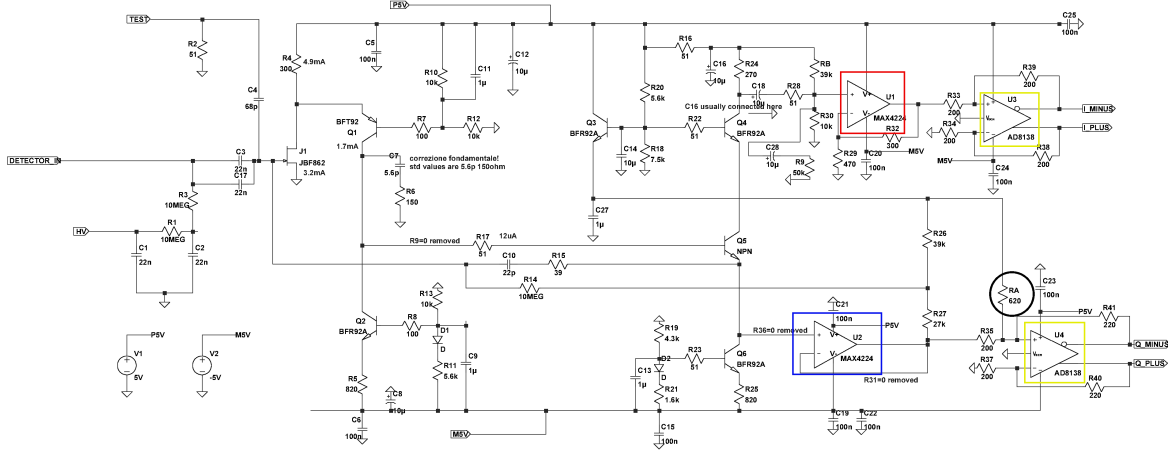


Figure 5.3: The electronic scheme of the PACI preamplifier. The operational amplifier in the red box amplifies the detector current signal, while the one in the blue box is the output of the charge preamplifier. The two buffers in the yellow boxes produce the differential outputs of the PACI.

unipolar signals, the differential output of the PACI in absence of signals from the detector is about  $-2V$ . A detector signal then rises the output voltage up to a maximum differential output of  $+2V$  (i.e. the total dynamic range is  $4V$ ). At rest, the “+” output of the differential output stage has a value of about  $-1V$ , while the voltage at the “-” output is about  $1V$ . However, the GARFIELD experiment, and thus our digitizing board, uses single-ended inputs. To feed the PACI signals to the digitizers, we have used cables with a differential connector on one side (PACI side) and two LEMO single ended connectors on the other (board side), using only one of the two components of the differential signal. This means that each of the single differential components has a non-zero DC level. In principle, changing the resistor  $R_A$  (black circle in fig. 5.3) allows for ajustement of the DC offset. Unfortunately, after noticing the problem there was not enough time to modify and test the PACI before the start of the experiment. Problems arise when applying gain to the signal. The aim of the variable gain stage is to fit the range of the signals into the range of the ADC, but it is also mandatory to avoid saturation of the analog input stage in order to maintain the linearity of the system. The presence of a non-zero DC level could then be a limitation, i.e. the DC level is amplified as part of the signal, eventually saturating the variable gain stage. In fact, the effect depends on the amplitude of the signals. Since the variable gain stage has a linear response for output voltages in a symmetric interval  $(-V_L, V_L)$ , we can compare the maximum gain  $G$  that can be applied

in the two cases (with or without DC offset). We obtain the following result ( $A$  is the signal amplitude):

$$G(V_{offs} = 0) = \frac{V_L}{A} \quad (5.1)$$

$$G(V_{offs}) = \begin{cases} \frac{V_L}{V_{offs}} & A < 2V_{offs} \\ \frac{V_L}{A - V_{offs}} & A \geq 2V_{offs} \end{cases} \quad (5.2)$$

When  $V_{offs} \neq 0$  and  $A < V_{offs}$ , one has to use a smaller gain, in order to avoid saturation, than if the offset is zero. This was the case as regards the present experiment.

## 5.2 Digitizing boards and firmware

The main element of the experimental setup were of course the digitizing boards we intended to test. Their design started a few years ago. They have been designed by Stefano Meneghini of the Electronic Group of INFN Sezione di Bologna, following the specifications and hints of the Florence group, as a new version of the digitizer board used at LNL for the GARFIELD apparatus [2]. The board is fully compatible with the VME motherboards used with the previous version. In this work, we are particularly interested into the PSA performance of the firmware, based on the concept illustrated in chap. 4. Each board hosts two independent channels<sup>2</sup>. Each channel is equipped with a 14-bit 125MHz ADC (Analog Devices AD9255 [26])<sup>3</sup>. The input of the ADC can be selected via an electronic switch (controlled by the FPGA), allowing for the following choices:

1. the channel input signal right after a variable gain stage;
2. the channel input signal attenuated by a factor of five and then passed through a variable gain stage;
3. an external sinusoidal signal (RF) used for timing measurement and clock synchronization, according to ref. [27];
4. the other channel's input signal taken after the variable gain stage (aux input).

A simplified schematic of the input stage is shown in fig. 5.4.

The first two choices allow for a full exploitation of the ADC range, according to the particular detector to be sampled. Though the ADC range is fixed at  $\pm 1$  V, the full scale, in terms of voltage at the board input connector, ranges from  $\sim 100$  mV to 10 V, depending on the switch and variable gain settings. The input impedance is  $50 \Omega$  for easy matching with the most commonly used coaxial cables.

The fourth option (which hasn't been tested yet) should be used to double the sampling

<sup>2</sup>The previous version has half the width of the new one and hosts just one digitizing channel.

<sup>3</sup>Until now, the ADC are used at 120MHz sampling rate. Replacement of the 24 MHz quartz with a 25 MHz one, in order to get a sampling rate of 125 MHz, i.e. 8 ns sampling period (as on the previous version of the board) is foreseen in the next future.

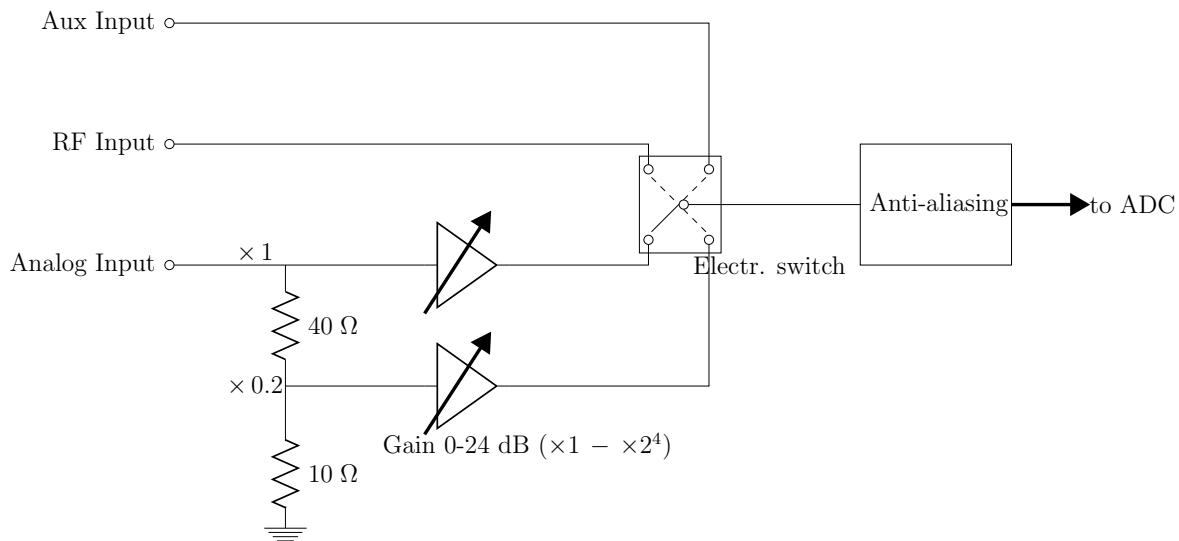


Figure 5.4: Simplified schematic of the input stage with the two variable gain stages, the electronic switch and the anti-aliasing filter.

frequency (dividing by two the number of available channels); in fact, the sampling clocks of the two channels have a mutual phase shift of half a period so that sampling the same signal on both channels and interlacing the two resulting sequences gives the same result as if the signal was sampled at double the frequency (e.g. 240 MHz). The two 120 MHz clocks are generated from a single 24 MHz clock using a dedicated PLL.

In addition to the switch and the variable gains, a variable offset, generated by a digital to analog converter (DAC, not shown in fig. 5.4), is added to the signal right before the ADC input. In fact, detector signals are usually unipolar, with positive or negative polarity depending on the particular detector and preamplifier. Thanks to the variable offset, the best conditions compatible with eq. 5.2 can always be granted. Finally, the signal passes through an anti-aliasing stage. The board implements a Bessel anti-aliasing stage, with three poles (18 dB/octave slope) and a  $-3$  dB frequency of 40 MHz.

Apart from the ADC, which is obviously the core component of any digitizing system, the two most important components of the board are the DSP and the FPGA.

### 5.2.1 The DSP

A DSP (**D**igital **S**ignal **P**rocessor) is a microprocessor optimized for digital signal processing. In the previous version of the digitizing board, all the online signal processing is performed by the DSP (Analog Devices AD2189N, [28]). The new board, instead, uses the DSP (same model as for the old board) mostly as an interface with the motherboard and the computer. Obviously, the DSP can still be used for signal processing, but now some calculations can be performed by the FPGA. Actually, in the measure at LNS we used the DSP only to access the FPGA memory and to compress and send the data to the computer.

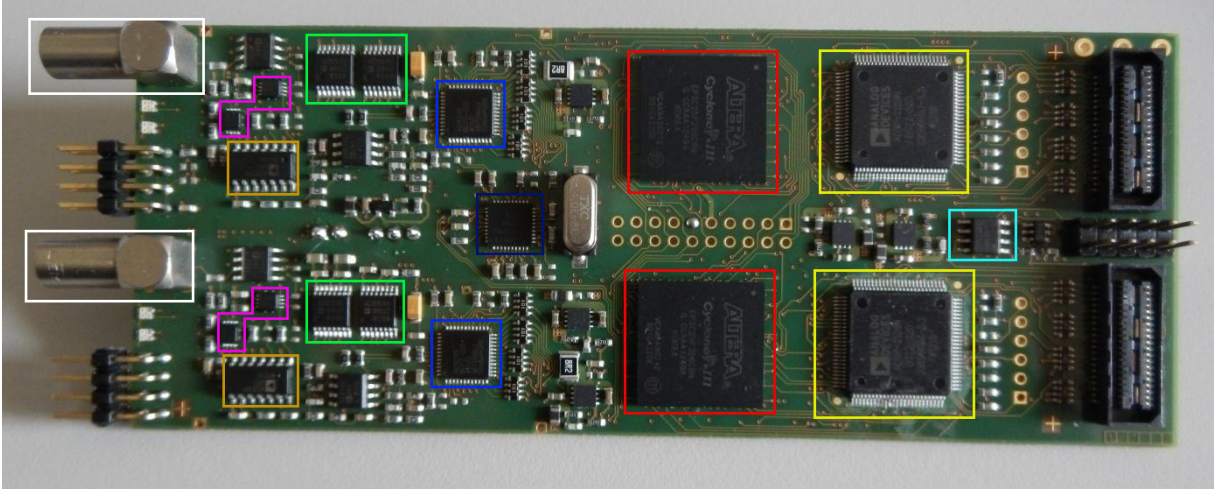


Figure 5.5: A picture of the new digitizing board with the key components highlighted: Altera CycloneIII FPGA (Red), Analog Devices AD2189N DSP (Yellow), Analog Devices AD9235 ADC (Light blue), PLL (Dark blue), DACs for gain/offset control (Green), variable gains (Purple), ADC input switches (Orange), EEPROM (Azure) and the input lemo connector (White).

### 5.2.2 The FPGA

The addition of a Altera CycloneIII EP3C25F256C8N FPGA [29] represents the main improvement introduced with respect to the previous design. The FPGA has several roles: it has been introduced as an upgrade with respect to the simple FIFO memory present on the old board, in view of developments which are required to speed up data processing.

A general overview of the firmware (developed during the last year) used for the measure is given in sec. 5.3. The CycloneIII EP3C25F256C8N FPGA is a powerful device, in fact it features:

- 66 M9K blocks (RAMs) for a total memory of 608.256 bits;
- 132 9x9 multiplier blocks (ref. [30]);
- 4 embedded PLLs (ref. [31]);
- 215 maximum user I/O pins;

For more details we refer to the device datasheet [29]. In our firmware, we have particularly exploited the RAM blocks, which are used for storing the signals and for the delay lines needed by the trapezoidal and bipolar shaper. The multiplier blocks are also of great utility when implementing linear filters: in the final version of the firmware, almost all the available multiplier blocks are used.

### 5.3 General overview of the firmware

The firmware used for the measure at LNS is the result of a few months of work. Actually, the design did not start from scratch. A basic firmware including the FIFO and a first version of the memory registers was already available. During our work the firmware has grown, with the addition of new blocks, up to the present state, represented in fig. 5.6. In this picture, the main blocks of the firmware are represented as white rectangles. Each arrow represents a path on which data can flow. The colors of the different paths have the following meaning:

- black solid: path for signals carrying physically relevant information (ADC output, shaper output, interpolated signal);
- red: path for internal settings signal (e.g. the TAU parameter for the trapezoidal shaper);
- blue: path for trigger signals;
- green: path for the final data (energy, current maximum, ADC signal from FIFO);
- black dashed: other signals (including clocks).

Finally, light blue rectangles represent other physical components that are placed on the board (e.g the EPCS) and light red ones represent physical devices which are not on the board.

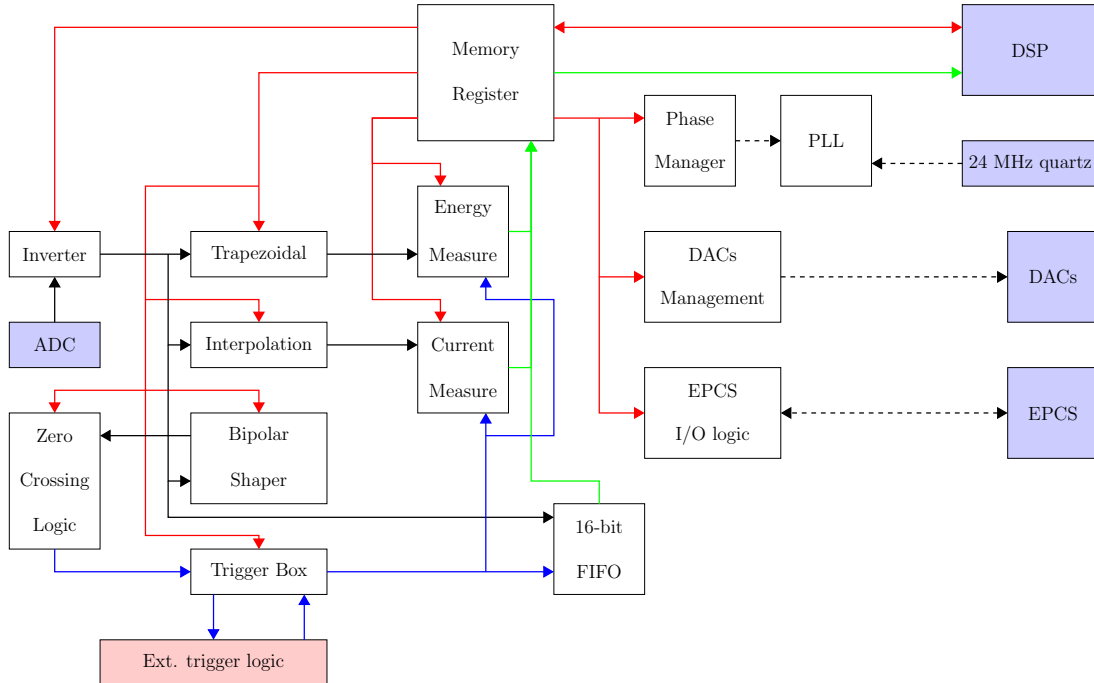


Figure 5.6: Block diagram of the FPGA firmware.



The core of the whole firmware is the memory register. This block is responsible for interfacing the FPGA with the DSP and, through it, with the computer. The block is used to store all the settings of the channel and to send the final data to the DSP. All the parameters stored in the memory register can be read/written by the DSP, viceversa the final data are read-only.

In order for the board to be able to operate with signals of any polarity an inverter block is used to complement each bit of the signal (one's complement) if necessary. In fact, the implemented filtering algorithms work correctly whatever the signal polarity. The problem arises when measuring the maximum current (which becomes the minimum current for negative signals) and when looking for the threshold crossing for zero crossing detection (sec. 3.2). To avoid this, the one's complement is applied to negative polarity signals, giving positive polarity signals. The inversion is controlled by a parameter (POLARITY) which can have values 0 (no inversion) or 1 (inversion).

