Regadío, A., Sánchez-Prieto, S. & Tabero, J. 2014, "Synthesis of optimal digital shapers with arbitrary noise using simulated annealing", Nuclear Instruments and Methods in Physics Research Section A, vol. 738, pp. 74-81

Available at https://doi.org/10.1016/j.nima.2013.11.099

Synthesis of optimal digital shapers with arbitrary noise using simulated annealing

Alberto Regadío[a,b,*], Sebastián Sánchez-Prieto[a], Jesús Tabero[b]

[a] *Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain*
[b] *Electronic Technology Area, Instituto Nacional de Técnica Aeroespacial, 28850 Torrejón de Ardoz, Spain*

Abstract

This paper presents the structure, design and implementation of a new way of determining the optimal shaping in time-domain for spectrometers by means of simulated annealing. The proposed algorithm is able to adjust automatically and in real-time the coefficients for shaping an input signal. A practical prototype was designed, implemented and tested on a PowerPC 405 embedded in a Field Programmable Gate Array (FPGA). Lastly, its performance and capabilities were measured using simulations and a neutron monitor.

Keywords: Spectroscopy, Noise, Shaping, Adaptive, Digital Signal Processing, Simulated Annealing

1. Introduction

In spectroscopy, the information about incident particles can be extracted from the peak amplitude of the input pulses coming from particle detectors. This method is called Pulse Height Analysis (PHA) and provides a value proportional to the incident particle energy. Thus, identical particles with the same energy must generate identical peak values. The resolution of these systems is affected by noise. In spectroscopy, the noise is classified into three types: white series, white parallel and $1/f$ noise [1]. The $1/f$-parallel noise [2] is not considered in this work as the contribution is negligible in all electronic devices used in modern front-end electronics. On one hand, each type of noise has a spectral density that depends on the type of detector and the features of the spectroscopy system. In fact, some of these three noise types could be negligible depending on the type of detector and spectroscopy electronics. On the other hand, spectroscopy systems have filters at the output of particle detectors or preamplifiers called shapers. A basic feature of shapers is their capability to filter out noise. This capability is generally measured using noise indices [3]. Thus, the noise index for each type of noise must be considered. As such, spectroscopy systems are affected by both noise spectral density and the noise index of the selected shaper. The Signal/Noise Ratio (SNR)

*Corresponding Author
Email addresses: aregadio@srg.aut.uah.es (Alberto Regadío), ssanchez@srg.aut.uah.es (Sebastián Sánchez-Prieto), taberogj@inta.es (Jesús Tabero)

is generally measured using the Full Width at Half Maximum (FWHM) or the Equivalent Noise Charge (ENC).

For each time-invariant spectroscopy system, at least one optimal shaper exists. The optimal shaper depends on the spectral density of each noise type. There exist methods to calculate the optimal shaper, one of the most popular is described in [1]. However, the complexity of this method sometimes imply that optimal shapers were selected using other procedures (e.g. [4]). In this article, an readily implementable optimal algorithm based on simulated annealing to find out automatically a shaper that filter the noise efficiently is developed.

This paper is structured as follows. Section 2 presents the fundamentals of the simulated annealing algorithm and the cost functions used in this work. Section 3 provides details of the FPGA platform, covering both, the hardware and the software. Section 4 presents the theoretical results of the simulated annealing. Section 5 presents the experimental results. Finally, Section 6 covers the conclusions and the future work.

2. Simulated annealing

Simulated annealing [5] is a technique for combinational optimization problems, such as minimizing functions of many variables. This technique was introduced by Kirkpatrick et al. [6] and it was motivated by an analogy to the statistical mechanics of annealing in solids. To understand why such a physics problem is of interest, we may consider how to coerce a solid into a low energy state. A low energy state usually means a highly ordered state, such as crystal lattice. To reach this state, the material is annealed: heated to a temperature that permits many atomic rearrangements and then cooled slowly until the material freezes into a good crystal. Thus, simulated annealing offers an appealing physical analogy for the solution of optimization problems, and more importantly, the potential to reshape mathematical insights from the domain of physics into insights for real optimization problems.

Interest in such algorithms is intense because few important combinational optimization problems can be solved exactly in a reasonable time. For our purposes, a combinational optimization problem is one in which we seek to find some configuration of parameters that minimizes a given function which is usually referred to as the *cost function*. This function is a measure of goodness of a particular configuration of parameters. The election of an appropriate cost function is crucial for achieving good results using this algorithm.

The simulated annealing is an iterative algorithm. In each iteration, it generates some random perturbation, such as moving a particle to a new location. The random perturbations are proportional to a simulated temperature T. Thus, at higher temperatures, the probability of large moves in energy is large; at low temperatures the probability is small. If the cost function is reduced, the new configuration is accepted as the starting point for the next move.

2.1. Proposed simulated annealing algorithm

In order to obtain an optimal shaper using this algorithm, the following steps are to be taken:

1. Establish the sampling period T_s and the shaping time interval $\tau_{\text{range}} = \{N_{\min}T_s, \ldots, N_{\max}T_s\}$, where the set $\{N_{\min}, \ldots, N_{\max}\} \in \mathbb{N}$ and $N \in [N_{\min