The output of the inverter is then sent to the three filters (trapezoidal, bipolar, interpolating) and to the 16-bit FIFO, which is capable of holding up to 2048 samples of the signal. The trapezoidal filter and the bipolar filter have already been described in chapters 2 and 3, the interpolating filter is described in detail in sec. 5.4. Each filter generates a different output signal. The output of the trapezoidal filter is used to measure the energy of the impinging particle (sec. 2.2.2), the interpolating filter produces a signal, related to the current signal (see sec. 5.4.2), which is used to measure the maximum current. As described in chapter 3, the output of the bipolar shaper is monitored by the zero-crossing detection logic, which generates an internal trigger signal. Depending on the settings of the trigger-box (sec. 3.3), the internal trigger can be used to generate a trigger event or sent outside the board to be processed by an external trigger logic system.

On the right side of the diagram, three other blocks are represented. The DAC management block was already present in the very basic firmware from which we started, and it acts as an interface to send the desired settings to the DAC (DAC output voltages control gain and offset). The EPCS input/output is a simple state machine, which was implemented to allow reading/writing of the EEPROM. With this feature it is possible to load a new FPGA firmware on the EEPROM without using the JTAG cable, so that it is not needed to remove the board and its motherboard from the VME crate for programming. Finally, the PLL is used to generate, starting from the external 24 MHz clock, the fast 120 MHz clock used by the FPGA and a slower 40 MHz clock used by the FPGA and the DSP. During the test phase preliminary to the measure, we discovered some problems in the signal acquisition process. The acquired signals suffered from digital "glitches" due to wrong bits caused by timing violations between the ADC output and the FPGA sampling clock. To solve this problem we have modified the PLL introducing the possibility to change the phase of the 120 MHz clock. Since the ADC 20 MHz clock is produced by an external PLL, by changing the mutual phase-shift of the two clocks it is possible to fulfill the timing requirements and avoid timing violations.

## 5.4 Implementation of the interpolating filter

The current signal is reconstructed in real-time using the algorithms described in chapter 4. Implementing the concepts of chapter 4 on the FPGA (we used VHDL language[32]) was not a trivial task. Since we are upsampling the current signal by a factor of ten, we need to deal with not time-limited signals having a sampling frequency of 1.2 GHz, while our device is clocked at 120 MHz. Actually, by exploiting a pipelined algorithm, these signals are treated in parallel, at a 120 MHz frequency. This section gives some insights about the implementation of the filter and related algorithms. The CycloneIII FPGA features 132 embedded 9x9 bit multipliers, which turn out to be very useful when implementing linear filters by means of the constant coefficients difference equations. Since, as described in sec. 4.6, we have to implement a set of linear filters, we must ensure that the design uses less than 132 multipliers<sup>4</sup>. It is also important to notice that the coefficients of the difference equations will be represented in fixed-point arithmetic.

### 5.4.1 Inverse B-Spline filter

In designing the inverse B-Spline filter (described by eq. 4.32) we have decided to represent the coefficients in a signed 16 bit integer notation. As already said in sec. 4.4.3, the coefficients are multiplied by a factor  $2^{14}$  and then expressed in signed integer notation. Therefore, one is represented as  $2^{14}$  (binary codev 0100000000000000). The quantized version of the filter impulse response is:

$$h_{BSinv,Q}[n] = \text{round}(2^{14} * \hat{h}_{BSinv}[n]) \quad (5.3)$$

Where the *round* operator is an approximation to the nearest integer value. Unfortunately, since the rounding operation is not linear, the discrete-time integral of the obtained function is not equal to  $2^{14}$ , as desired. In fact, in sec. 4.4.2 we asked the discrete-time integral of the truncated impulse response to be equal to one. For the same reason, we now request that the discrete-time integral of the truncated and **quantized** impulse response is  $2^{14}$ . By operating in analogy with eq. 4.32 we define (remembering that  $\tilde{n} = 8$ ):

$$\hat{h}_{BSinv,Q}[n] = \begin{cases} \text{round}(2^{14} * h_{BSinv}[n]) & |n| < 7 \\ C & |n| = 7 \\ 0 & otherwise \end{cases} \quad (5.4)$$

with  $C = \sum_{n=-6}^6 \frac{\text{round}(2^{14} * h_{BSinv}[n])}{2}$ . By exploiting the evenness of  $h_{BSinv}[n]$  we can see that:

$$C = \frac{\text{round}(2^{14} h_{BSinv}[0])}{2} + \sum_{n=1}^6 \text{round}(2^{14} * h_{BSinv}[n]) \quad (5.5)$$

and, in order for  $C$  to be an integer, we need  $\text{round}(2^{14} h_{BSinv}[0])$  to be even. Fortunately, since  $h_{BSinv}[0] = \sqrt{3}$ , we obtain  $\text{round}(2^{14} h_{BSinv}[0]) = 23778$ .

---

<sup>4</sup>The pole-zero correction algorithm of the trapezoidal shaper uses 8 multipliers.



The same process expressed by the response of eq. 5.4 can be obtained using a difference equation. In fact, calling  $x[n]$  the input of the system, its output is given by

$$y[n] = \sum_{m=-7}^7 \hat{h}_{BSinv,Q}[m]x[n-m] \quad (5.6)$$

In this form, the filter cannot be implemented in real-time, since it is not causal. To make it causal we just need to add a delay of seven samples to  $x[n]$  before applying the filter, generating an intermediate signal  $x_d[n] = x[n-7]$ . We obtain:

$$y[n] = \sum_{m=-7}^7 \hat{h}_{BSinv,Q}[m]x'[n-m-7] \quad (5.7)$$

which represents a causal filter. By exploiting the evenness of  $\hat{h}_{BSinv,Q}[m]$  we can rewrite eq. 5.7 in a simpler form:

$$y[n] = \hat{h}_{BSinv,Q}[0]x_d[n-7] + \sum_{m=1}^7 \hat{h}_{BSinv,Q}[m](x_d[n-m-7] + x_d[n+m-7]) \quad (5.8)$$

This is not just a formal simplification. In fact, implementing eq. 5.8 requires eight multiplications, while eq. 5.7 requires fifteen multiplications.

The block scheme of the implementation of this filter is shown in fig. 5.7. During its development we forgot to count the sign-bit of the coefficients and performed all the calculations assuming to represent the coefficients in a 1.15 signed notation (see note in sec. 4.4.3). With this assumption one obtains  $\tilde{n} = 10$  (instead of 8). Therefore, the actually implemented filter uses nineteen coefficients (instead of seventeen), and it needs ten multiplications (instead of eight) and a delay of nine samples (instead of seven). In the future we plan to correct this error in order to reduce the FPGA resources used by the filter. The block in a green square in fig. 5.7 is simply a memory block which contains the coefficients of the difference equation. The delay block (enclosed in a red solid rectangle) implements the delay line as well as a part of the summation. It has ten outputs ( $S_m$ ,  $m = 0..9$ ) obtained by the delayed signal:

$$\begin{aligned} S_0 &= x_d[n-9] \\ S_{(m \neq 0)} &= x_d[n-m-9] + x_d[n+m-9] \end{aligned}$$

With this notation, eq. 5.8 becomes:

$$y[n] = \sum_{m=0}^9 \hat{h}_{BSinv,Q}[m]S_m \quad (5.9)$$

The five identical blocks enclosed in the black rectangle perform the product of each  $S_m$  and its coefficient. Each block takes four inputs (two signals and two coefficients) and it outputs, after a constant delay, the sum of the two products. The sum of the five numbers is then performed by a pipelined adder (evidenced by the blue box). The pipeline is needed because it is not possible to sum more than two values in the same clock cycle. With five numbers to sum ( $n_1, n_2, \dots, n_5$ ) the system works as follows

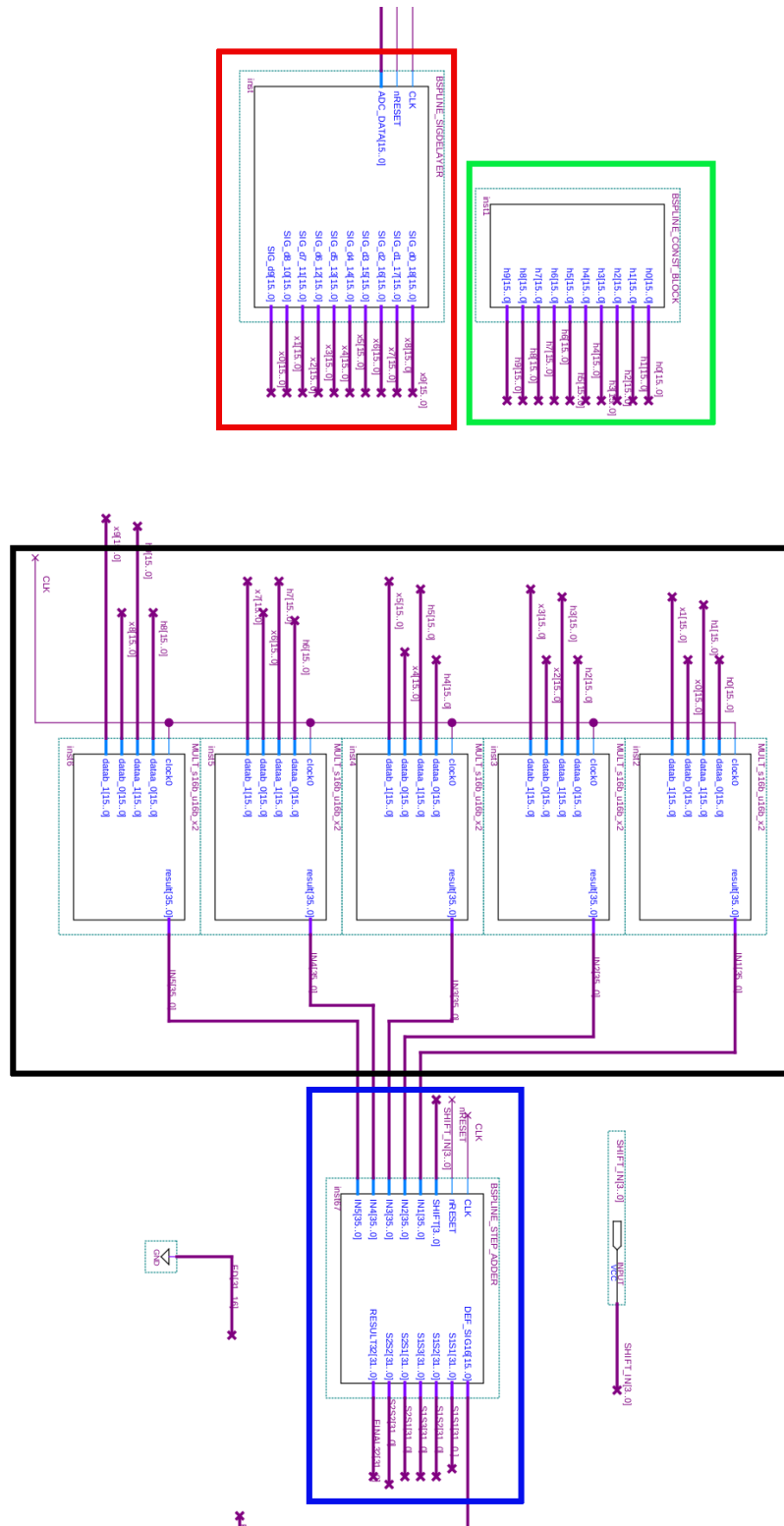


Figure 5.7: The block scheme of the inverse B-Spline filter.

1. on the first clock cycle,  $n_1 + n_2 = n_{12}$  and  $n_3 + n_4 = n_{34}$  are evaluated,  $n_5$  is kept;
2. then  $n_{12}$  and  $n_{34}$  are added to obtain  $n_{1234}$ ,  $n_5$  is kept;
3. finally, the two remaining values  $n_{1234}$  and  $n_5$  are added to obtain the final result.

The output of the adder block is a 36 bit word and it must be truncated to 16 bits. In fact, the filter used to reconstruct the current signal exploits 16-bits signals. Truncation is performed inside the adder block, and it is controlled by a parameter that selects the interval of bits to be kept, in order to avoid:

- saturation of the filtering stages present after the truncation;
- suppression of the signal with subsequent loss of information.

### 5.4.2 Reconstruction filter

The reconstructed current signal is obtained from the B-Splines coefficients  $b[n]$  produced by the inverse B-Spline filter, in the way described in chapter 4, and then sampled with a frequency (1.2 GHz) ten times higher than the FPGA clock frequency. The implementation of this filter follows directly from equation 4.57 and from the results of chapter 4. The overall filter is obtained by using ten distinct LTI filters, whose impulse response is given by:

$$h_m[n] = B_{3PZC;DER}(n + m/10) \quad 0 \leq m < 10 \quad (5.10)$$

A filter with impulse response given by eq. 5.10 can be implemented with the following difference equation<sup>5</sup>:

$$y_m[n] = i_{rec}[10 * n + m] = \sum_{k=-2}^{k=1} h_m[k]b[n - k] \quad (5.11)$$

These filters are not causal, but they can be made causal by introducing a delay of two samples. In fact:

$$y_{dm}[n] = \sum_{k=-2}^{k=1} h_m[k]b[n - k - 2] \quad (5.12)$$

where we have defined  $y_{dm}[n] = y_m[n - 2]$ . The filter block scheme is depicted in fig. 5.8.

---

<sup>5</sup>By inspecting eq. 5.10 we can observe that the set of coefficients needed to describe all of the ten linear filters is nothing but a sampled version of the kernel function.

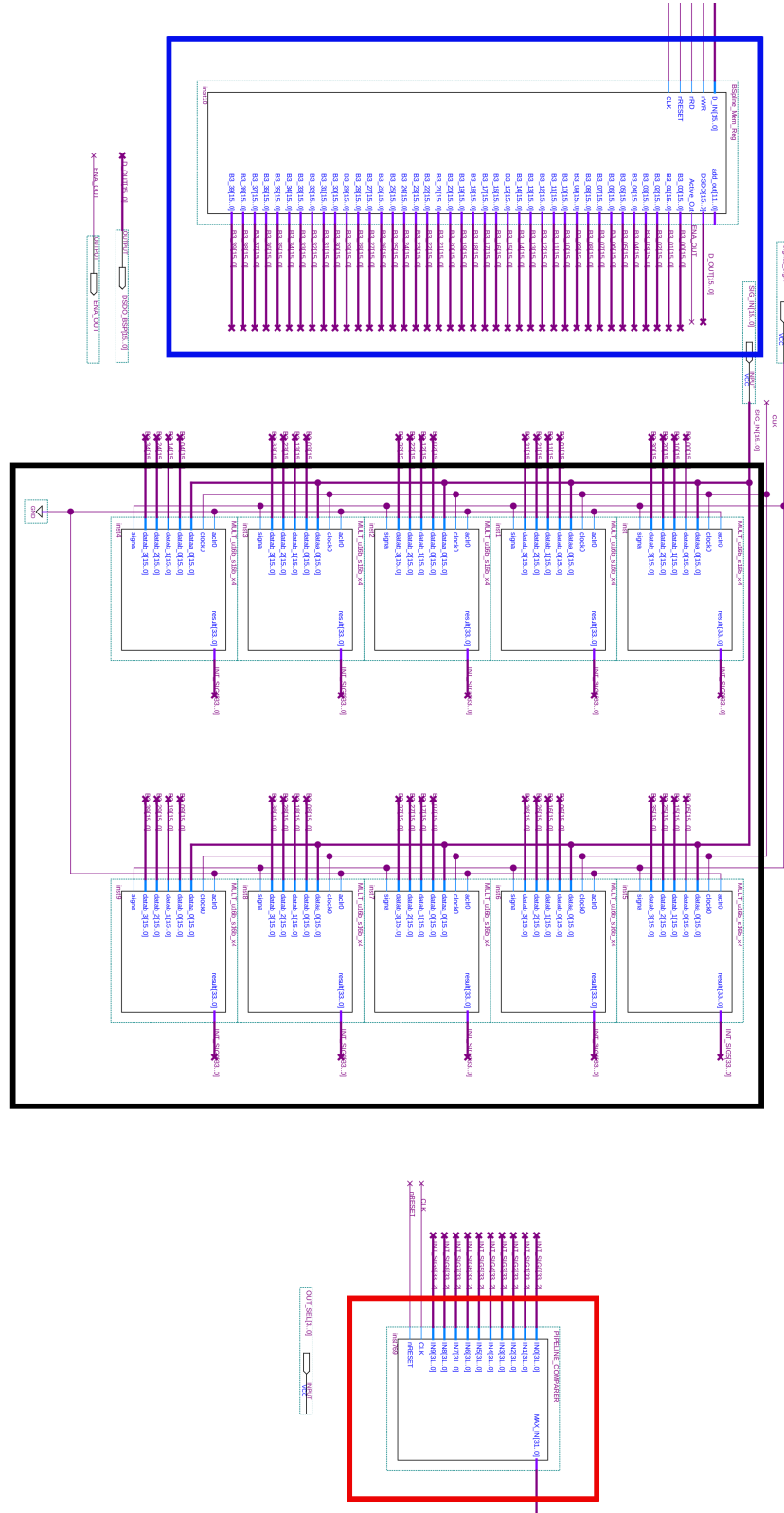


Figure 5.8: The block diagram of the reconstruction filter.

The ten LTI filters are implemented using ten multipliers (evidenced by the black rectangle), each one featuring:

- a 16-bit signal input (the signal delay line is embedded in the multiplier);
- four 16-bit inputs for the coefficients.

The forty coefficients are stored in the memory block visible on the left (enclosed in the blue rectangle). Since the coefficients are nothing but samples of the kernel function, and since the kernel function used here ( $B3_{PZC;DER}$ ) depends on external parameters such as the preamplifier decay time, these coefficients aren't fixed. They can be adjusted depending, e.g., on the particular preamplifier used in the experiment. Therefore, the memory block can be accessed for read/write operations in order to change the coefficients. The particular kernel used for reconstruction depends on the operation to be performed on the signal. Sometimes simple interpolation is required, e.g. when a current signal has been sampled and we just need an upsampled version for a better maximum determination. In this case, to evaluate the maximum current, one uses simple cubic B-Spline interpolation, writing the coefficients obtained from the  $B3(x)$  kernel (instead of  $B3_{PZC;DER}(x)$ ) in the memory block of the reconstruction filter. If, as already discussed, one uses the coefficients derived from the  $B3_{PZC;DER}(x)$  kernel, an upsampled version of the derivative of the input signal is obtained.

The reconstructing filter continuously generates ten samples for every FPGA clock cycle. Managing such an upsampled signal can be really cumbersome. Fortunately, we are just interested into measuring the maximum amplitude of the signal, so that we can again exploit a pipelined architecture. The block evidenced in red on the right side of fig. 5.8 performs a pipelined comparison of the ten samples produced. Each clock cycle, samples are compared in pairs, and the larger value of each pair is kept in memory: one goes from ten samples to five, from five to three, then to two, until only the biggest sample remains. The output of the comparator is then a 32-bit signal  $I[n]$  which is sampled at the FPGA clock frequency given by:

$$I[n] = \max \left\{ i_{rec} \left[ n + \frac{m}{10} \right] \quad m = 0..9 \right\} \quad (5.13)$$

### 5.4.3 Overall maximum detection

The signal  $I[n]$  is used to detect the overall maximum of the reconstructed current signal. The search for the maximum is performed only when a trigger event occurs. It is performed over a fixed amount of samples following the trigger event itself. Actually, we don't use  $I[n]$  to perform the maximum current detection but a delayed copy of  $I[n]$ , with user defined delay. The delay line is needed when the trigger is generated using an external logic device (as described in sec. 5.5). In fact, using external logic introduces a delay between the trigger signal and the charge/current signal, so that it is possible that the current signal reaches its maximum before the trigger, thus spoiling the measurement. The delay applied to  $I[n]$  make so that the maximum of delayed signal arrives after the trigger, and therefore it is measured correctly. The current maximum detection logic is triggered using the Experiment trigger (see sec. 5.5).

### 5.4.4 Introduction of the moving average

In section 4.7 we described the main problems related to the detection of the current maximum:

- the error committed when taking the maximum amongst a finite set of sampled values.
- the accuracy of the signal reconstruction;
- the presence of noise over the charge signal, noise which, even after filtering, is still present on the final current signal.

The first contribution (see sec. 4.7.1) can be reduced either by slowing down the signal or by increasing the sampling frequency. Since the current signal is reconstructed with an upsampling of ten, the sampling frequency has been actually increased. The second contribution depends mainly on the presence of sudden steps in the signal. Unfortunately, the most critical part is the rising edge of the signal, which is exactly where the current signal information is located, so the solution of this problem is mandatory. Finally, the noise impact on the maximum detection can be reduced by reducing the SNR of the charge signal before interpolating it.

Suppose now to apply a moving average (performed over  $M$  samples) to the charge signal. By doing so we obtain that:

- the transients of the charge signal (therefore also the current signal) are slowed down, reducing the value of its second derivative (e.g. a sudden step is transformed into a ramp);
- the noise will be reduced so that, since the charge signal amplitude stays the same, the SNR is increased.

This last point is to be discussed in more detail. In fact, by slowing down the charge signal, its derivative, the current signal, is reduced in amplitude. Since the aim is to measure the maximum of the current signal, we must now evaluate the SNR for the current signal. The effects of the moving average have already been described in sec. 4.7.3 using a simplified model. To get some insights into the effect of applying a moving average, one should to take into account the actual shape of the charge/current signal and one should apply the actual interpolating algorithm instead of a simple first backward difference. Since such a calculation would be cumbersome, we decided to run a numerical simulation instead. The results of the simulation will be illustrated in sec. 6.3. The moving average has been implemented during the beam time, exploiting a pause of one day in data taking. With such a small time to implement and test it, we decided to use a simple implementation. In fact, the moving average can be performed over 4 or 8 samples only. Furthermore, it is not actually a moving average. When “averaging” over 4 samples, one actually obtains the sum of the four samples, while with 8 samples one obtains the sum of the 8 samples divided by two. Since we are dealing with a 14-bit signal, the output of the system described above is a 16-bit signal. The further division, needed to perform a true average, is performed inside the pipelined adder block (see sec. 5.4.1) when truncating the result of the summation.

## 5.5 Acquisition and trigger logic

The data from the three detectors have been acquired using three digitizing boards. A total of 5 independent digitizing channels were used:

- two channels for Si1 (charge+current);
- two channels for Si2 (charge+current);
- one channel for CsI (charge only).

For each channel, acquired data included the full signal (about  $2\mu s$ , including a portion of the baseline) and the data produced by the FPGA as a result of the various algorithms implemented.

The acquired signals are then analyzed offline to get the same information produced by the FPGA, to be compared to the real-time calculation. In particular, the sampled current signal are used to compare the maximum of the current as obtained from the current signal itself to the results of the reconstruction (obtained as described in chapter 4). Here is a summary of the various operations performed offline on the signals:

- evaluation of the energy for Si1 and Si2 using a trapezoidal shaper;
- evaluation of the energy for the CsI using a trapezoidal shaper (used for vetoing punch-through);
- evaluation of the maximum of the current signals from Si1 and Si2 with Cubic B-spline interpolation (for comparison with the reconstructed  $I_{max}$ );
- offline replica of the interpolating filter applied to the charge signal, to check for errors in the implementation of the real time filtering.

Some of the results produced by our analysis are exposed in chapter 6.

For the whole duration of the experiment, we have used as acquisition trigger (called Experiment trigger from now on) the logic **or** of the single detector triggers. These triggers are generated, as described in chapter 3, by the channels dedicated to the acquisition of the charge signals.

When a Experiment trigger occurs, all the digitizing channels are acquired, regardless of their status. This way, even if only one out of three telescope stages is above its triggering threshold, all the channels are acquired. The bipolar shapers were set to a rise-time ( $RT$ ) of ten samples and a rise-time plus flat-top ( $RTFT$ ) of twenty samples, while the threshold for the zero detection algorithm has been chosen individually for each channel depending on the noise level (in order to avoid triggering on noise), and it has been adjusted every time the gain of the channels was changed. All the channels were set to output their internal trigger (trigger signal generated by the zero-crossing detection system) and to respond to the CFDIN trigger (see section 3.3). The internal triggers of the boards which were sampling the current output of the PACI were not used for triggering. In order for all the channels to be acquired on every event, the Experiment trigger produced is then injected into the motherboard 8-input ECL connector (connected to the CFDIN inputs of the digitizing channels). The Experiment trigger is then responsible for:

- triggering the signal acquisition on the FIFO;
- triggering the current maximum detection (sec. 5.4.2);
- triggering the peaking time for energy measurement (sec- 5.6);
- triggering the DSP acquisition routine.

## 5.6 Energy measurement: trapezoidal shapers settings

Apart from the pole-zero correction constant, the trapezoidal shaper parameters were set to the same value for all three charge channels, namely a rise-time ( $RT$ ) of 128 samples, a rise-time plus flat-top ( $RTFT$ ) of 384 samples (so a flat-top of 256 samples) and a peaking time of 310 samples.

In principle, as described in section 2.2.2, the peaking time method consists in acquiring a single sample of the trapezoidal signal, taken a fixed amount of time after a triggering event has occurred. To ensure that the acquired sample has a fixed position with respect to the signal's shape, we use as triggering event the internal trigger of the channel. However, using the internal trigger has some inconvenience. As described in sec. 5.5, all channels are acquired each time one of the charge channels generates an internal trigger. So, if an event occurs with only one channel producing an internal trigger, the two other channels will not update their trapezoidal amplitude information, and the acquisition system will read the value acquired for the last internal trigger occurred. This typically happens with low-energy heavy fragments (stopped in Si1) and with light fragments (mainly  $Z = 1$ ) that release a very small amount of energy in the two silicon detectors. To avoid this effect, it is then necessary to trigger the amplitude acquisition using the Experiment trigger, so that the energy value is updated regardless of the absence of an internal trigger. Therefore, since the Experiment trigger has no fixed position with respect to the signal's shape, we expect the measure to be less precise (expecially if the preamplifier decay is not corrected properly). With this in mind, we have inserted two identical blocks in the firmware. Each block performs the energy measurement as described in sec. 2.2.2. One block is triggered by the internal trigger while the other uses the Experimental trigger. This duplication is exploited only when using the peaking time method. If, for example, one takes the absolute maximum of the shaped signal, then the result is independent on the trigger, so in that case it would be better to trigger only on the Experiment trigger. In fact, for what concerns the detection of the maximum current, this solution has been used (sec. 5.4.3).



## Chapter 6

# Data Analysis and related remarks

The beam test at LNS, Catania, has been performed during the FAZIAsym experiment, from the 9th to the 20th of December 2015. The experiment is divided in two parts:

- in the first part a 35 AMeV  $^{40}\text{Ca}$  beam is used, with different targets;
- in the second half, we use a 35 AMeV  $^{48}\text{Ca}$  beam, again with different targets.

Basically, the particular nuclides involved in the reaction are of no interest for our purpose, provided that nuclear fragments are produced and imping on the telescope with a large enough range of energy, charge and mass. As described in sec. 5.6, the first part of the experiment has been spoiled by some bugs and limitations of the firmware. It served mainly as a first test for the firmware itself, leading to some modifications. The corrections, performed during the beam change procedure, consisted in:

- addition of a duplicate of the peaking time logic, to have a “backup” measure of the Energy even when the internal trigger is missing (sec. 5.6);
- introduction of the moving average before the interpolating filter (sec. 5.4.4).

Unfortunately, as a side-effect of the increase of complexity, timing violation has appeared in the firmware. Timing violations were associated to the read-out path from the internal memory of the FPGA to the DSP. Therefore, they affected the signals acquired and stored for offline analysis. On the other hand, the online processing chain was not affected. As a consequence, acquired current signals suffer from digital glitches (isolated wrong samples) due to wrong bits. To reduce this problem the design should go through an optimization process, that we could perform only partially, due to the lack of time. In this chapter the results of the second half of the experiment are presented and discussed. In the following, we will refer to the energy measured by the FPGA with respect to the Experiment trigger as “online energy”, while neglecting the other measure, based on the internal trigger, since it is not always present.

## 6.1 Offline data analysis

The collected data is analyzed offline using a C++ code within the ROOT framework [33]. Apart from reading from disk the data produced by the FPGA during the experiment, the offline analysis program performs a set of different analysis on the sampled signals. For each event we have:

- two signals from Si1 (charge+current);
- two signals from Si2 (charge+current);
- one charge signal from the CsI.

For the two charge signals of the silicon detectors we perform:

- offline energy measurement exploiting a trapezoidal shaper;
- offline replica of the algorithm implemented on FPGA for current signal reconstruction;
- evaluation of the average value and variance of the base-line (useful, e.g., for pile-up event rejection);
- evaluation of the charge signal rise-time (20%-70%).

The sampled current signal has also been used to determine offline the maximum current. The evaluation of the maximum is performed on an upsampled version of the sampled current signal, obtained by means of cubic B-spline interpolation. Unfortunately, this last processing is affected by the glitches that are present over the sampled signal as anticipated in the introduction to this chapter. The analysis program also features a glitch-removing algorithm, implemented to reduce the problem. The algorithm has worked quite well, making the results better, though some effects of the glitches are still visible (see fig. 6.9). Finally, the CsI energy is extracted from the CsI charge signal, using a trapezoidal shaper. The output data produced by the analysis are then stored and used to generate the pictures in sections 6.2.3 and 6.2.4.

## 6.2 Results of the measure at LNS

The results obtained during the second half of the FAZIAsym experiment with our experimental setup are reported in this section. The presentation is split into three main parts. The first part is a direct comparison of the offline and online results for energy and maximum current, in order to demonstrate that the FPGA firmware is working as expected. The second part is the energy calibration of the Si detectors and the last part contains the correlation matrices used for fragment identification (online and offline). Furthermore, another section exposes the results of an offline simulation used to better understand the performances of the reconstructing filter described in sec. 5.4 .

### 6.2.1 Comparison of online and offline data

In order to compare, event by event, the energy and maximum current obtained online with those calculated offline, we have built their correlation matrices. In the ideal case, one should obtain straight lines passing through the origin of the cartesian coordinates. For instance, correlating the online and offline energy values, for each of the two silicon detectors, we obtain the plots in fig. 6.1. Both sets of data have a linear regression coefficient  $R = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y}$  which differs from one by about 0.1%, indicating a strong linear correlation of the two variables.

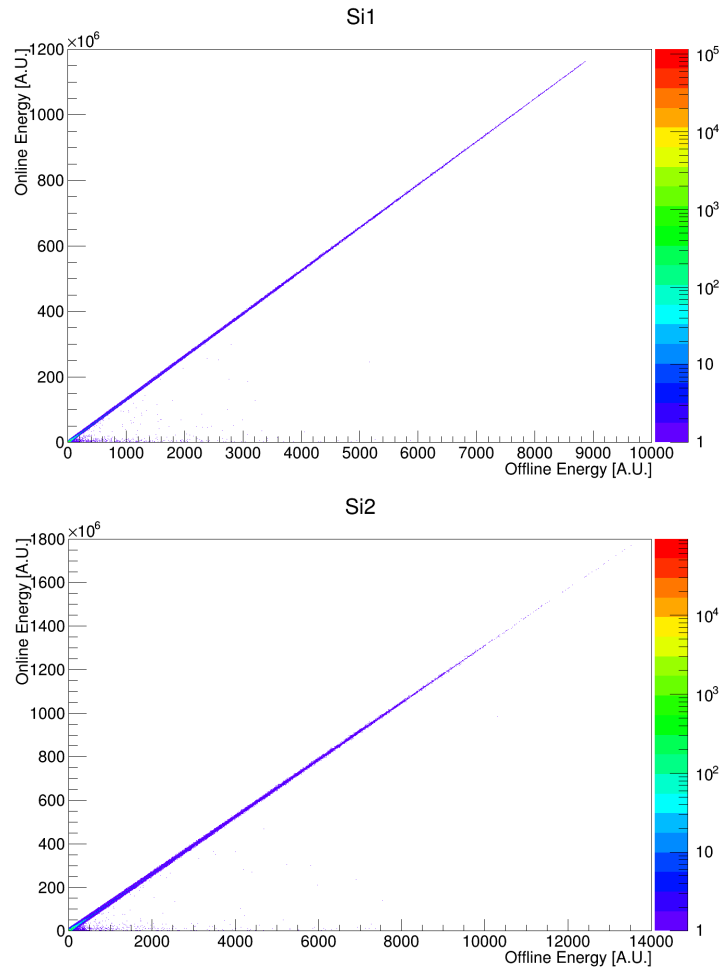


Figure 6.1: Online-offline energy comparison for the two silicon detectors.

A similar plot can be made using the online and offline replica values of the maximum current to check if the real-time interpolating algorithm implemented on the FPGA is working well. Since the maximum current obtained from the sampled current signal is evaluated without using a moving average, it is pointless to compare it with the online value. However, the offline replica of the online process is performed with the moving average and, apart from a scaling factor, it yields results that are compatible with the online

values, as can be seen in fig. 6.2, thus indicating the absence of possible implementation bugs in the firmware. This is also confirmed by the linear regression factor obtained for the two plots, differing from one by about 0.2%.

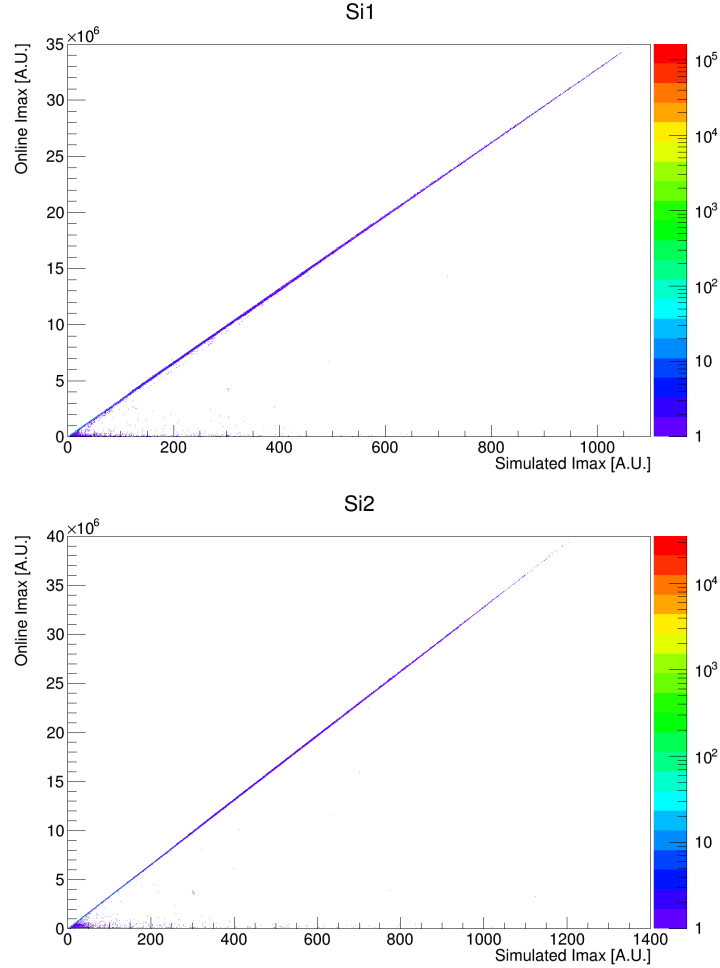


Figure 6.2: Online-offline replica Imax comparison for the two silicon detectors.

### 6.2.2 $\Delta E$ -E matrix and energy calibration

The  $\Delta E - E$  correlation matrices (shown in figs. 6.5 and 6.6) have been used for calibrating the measured energy (which is originally expressed in arbitrary units, namely those of the output amplitude of the trapezoidal shaper). As mentioned in the introduction to this work, a  $\Delta E - E$  correlation presents distinct curves for different isotopes. This result, however, is obtained with the constraint that the impinging particle has to stop in the second detector, e.g. when the range of the impinging particle is less than the total thickness of the two detectors. Since the range is related to the starting energy of the particle,  $E_0$ , this means that  $E_0$  must be lower than a value,  $E_{punch}$ , which depends on the particle type as well as on the two detectors properties. The energy  $E_{punch}$  can be obtained

from energy loss tables, and it is the key to calibration. In fact, by identifying the punch-through points of various isotopes on the  $\Delta E - E$  correlation matrix, one selects those fragments with  $E_0 = E_{punch}(A, Z)$ . Once  $E_0$  is known, it is possible to evaluate (using energy loss tables) the energy deposited in each detector and then, with a fit procedure, to extract a calibration function. This operation has been performed for both the online and the offline data using the punch-through points of medium-light fragments (up to  $^{17}\text{O}$ ) and assuming a linear calibration function having no offset, i.e.  $E [\text{MeV}] = K \cdot E [\text{A.U.}]$ . The error on each point is evaluated by visual inspection of the  $\Delta E - E$  correlation, looking at the spread of each isotopic line near the punch-through point. The online and offline energies, obtained in different ways, are to be calibrated separately. In fig. 6.3 the calibration points for both detectors are plotted with the calibration function obtained. In the two topmost panels the calibration of the offline energy is shown, in the others, the calibration of the online energy. The values of the calibration parameters are reported in tab. 6.1 with their errors, as well as the  $\chi^2/\text{DOF}$  values (in table 6.2). In addition, the calibration residuals for the various points are shown, with their error, in fig. 6.4. Again, the offline data are represented in the two topmost panels.

|         | Si1                             | Si2                             |
|---------|---------------------------------|---------------------------------|
| Offline | $0.1253 \pm 0.0004$             | $0.07934 \pm 0.00004$           |
| Online  | $(9.75 \pm 0.02) \cdot 10^{-7}$ | $(6.05 \pm 0.10) \cdot 10^{-7}$ |

Table 6.1: The calibration parameter for the two silicon detectors for online and offline data. The values are expressed in  $[\text{MeV}]/[\text{A.U.}]$

|         | Si1   | Si2   |
|---------|-------|-------|
| Offline | 3.7   | 21.76 |
| Online  | 13.98 | 3.48  |

Table 6.2: The  $\chi^2/\text{DOF}$  value for the calibration, both for online and offline data.

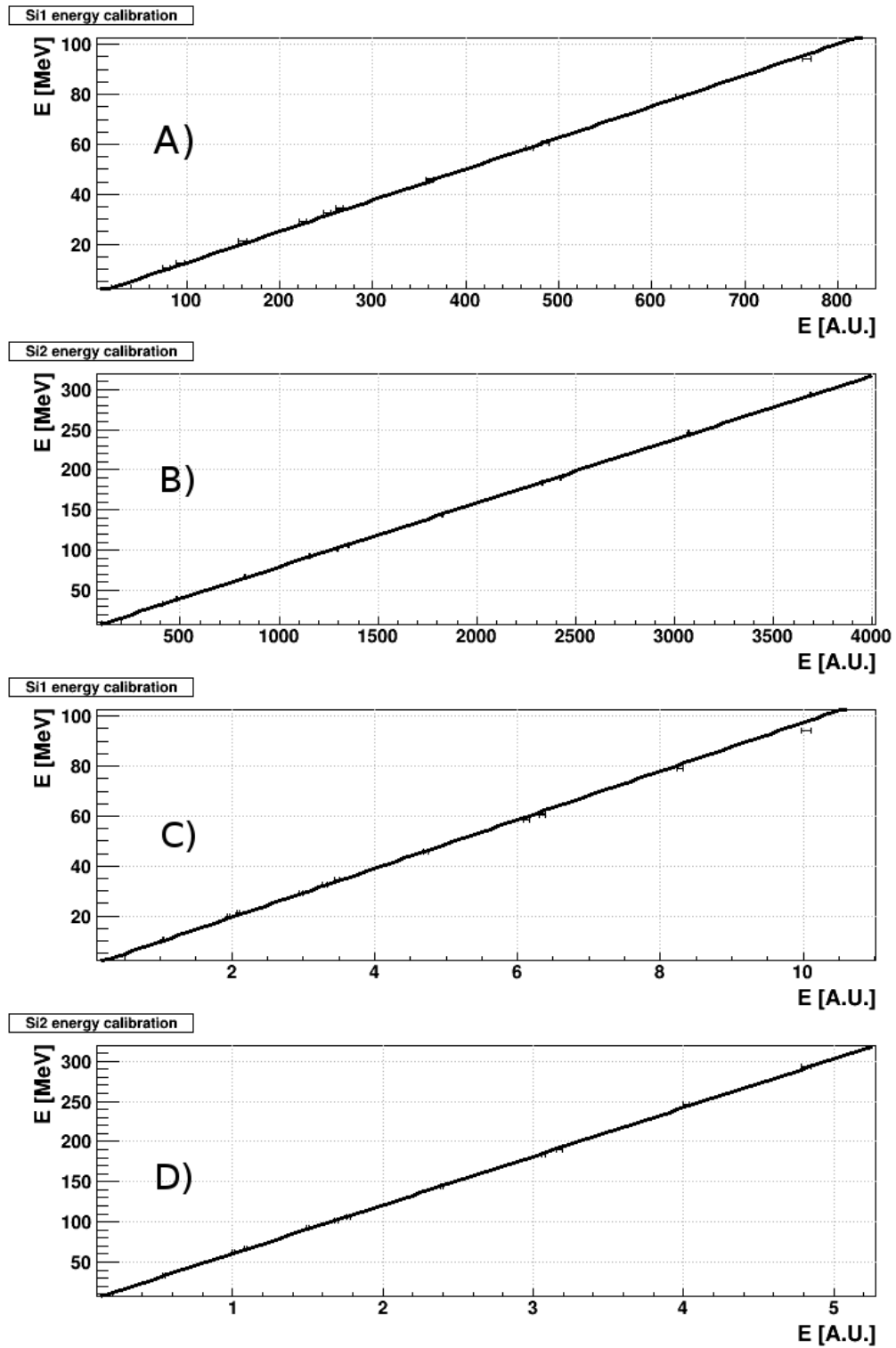


Figure 6.3: The calibration points and functions for the offline energy (panels A and B) and the online energy (panels C and D)

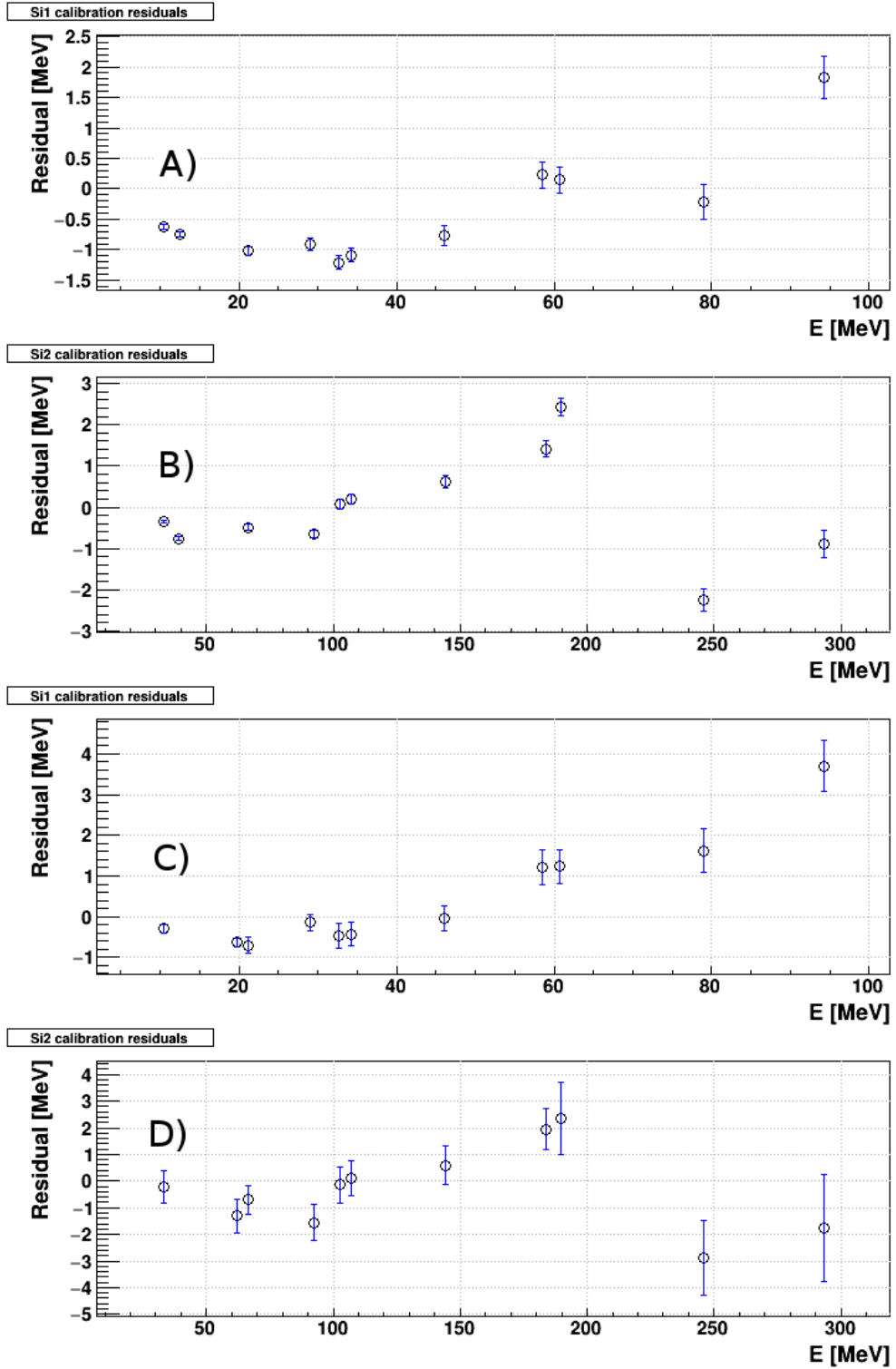


Figure 6.4: The calibration residuals for the offline energy (panels A and B) and the online energy (panels C and D)

### 6.2.3 $\Delta E$ -E plots

In figs. 6.5 and 6.6 the  $\Delta E$ -E plots obtained with the data acquired at LNS are presented. The plots represent the correlation obtained with offline energies, which are evaluated using the sampled charge signals (upper panel) and with FPGA data, produced in real-time by the trapezoidal shaper (lower panel). Fig. 6.6 shows an expanded view of the plots in fig. 6.5, to evidence the isotopic separation for the carbon line ( $Z=6$ ). The energies are calibrated using the procedure described in sec. 6.2.2. To exclude the punch-through events from the correlations, the CsI energy is used as a veto (requiring it to be  $< 10$  arbitrary units for offline correlations, and  $< 10^6$  arbitrary unit for online correlations). To be consistent, in the online plots the punch-through events are vetoed using the online CsI energy. In the full scale plots, the isotopic lines are barely visible, though we are able to view the Z-lines and to have an overview of the detected fragments. Since we are using a  $^{48}\text{Ca}$  beam, we expect to measure, neglecting contributions from the target itself, fragments up to  $Z = 20$ . In fact, due to the polar angle of the detector, higher-Z fragments have not been detected (except for  $Z = 20$ ). The expanded plots show only the Carbon isotopes and their immediate neighbours. In these plots the isotopic separation is evident: there are five isotopic lines for Carbon (from  $^{11}\text{C}$  to  $^{15}\text{C}$ ). By looking at the plots with online and offline data, it seems that the offline data are slightly better than the online data in terms of isotopic resolution.

### 6.2.4 Energy-Imax PSA plots

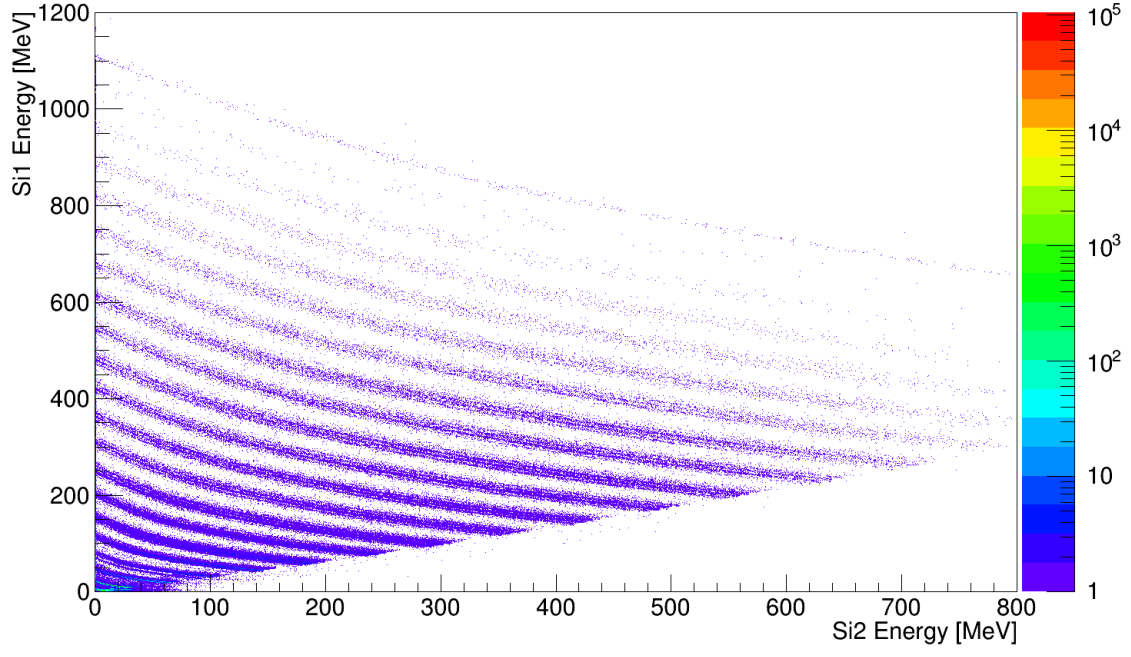
Since PSA exploits the information produced by a single detector, we will present distinct matrices for the two silicon detectors. Furthermore, the current signal shape, thus its maximum, strongly depends on the bias voltage applied to the detector. As for the  $\Delta E - E$  plots, energy information is calibrated using the procedure described in sec. 6.2.2. Si1 detector was biased at 150V for the whole duration of the experiment. In fig. 6.7 the top panel contains the correlation obtained with offline data, the bottom one the correlation obtained from online data. Since the particles must stop in the detector (see sec. 1.2.1) the energy information for Si2 is used to suppress punch-through events (i.e. events when the particle does not stop into Si1). As can be seen from the plots in fig. 6.7, no isotopic resolution is obtained. The lack of isotopic resolution could be due to several reasons, e.g.:

- the actual depletion voltage of Si1 could be lower than 150V so that the detector could be biased at a higher voltage with respect to its depletion voltage;
- the doping uniformity could not be as good as expected

We also note that the moving average filter, which is present only in the online processing, introduces a deformation in the correlation matrices, as can be seen by comparing the two plots in fig. 6.7. Furthermore, in the offline data plot (upper panel) we see the negative effects of the glitches on the system performance, appearing as regions with a higher concentration of point “in between” the correct lines (centered at Imax values of about 350, 600, 800,...).



$\Delta E(\text{Si1})-E(\text{Si2})$  correlation, offline data.



$\Delta E(\text{Si1})-E(\text{Si2})$  correlation, online data.

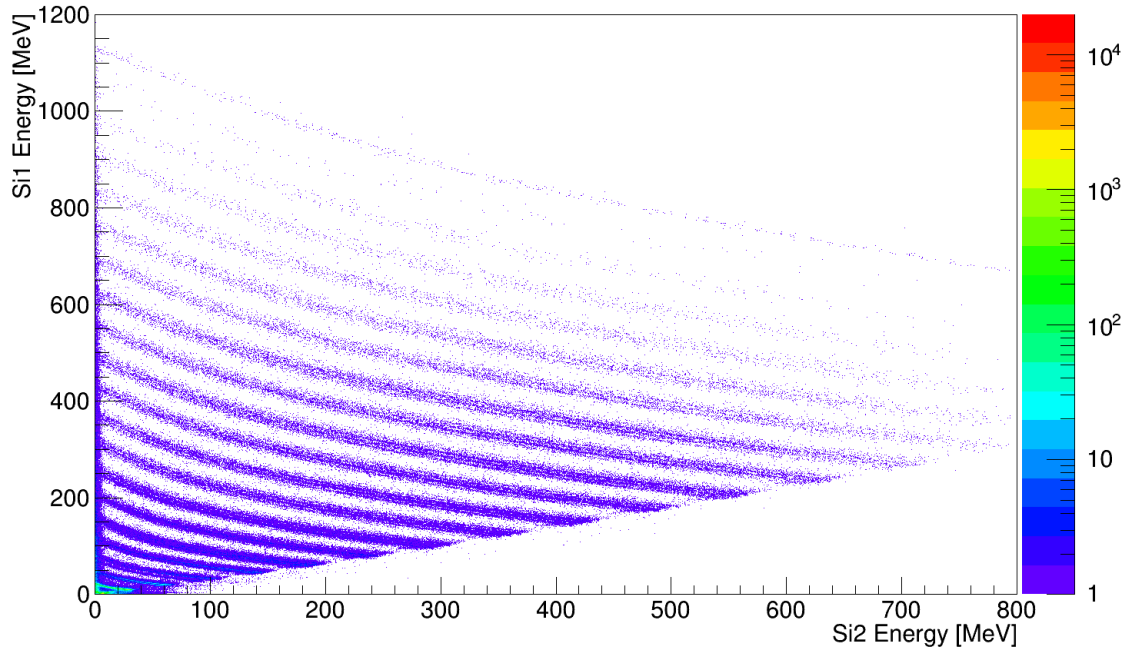
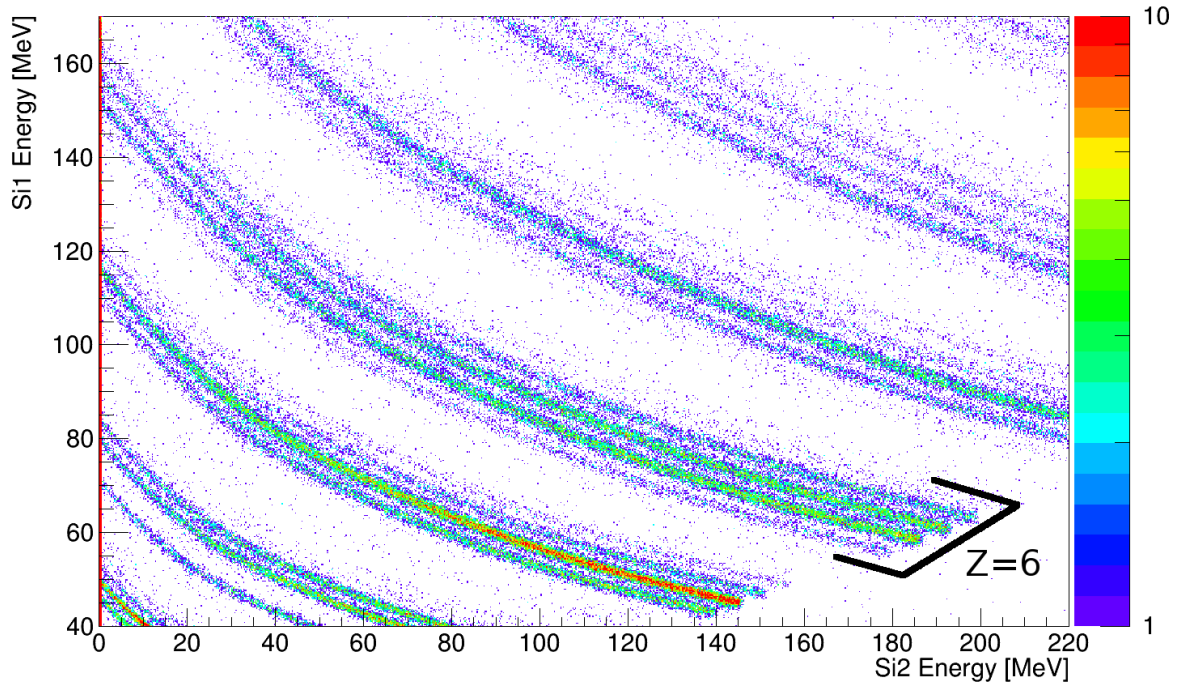


Figure 6.5:  $\Delta E - E$  plots (full scale)

$\Delta E(\text{Si1})\text{-}E(\text{Si2})$  correlation, offline data.



$\Delta E(\text{Si1})\text{-}E(\text{Si2})$  correlation, online data.

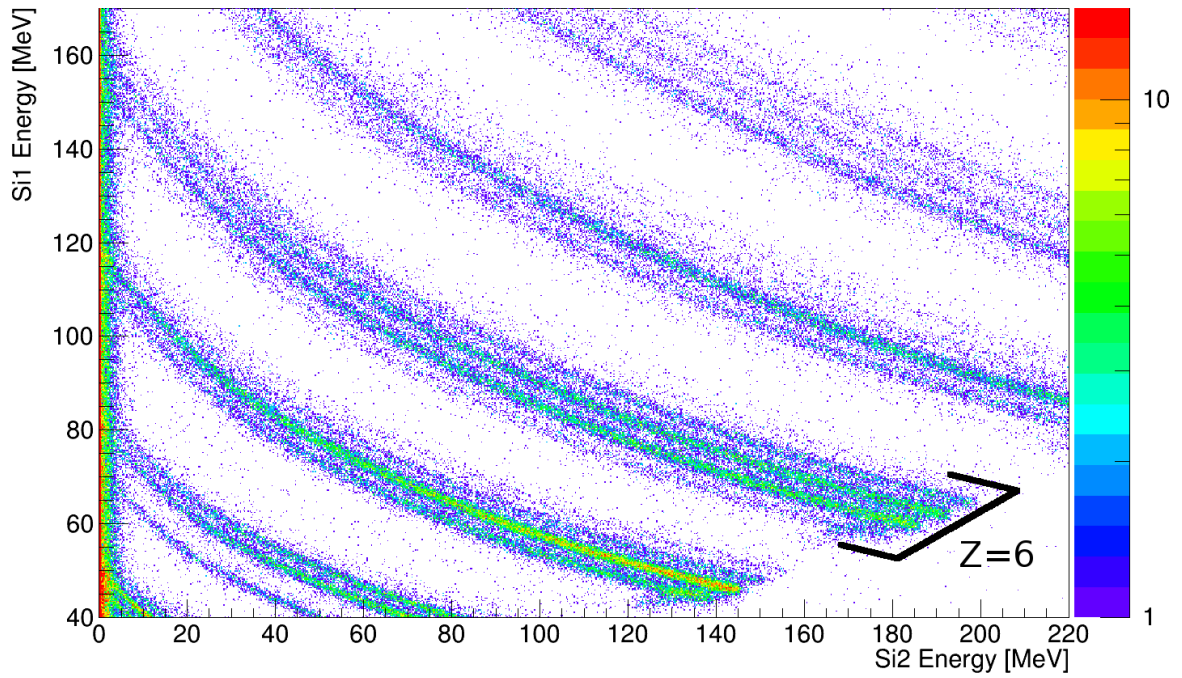
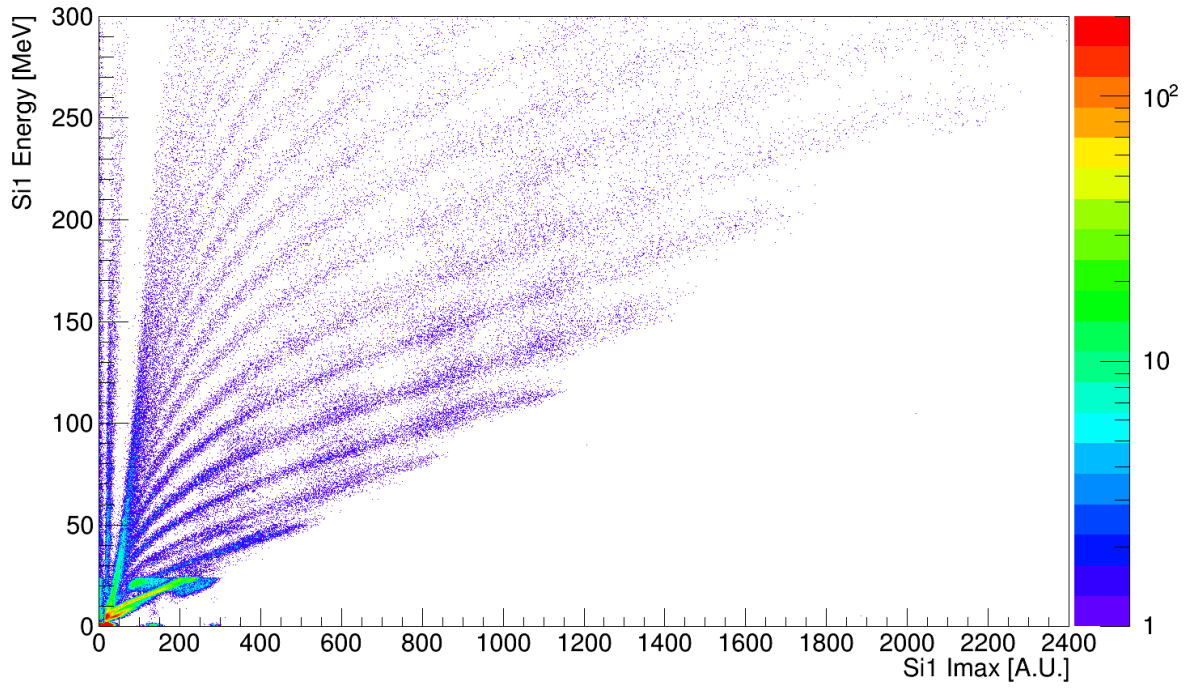


Figure 6.6:  $\Delta E - E$  plots (detail)

Energy-I<sub>max</sub> correlation, offline data, Si1@150V



Energy-I<sub>max</sub> correlation, online data, Si1@150V

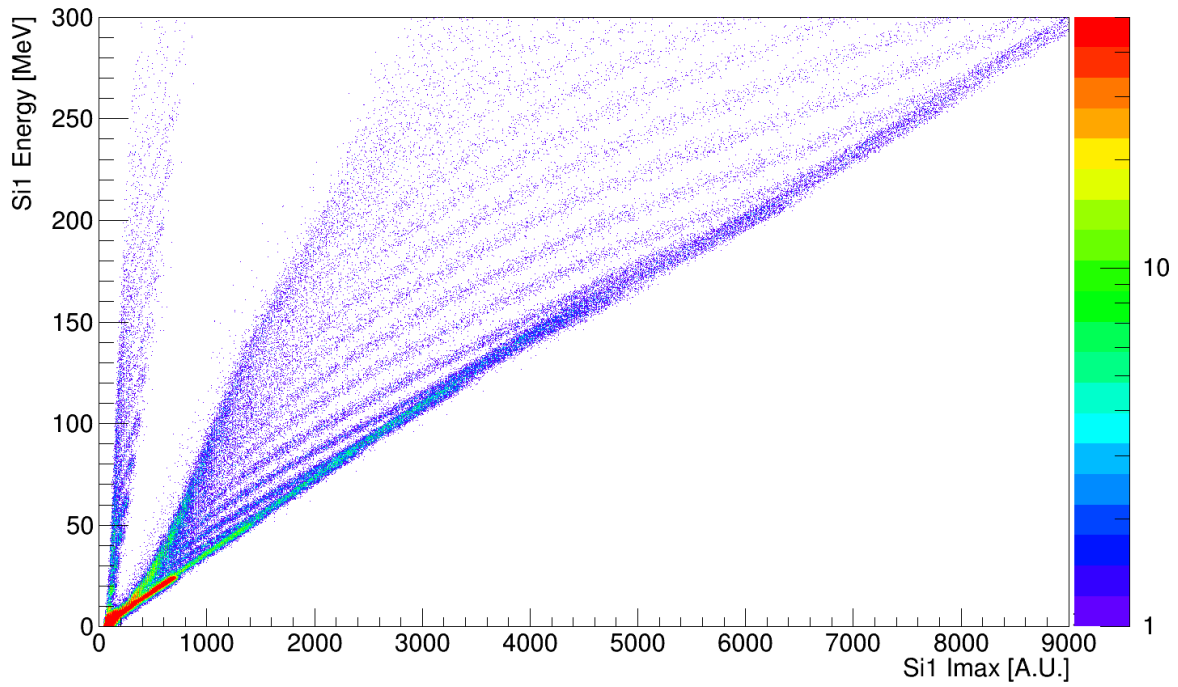


Figure 6.7: Energy-imax plot for Si1 (detail)

The main aim of the experiment was to use data from Si2 for testing the performance of the identification procedure. The correlation matrices obtained with the two bias voltages (220 V and 270 V) confirmed that reduction of the bias voltage increases the isotopic resolution of PSA, though at the cost of increased energy thresholds for identification [14]. Since isotopic separation is marginal for both offline and online data at a bias of 270V, we will present only data taken at a bias of 220V.

We report also the correlation matrices obtained with the offline replica of the FPGA calculation, to have a further test of the correct implementation of the filter in the FPGA firmware. From the two correlations in fig. 6.8 it is clear that the offline replica of the algorithm and the online real-time algorithm give compatible results, so that we can exclude the presence of implementation bugs (or, at least, of bugs so critical to spoil the isotopic resolution of PSA).

Coming to the online-offline comparison, for each set of data (offline, online) three PSA plots are shown:

- full scale (top panels in figs. 6.9 and 6.10);
- expanded scale (bottom panels in figs. 6.9 and 6.10);
- detail of the Carbon isotopes (fig. 6.11).

From the plots in fig. 6.11 one can infer that data obtained from the online FPGA processing give worse isotopic resolution. Though the offline data yield better results, the online results also show a slight isotope resolution, at least for light fragments.

In addition, the online data and the offline replica yield fully compatible results, ensuring that the filter implementation is working as expected.

A more quantitative method to express the quality of PSA identification is the so called "Figure of Merit" (FoM). Before introducing the *FoM*, some details on the particle identification procedure are needed.

As said in chapter 1, the identification of nuclear fragments can be performed using two different techniques ( $\Delta E - E$  and PSA). Both techniques exploit the correlation between two physically relevant variables, to obtain an estimate of the A and Z values of the fragment. From a more formal point of view, every detected fragment will be represented by a pair of values (let's call them  $X$  and  $Y$ ), and it can be represented as a point in  $R^2$ ,  $(X, Y)$ . The identification is then achieved by defining a function  $PI(X, Y)$  that goes from  $R^2$  to  $R$ , assigning to each point of  $R^2$  a single value (PI is a contraction of **P**article **I**dentification) that represents its characteristics. The core of the identification is then the evaluation of the analytical form of the function  $PI(X, Y)$ . Since  $PI(X, Y)$  depends on many factors (radiation-matter interaction, charge collection process, signal filtering etc.), it is usually simpler to extract  $PI(X, Y)$  from the data themselves. By producing a correlation histogram of the two variables for all the acquired events, one obtains correlation matrices such as the one shown in figures 6.6 and 6.8. Then one usually proceeds by graphically defining "isotopic lines", drawn above the ridges in the correlation matrix associated to the different isotopes. To every isotope (thus to every line) a specific value of  $PI$  is assigned.



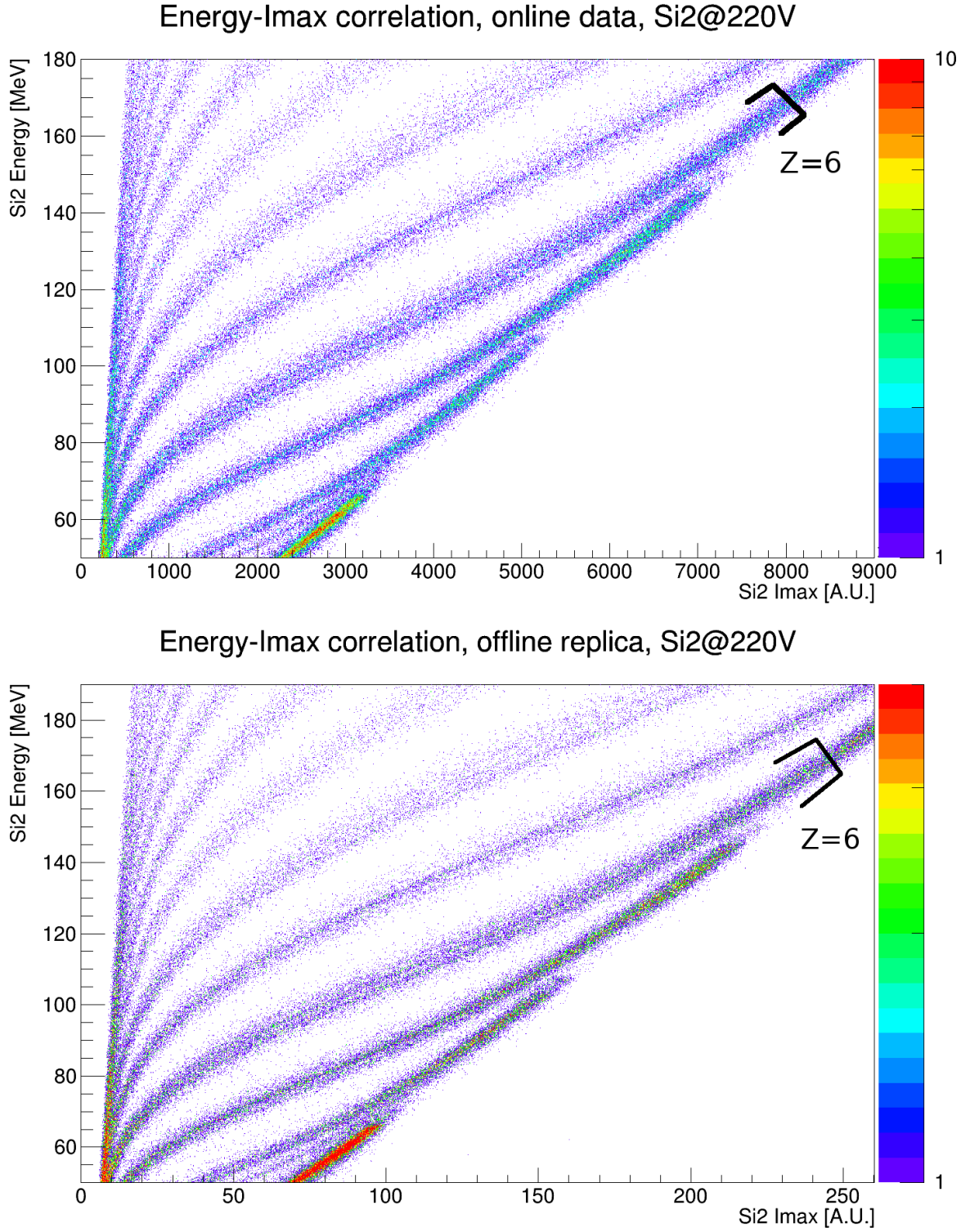


Figure 6.8: PSA plots with online (top) and offline replica (bottom) data. Detail of Carbon isotopes.

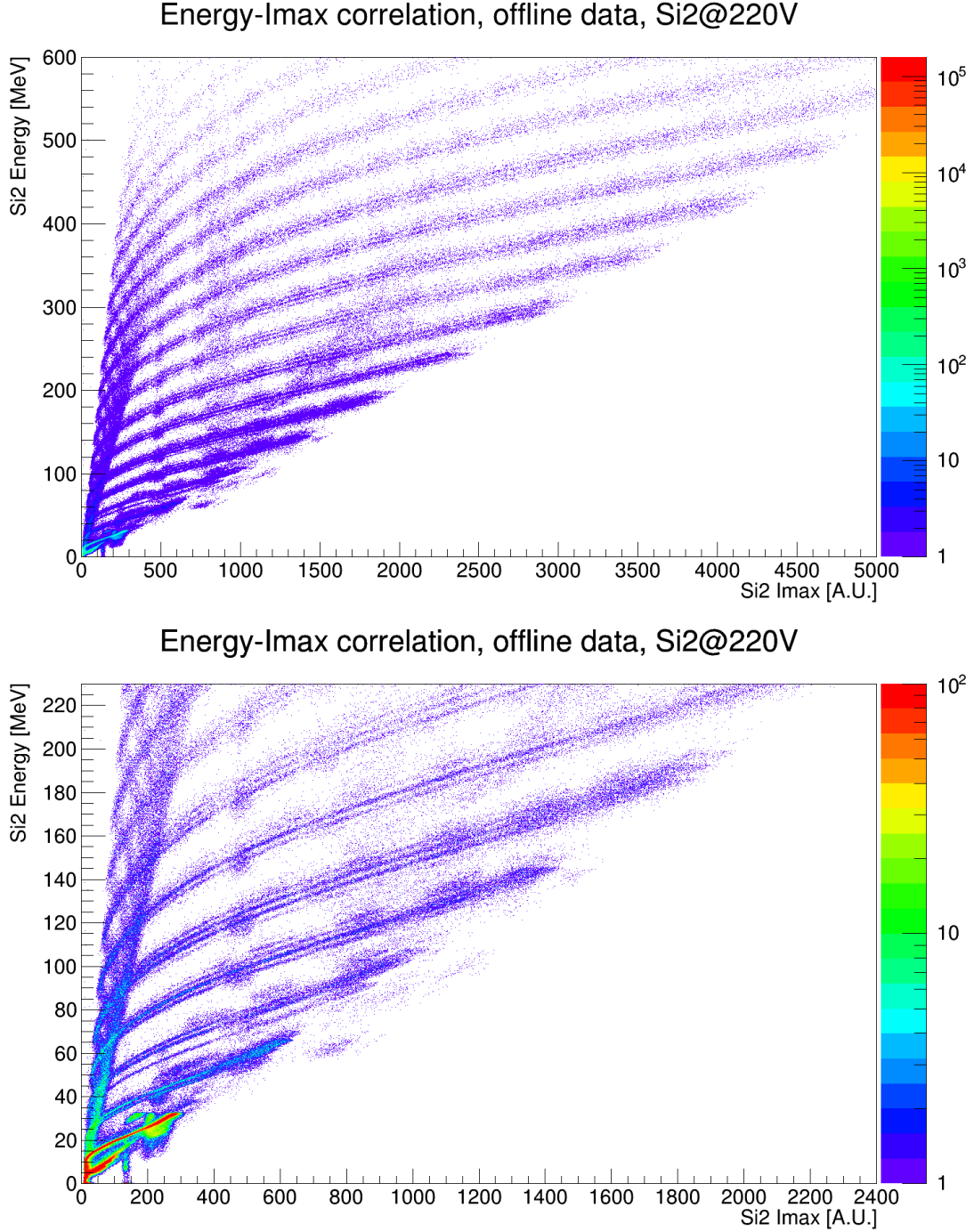


Figure 6.9: PSA plots produced by the offline analysis. The bottom panel shows an expanded view of the top panel. The events outside the main curves (concentrated around  $I_{\max}=400, 800$ , etc.) are due to digital glitches on the stored current signal, not completely recovered by our correction procedure. Isotopes are resolved at least up to  $Z=12$ .

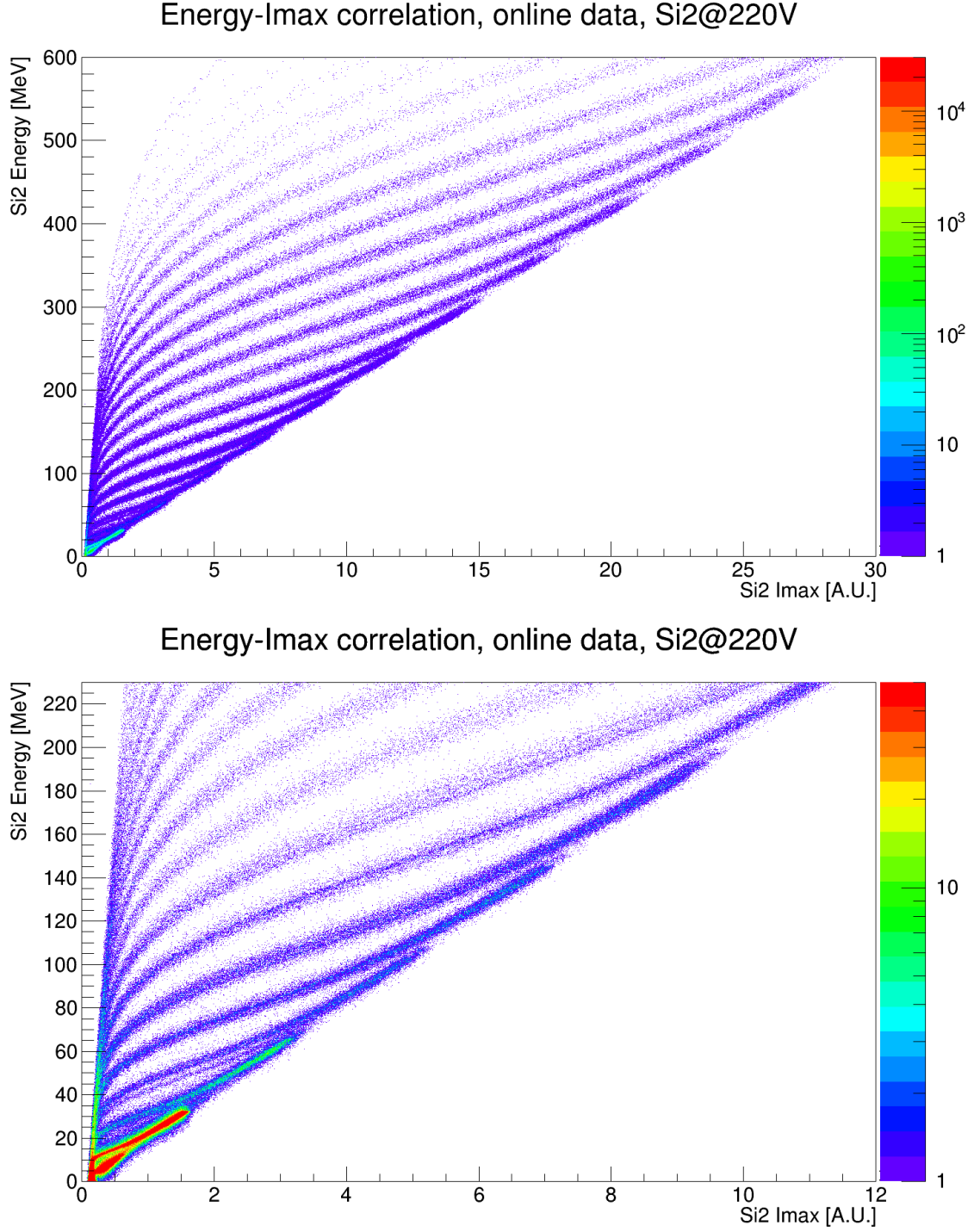


Figure 6.10: PSA plots obtained from the energy and current maximum evaluated in real-time by the FPGA. The bottom panel presents an expanded view of the top panel plot. A hint of isotopic resolution is present for  $Z=5-8$  (though in a limited energy interval).

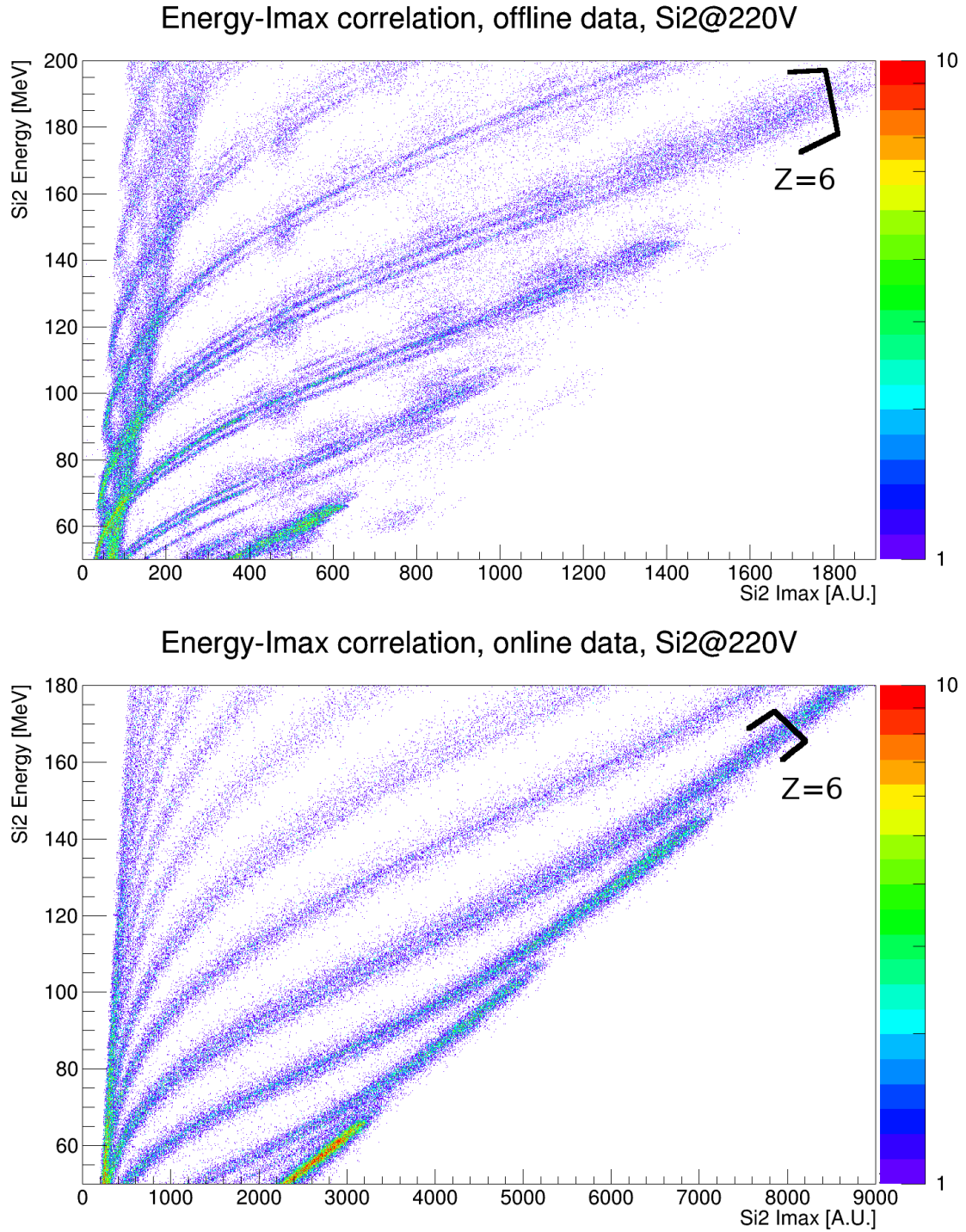


Figure 6.11: PSA plots showing a detail of Carbon line, for offline data (top) and online data (bottom).



In general the  $PI$  value is assigned to a given line using a rule like  $PI = KZ + A$ , where  $K$  is chosen in order to prevent two isotopes to have the same  $PI$ . Usually, one uses  $K = 8$  or  $K = 10$ . In the following we have used  $PI = 8Z + A$ . To obtain the  $PI$  for points that are between different lines, one then uses interpolation. Using the  $PI(X, Y)$  function, leads from a two-dimensional scatter plot to a unidimensional histogram containing the  $PI$  distribution of the measured fragments [34]. Now we can get back to the  $FoM$  definition. The  $FoM$  is a property of two adjacent peaks in the  $PI$  spectrum, defined as [34]:

$$FoM = \frac{C_1 - C_2}{FWHM_1 + FWHM_2} \quad (6.1)$$

where  $C_1$  and  $C_2$  are the centroids of the two peaks and  $FWHM_1$  and  $FWHM_2$  their Full Width at Half Maximum. The  $FWHM$  of a gaussian is equal to 2.35 times its standard deviation. To evaluate the  $FoM$ , one first has to linearize the correlations in fig. 6.11, calculating the  $PI$  value. We have extracted the  $PI$  distribution for the carbon lines, both for offline and online data obtaining the histograms plotted in fig. 6.12. The most prominent peaks are those associated to  $^{12}C$  and  $^{13}C$  isotopes. We fit the adjacent  $^{12}C$  and  $^{13}C$  peaks with the following function:

$$f(x) = g(x; A_1, C_1, S_1) + g(x; A_2, C_2, S_2) + A + Bx + Cx^2 \quad (6.2)$$

where

$$g(x; a, c, s) = \frac{a}{\sqrt{2\pi}s} e^{-\frac{(x-c)^2}{s^2}} \quad (6.3)$$

is a gaussian and the quadratic term  $A + Bx + Cx^2$  has been introduced to get the background counts into account. In fig. 6.12 the best fit are plotted, superimposed over the  $PI$  distribution. The fit parameters, together with the calculated  $FoM$  values, are reported in tables 6.3 and 6.4. According to a rule commonly used within the FAZIA collaboration, two peaks are considered reasonably resolved if their  $FoM$  is at least 0.7<sup>1</sup>. Therefore, though certainly worse than the offline calculations, the  $FoM$  for  $^{12}C$  and  $^{13}C$  obtained from online data is at the limit of acceptability. The main limitation of the online method, as shown in sec. 4.6, is the additional contribution due to sampling noise, which is enhanced by a processing which includes a derivative. Considering that it was not possible to fully optimize the gain in this measurement and that at LNL a reduction in dynamic range of at least a factor of two is expected, one can hope that at GARFIELD the importance of sampling noise will be reduced with respect to the present measurement.

| Offline Data |                    |                   |                 |
|--------------|--------------------|-------------------|-----------------|
| Isotope      | Centroid           | Sigma             | $FoM$           |
| $^{12}C$     | $59.948 \pm 0.003$ | $0.222 \pm 0.005$ | $1.00 \pm 0.02$ |
| $^{13}C$     | $60.916 \pm 0.004$ | $0.188 \pm 0.004$ |                 |

Table 6.3: Fit parameters and  $FoM$  obtained from offline data

<sup>1</sup>This value corresponds, for two gaussian peaks with same area and same  $\sigma$ , to having a peak/valley ratio of 2.

| Online Data     |                  |                 |                 |
|-----------------|------------------|-----------------|-----------------|
| Isotope         | Centroid         | Sigma           | $FoM$           |
| $^{12}\text{C}$ | $59.93 \pm 0.01$ | $0.32 \pm 0.01$ | $0.71 \pm 0.02$ |
| $^{13}\text{C}$ | $60.88 \pm 0.01$ | $0.25 \pm 0.01$ |                 |

Table 6.4: Fit parameters and  $FoM$  obtained from online data

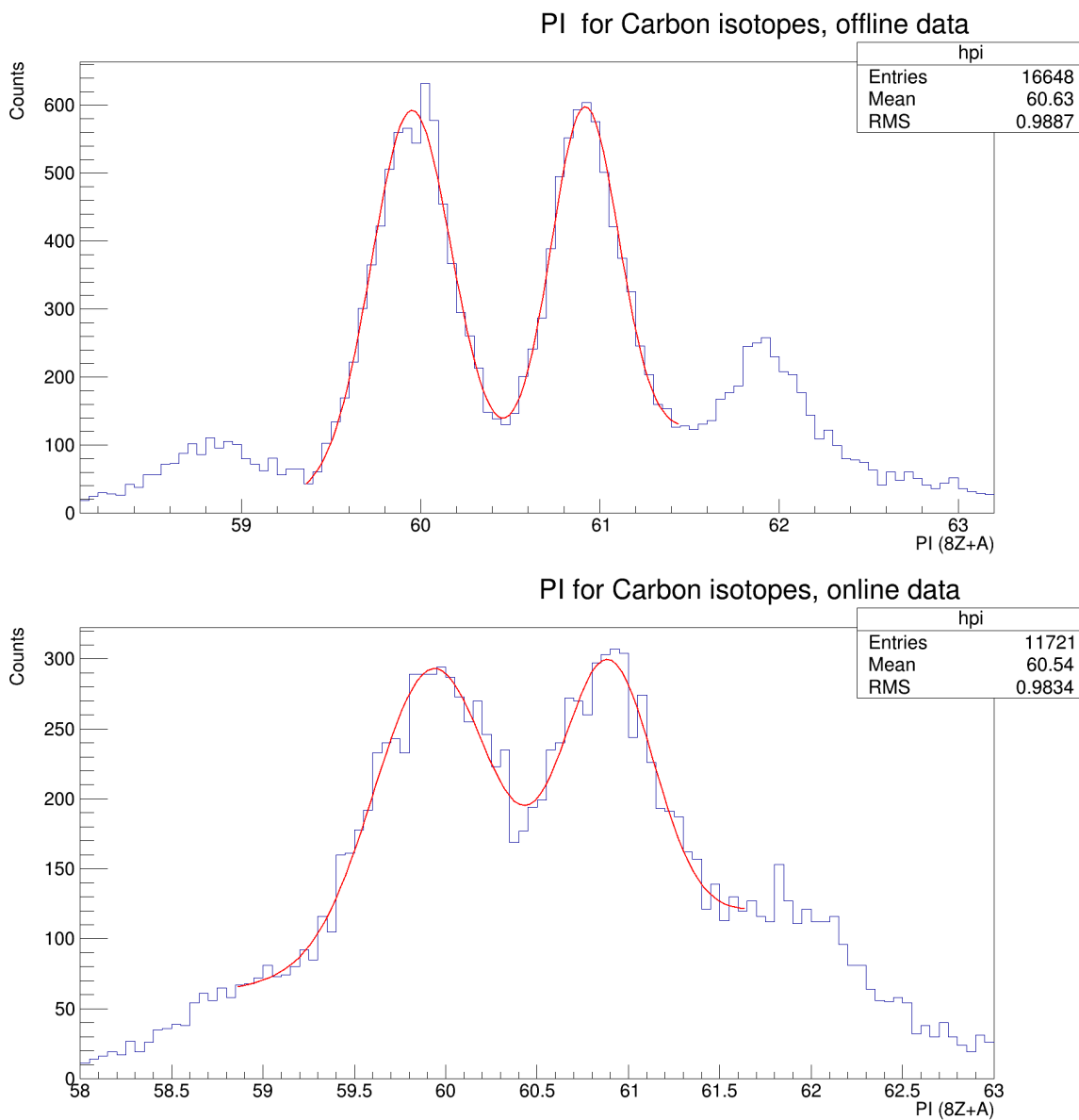


Figure 6.12: The PI distributions obtained with offline data (top) and online data (bottom)

### 6.3 Simulation of the process

To have a better understanding of the obtained results, we have implemented a simple simulation with the aim of comparing the different methods for deriving the current signal from the charge signal. In particular, we are interested in evaluating how the noise and the random phase of the clock with respect to the signal affect the maximum current reconstruction.

As we already did in section 4.4.4, we generate analytical current signals whose shape is defined as follows:

$$i(t; \tau_1, \tau_2, \gamma) = \frac{A}{N} \left( 1 - \left( 1 + \frac{\gamma t}{\tau_1} \right) e^{-\frac{\gamma t}{\tau_1}} \right) e^{-\frac{\gamma t}{\tau_2}} \quad (6.4)$$

where  $N$  is a normalization factor and  $\gamma$  is a time scaling factor. From the current signal, the charge signal is obtained analytically by integration (also taking into account the decay time of the preamplifier,  $\tau_3$ ) obtaining:

$$c(t; \tau_1, \tau_2, \tau_3, \gamma) = \frac{A}{N} \left[ e^{-\frac{t}{\tau_3}} \left( \frac{1}{\alpha_3} - \frac{1}{\alpha_1} - \frac{\gamma}{\tau_1 \alpha_3} \right) + e^{-\alpha_2 t} \left( \frac{\gamma t}{\tau_1 \alpha_3} - \frac{1}{\alpha_3} + \frac{\gamma}{\tau_1 \alpha_3^2} \right) + \frac{1}{\alpha_1} e^{-\frac{\gamma t}{\tau_2}} \right] \quad (6.5)$$

$$\begin{aligned} \alpha_1(\gamma) &= \frac{1}{\tau_3} - \frac{\gamma}{\tau_2} \\ \alpha_2(\gamma) &= \gamma \left( \frac{1}{\tau_2} + \frac{1}{\tau_1} \right) \\ \alpha_3(\gamma) &= \frac{1}{\tau_3} - \alpha_2 \end{aligned}$$

The normalization factor is determined by imposing the condition that the integral of the current signal integral has value  $A/\gamma$ , so that

$$N = \tau_2 - \frac{1}{\alpha_2(1)} - \frac{1}{\tau_1 \alpha_2^2(1)} \quad (6.6)$$

In the simulation, the parameters  $A$ ,  $\tau_1$  (80 ns),  $\tau_2$  (80 ns) and  $\tau_3$  (100  $\mu$ s) are kept constant, while the parameter  $\gamma$  is assigned value  $1/K$  with  $1 \leq K \leq 30$  and  $K$  integer<sup>2</sup>. The simulation loops over all values of  $K$  generating a fixed amount of signals (100000) for each  $K$  value. As it can be seen by inspecting equation 6.4, smaller values of  $\gamma$  means slower current signal (e.g larger charge signal rise-time). Furthermore,  $N$  is constant, since it does not depend on  $\gamma$ , so that the maximum of the current signal will be the same for every value of  $\gamma$  (i.e. generated events all fall on a vertical line in Energy vs max current plots like those of fig. 6.9).

For each generated event, the charge signal is sampled with a clock whose frequency is chosen so that the fastest current signal (that with  $K=1$ ) generates a charge signal with

<sup>2</sup>This values has been choosed in order to have, with  $K = 1$ , a charge signal rise-time of a few samples.

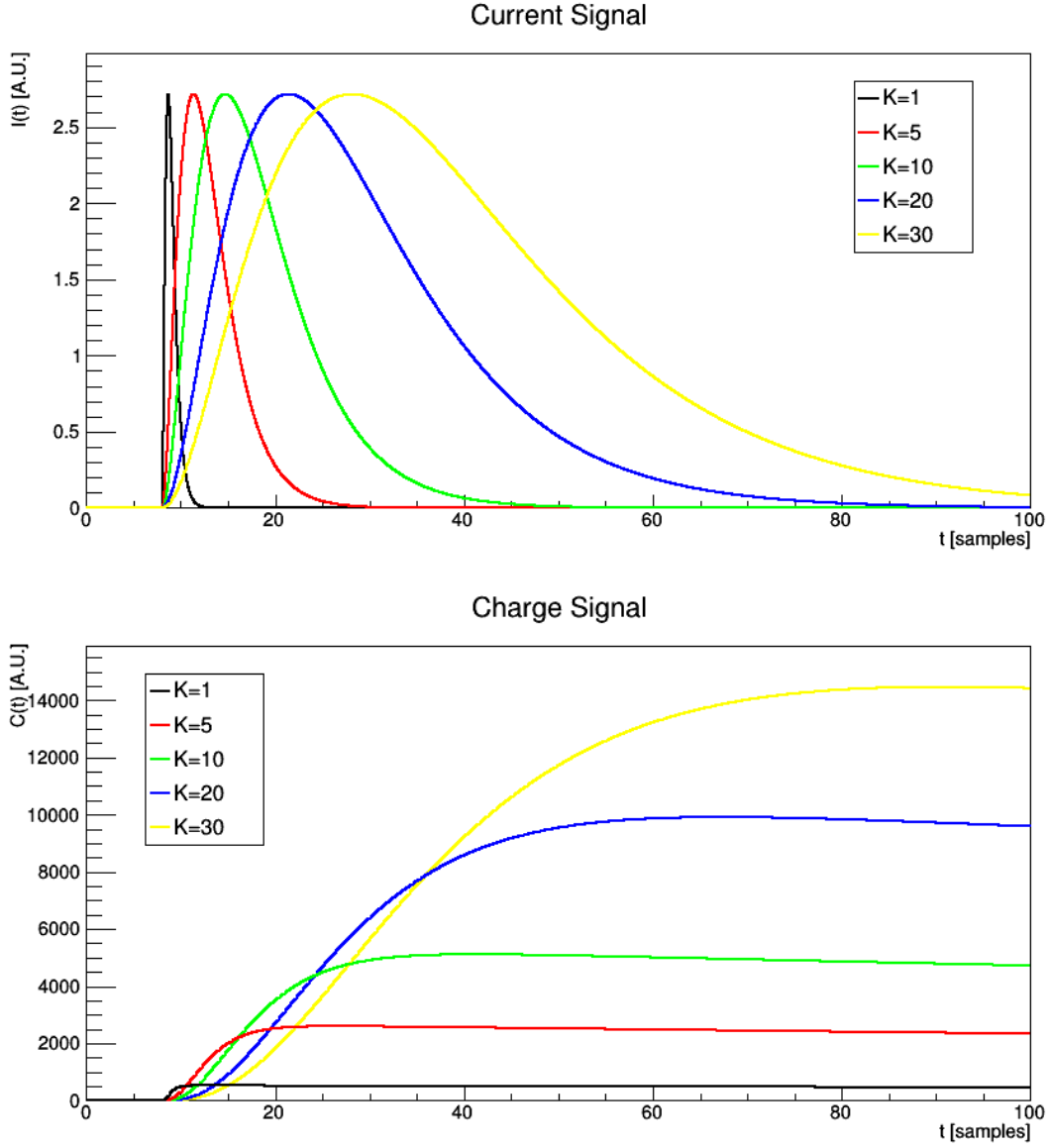


Figure 6.13: Some examples of the current and charge signals obtained with different values of  $K$ , with the parameters described below. The charge signal with  $K = 30$  doesn't have amplitude 16000 due to the deficit introduced by the decay constant of the integrator ( $\tau_3$ )

a rise-time of about two samples<sup>3</sup>. The clock phase is random (uniformly distributed in  $[0, 2\pi]$ ). For each event, several processes are applied to the charge signal:

- first-backward difference (FBD from now on) with Pole-Zero Cancellation;
- same processing as in the previous case, with addition of a 8-samples moving average

<sup>3</sup>The results are comparable with the one obtainable with a 120 MHz clock, using signals with a different time scale.

(MA from now on);

- B-Spline reconstruction (BS from now on) with Pole-Zero Cancellation;
- same processing as in the previous case, with addition of a 8-samples MA;

We have processed signals both with and without added noise. To simulate noise we assume that the ADC input stage noise is the main contribution to the overall noise, so that the noise is white in the frequency domain  $(0, \nu_s/2)$ . We have added gaussian noise with a fixed standard deviation, so that smaller amplitude charge signals have a worse SNR. For each value of  $\gamma$  the charge rise-time (10%-90%) is also calculated.

The amplitude parameter  $A$  is chosen so that the slower current signal (the one with  $\gamma = \frac{1}{30}$ ) has integral 16000, compatible with a full range signal on a 14-bit ADC.

The standard deviation of the noise present on the charge signal has been set to 3 LSB, according to the first measurements performed on our digitizing system. The results are shown in figures 6.14, 6.15 and 6.16. The first plot (fig. 6.14) represents the reconstructed maximum current as a function of the rise-time. For these plots we used the results obtained, in presence of noise, with the four processing proposed above. The average maximum current value measured,  $\langle I_{max} \rangle$ , are divided by the exact maximum current value  $IMAX$ , to make the data interpretation more immediate. Due to noise and to the random clock phase, the measured value will fluctuate from event to event, generating a distribution of values for each settings. The error assigned to each point is nothing but the standard deviation of the related distribution. In the plot the effects of the moving average are evident; in fact, the two plots obtained with MA show:

- a value of  $I_{max}$  significantly different from the real  $I_{max}$  value;
- much smaller errors ( actually smaller than the size of the marker).

Figures 6.15 and 6.15 show the dependence of the  $I_{max}$  resolution on the rise-time of the charge signal. For a gaussian peak, the resolution is defined as:

$$RES = \frac{FWHM}{\langle IMAX \rangle} \quad (6.7)$$

where  $\langle IMAX \rangle$  is the centroid of the distribution.

In the first plot (fig. 6.15), the results obtained in presence of noise are shown, both with and without MA. Again, the positive effects of the MA are apparent, producing much better resolutions. The errors on the resolution are evaluated assuming that  $FWHM$  has no error and propagating the error on  $\gamma$  ( $I_{max}$ ). The second plot (fig. 6.16), shows the result obtained with the B-Spline method only (with and without noise). The worsening of the resolution for decreasing rise-time values is apparent. Since it is present both with and without noise, the effect is probably associated to the random clock phase. The effect of noise is dominant for larger rise times, thus reducing the resolution, though the MA reduces the impact of noise on the reconstructed  $I_{max}$  resolution.

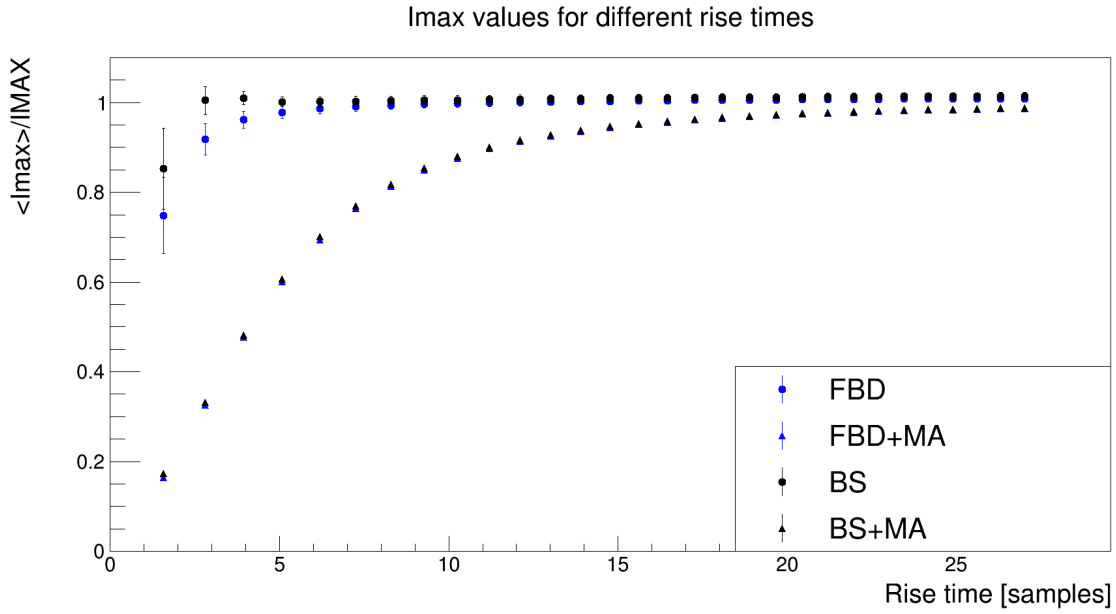


Figure 6.14: Reconstructed I<sub>max</sub> value (normalized to its nominal value I<sub>MAX</sub>) for different risetimes. The errors are evaluated by taking the standard deviation of the I<sub>max</sub> distributions.

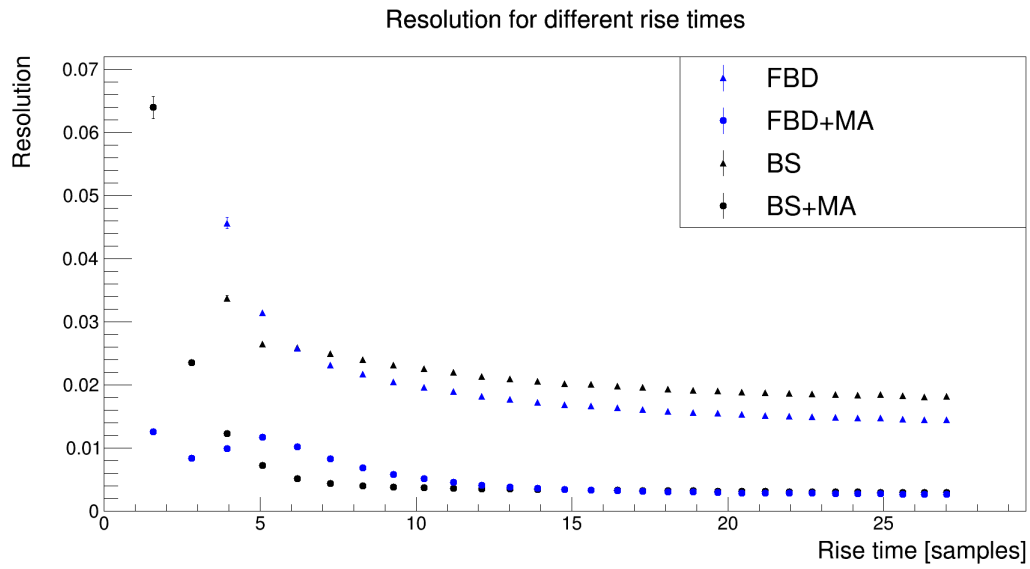


Figure 6.15: Resolution in I<sub>max</sub> obtained for different values of the rise-time.

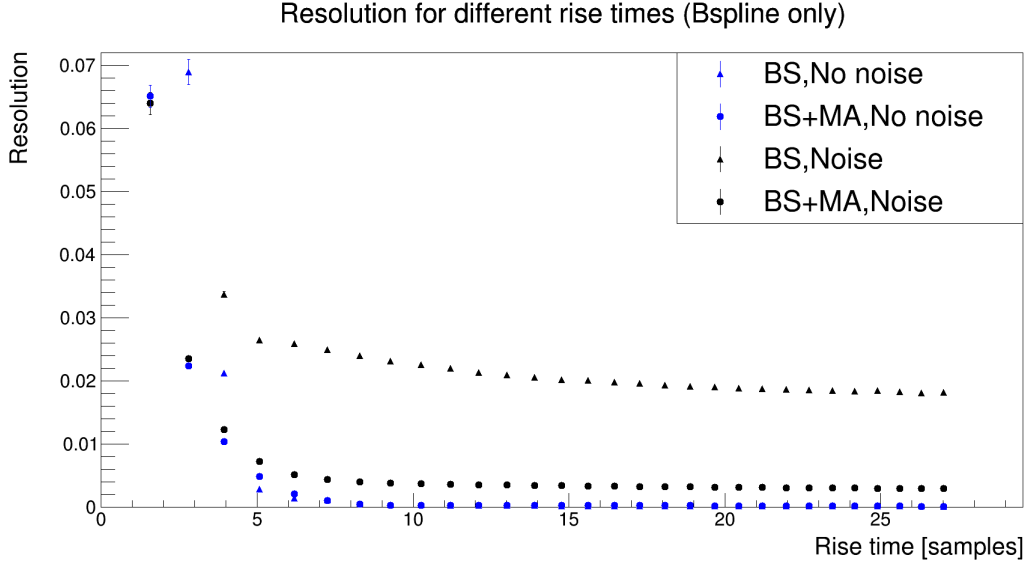


Figure 6.16: Resolution in I<sub>max</sub> obtained using the B-Spline interpolation with and without the presence of noise.

All these considerations are the result of a simulation, which is performed only for a defined set of conditions. In the future, we plan to perform a more comprehensive simulation, to extend these results. However, the present simulation can lead anyway to some useful considerations. In general, the rise-time of the signal falls in an interval which goes from the minimum rise-time compatible with the electronics (e.g. with the anti-aliasing filter) and a maximum rise-time which depends mainly on the detector. Fig. 6.15 tells us that, in a interval of rise-time values between four and fifteen samples, the BS+MA processing produces the best results. If the ADC sampling period ( $\tau_{clk}$ ) is short enough, so that the minimum rise-time falls inside this region, then it will be better to use BS. In our case,  $\tau_{clk}$  is approximately 8 ns and the minimum risetime compatible with the anti-aliasing filter is about 30 ns, e.g. about 4 sampling periods. Therefore, the BS+MA processing should be the best solution for our setup.





## Concluding remarks and future developments

In this work, we have designed and implemented the firmware of a newly developed digitizer for nuclear radiation detectors. Though the digitizer has been designed as a “general purpose” unit, its use with silicon detectors is of particular interest. In fact, digitized signals from silicon detectors, e.g. the current pulse induced on the collecting electrodes by the drifting electron-hole pairs, can be used for identification of nuclear fragments through Pulse Shape Analysis (PSA). In state-of-the-art detector arrays for charged nuclear fragments signals are usually stored on disk and then analyzed offline. The analysis procedure involves interpolation between samples, to better reconstruct the signal shape (e.g. the maximum value of the current pulse). Transferring digitized signals to the acquisition system and storing them to disk has of course an impact on dead time, data bandwidth and storage resources. On-board interpolation and extraction of the relevant information would be a welcome development, provided that the identification performance is preserved. Concluding remarks and future developments The core of this work is actually the design and implementation of a real-time filter for interpolation and differentiation of charge preamplifier signals from Si detectors. The filter allows for a detailed reconstruction of the detector current pulse, sampled at more than 1 GSample/s, starting from a charge signal sampled at 120 MSamples/s. The maximum of the current pulse is then determined online with no need to transfer and store the entire signal. At the beginning of this work, a few months ago, the digitizing board had a minimal firmware. In the last months the firmware has been greatly expanded. A trapezoidal shaper (including PZC) has been introduced in order to extract the energy information from the charge signals. The firmware has been also equipped with a real-time trigger generation system. The realization of the interpolating algorithm has required some time, both for studying the theory and for designing and implementing the real-time calculations.

However, the test under beam performed at LNS in Catania, whose results have been proposed in chapter 6 has given reasonable results.

The data, acquired with a  $\Delta E$ -E prototype telescope of the FAZIA collaboration, feature quite good isotopic resolution in the  $\Delta E$ -E correlations. The PSA performance of the on-line algorithm is somewhat worse than the one obtained from the offline analysis. Only marginal isotopic resolution is obtained, as illustrated in chapter 6. This behaviour can be explained by considering the large dynamic range of the acquired signals and the effect of sampling noise on the interpolation results. As expected from the theoretical calculations, if the ADC contribution to noise is not negligible, differentiating digitized

charge signals gives worse results than sampling the current signals directly. When the expected dynamic range in energy is smaller, like at the GARFIELD apparatus, higher gain charge preamplifier are used. In this situation, the noise coming from the charge preamplifier could dominate the SNR of digitized signals. The PSA performance of the on-line interpolation/differentiation algorithm would be in that case comparable to that of the offline analysis. Since the GARFIELD charge preamplifier and FEE can not produce a current signal, the use of the new digitizers, with the firmware developed in this work, will anyway extend the capabilities of the apparatus, possibly enhancing its identification performance.

## **Firmware modifications**

The firmware used for the measurement at LNS is still a work in progress. Apart from the glitch problem, which is obviously a priority, we plan to add several features to the firmware. First of all, the present firmware must go through an optimization procedure, in order to eliminate timing violations.

Alternative methods to extract the amplitude information from the trapezoidal shaper output signal (as that described in sec. 2.2.2) could be implemented and characterized. We also plan to use the zero-crossing time of the bipolar shaper output signal to perform timing measurements. Other type of time-mark extractions could also be studied and implemented (e.g. Amplitude Rise-Time Compensated timing).

For what concerns interpolation, a more detailed simulation of the interpolation process is required to better characterize the performances of the proposed algorithm. In ref. [3], Unser also suggests a sort of approximated B-Spline interpolation, in which the reconstructed function is not constrained to go through each sample. Such a procedure is claimed to be intrinsically less sensitive to noise. The real-time implementation of such an algorithm could reduce the effect of noise on the reconstructed maximum current value, leading to a better precision and thus to better identification performances.

## Greetings

I would like to thank my girlfriend Dalia, my family and hers for supporting me, in any way, during this months of hard work. A special thanks goes to Stefano Meneghini, for the help he gave me when I was moving my first steps in understanding the functionalities of his original firmware. In addition, I would obviously like to thank the whole Florence experimental Nuclear Physics group, and in particular my supervisor, Gabriele, and Maurizio Bini. Without them, I would never have had the possibility of writing a thesis like this one. Thanks also to the technician of LNS, in particular to D. Rifuggiato, for making the test measurement at LNS possible. In general, i feel like I must thank every people I have met during this months, since everyone has accepted me as a colleague, rather than a student, giving me a great motivational boost. In the end, I must thank my High school physics teacher, Anna Bartolini, which introduced me to the world of Physics about ten years ago.



# Bibliography

- [1] G. Pasquali et al.: *A DSP equipped digitizer for online analysis of nuclear detector signals*, Nucl. Instr. and Meth. A 570 (2007) 126-132
- [2] M. Bruno et al.: *GARFIELD RCO digital upgrade: A modern set-up for mass and charge identification of HI reaction products*, Eur. Phys. Jour. A 2013 49
- [3] M. Unser, A. Aldroubi, M. Eden : *B-Spline Signal Processing: Part I-Theory*, IEEE TSP 1993 41
- [4] M. Unser, A. Aldroubi, M. Eden : *B-Spline Signal Processing: Part II-Efficient Design and Applications*, IEEE TSP 1993 41
- [5] “Four-pi A and Z Identification Array”, <http://fazia.in2p3.fr/>
- [6] G. G. Knoll: *Radiation detection and measurement 3d ed.*, John Wiley & Sons, 2000
- [7] H. Spieler: *Semiconductor Detector Systems*, Oxford University Press, 2005
- [8] W. Seibt, K. E. Sundstrom, P. A. Tove: *Charge collection in silicon detectors for strongly ionizing particles*, Nucl. Instr. and Meth. 113 (1973) 317-324
- [9] N. Le Neindre et al.: *Comparison of charged particle id using PSD and DEE between Front and Rear side Injection*, Nucl. Instr. and Meth. A 701 (2013) 145
- [10] R. Bougault et al.: *The FAZIA project in Europe: R&D phase*, Eur. Phys. J. A 50, 47 (2014)
- [11] H. Hamrita et al.: *Charge and current-sensitive preamplifiers for pulse shape discrimination techniques with silicon detectors*, Nucl. Instr. and Meth. A 531 (2004) 607
- [12] L. Bardelli et al.: *Progresses in the pulse shape identification with silicon detectors within the FAZIA Collaboration*, Nucl. Instr. and Meth. A 654 (2011) 272-278
- [13] S. Carboni et al.: *Particle identification using the  $\Delta E - E$  technique and pulse shape discrimination with the silicon detectors of the FAZIA project*, Nucl. Instr. and Meth. A 654 (2012) 251-263
- [14] G. Pasquali et al.: *Energy measurement and fragment identification using digital signals from partially depleted Si detectors*, Eur. Phys. J. A (2014) 50

## BIBLIOGRAPHY

---

- [15] G. Pasquali: *Appunti del corso Metodi sperimentali per la fisica nucleare*, Dottorato in fisica e astrofisica, Università degli Studi di Firenze
- [16] P. Ottanelli: *Study and characterization of a trapezoidal filter for digitized signal of nuclear physics detectors*, Graduation thesis, University of Florence, A.A 2012-2013
- [17] F. Goulding: *Pulse Shaping in Low-Noise nuclear amplifiers*, Nucl. Instr. Meth. 1972 493
- [18] E. Gatti e P. F. Manfredi: *Processing the Signals from Solid-State Detectors in Elementary-Particle Physics*, Rivista del Nuovo Cimento, vol 9 n.1
- [19] B. Loo, F. Goulding, D. Gao: *Ballistic Deficit in Pulse Shaping Amplifiers*, IEEE TNS 35-1998-114
- [20] G. Poggi: *Appunti sul rumore elettronico*, Università degli studi di Firenze, 2004
- [21] A. Oppenheim, W. Schafer: *Discrete Time Signal Processing*, Prentice Hall Signal Processing
- [22] C. E. Shannon: *Communication in the Presence of Noise*, Proceedings of the Institute of Radio Engineers, v. 37, No. 1, Jan., 1949, pp. 10–21
- [23] M. Unser: *Sampling—50 Years After Shannon*, Proceedings of the IEEE, vol. 88, no. 4, april 2000
- [24] R. G. Keys: *Cubic Convolution Interpolation for Digital Image Processing*, IEEE transactions on acoustics, speech, and signal processing, vol. assp-29, no. 6, december 1981
- [25] H. S. Hou, H. C. Andrews: *Cubic Spline for Image Interpolation and Digital filtering*, IEEE transactions on acoustics, speech, and signal processing, vol. assp-26, no. 6, december 1978
- [26] ADC AD9255 datasheet, <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9255.pdf>
- [27] L. Bardelli et al.: *Time measurements by means of digital sampling techniques: a study case of 100 ps FWHM time resolution with a 100 MSample/s, 12 bit digitizer*, Nucl. Instr. and Meth. A 521 (2004) 480-492
- [28] DSP ADSP 218x Datasheet, <http://www.analog.com/media/en/technical-documentation/data-sheets/ADSP-2181.pdf>
- [29] CycloneIII EP3C25F256C8N datasheet, <https://www.altera.com/products/fpga/cyclone-series/cyclone-iii/support.html>
- [30] Altera user guide, [https://www.altera.com/zh-CN/pdfs/literature/ug/ug\\_altmult\\_add.pdf](https://www.altera.com/zh-CN/pdfs/literature/ug/ug_altmult_add.pdf)

- [31] Altera user guide, [https://www.altera.com/en\\_US/pdfs/literature/ug/altera\\_pll.pdf](https://www.altera.com/en_US/pdfs/literature/ug/altera_pll.pdf)
- [32] IEEE Standard VHDL Language Reference Manual, IEEE Std 1076-2008
- [33] <http://root.cern.ch>
- [34] S. Carboni: *Studio di problemi connessi con l'utilizzo dell'analisi di forma dei segnali per l'identificazione di ioni di alcuni MeV/u arrestati in silicio*, Master Thesis, 2007/2008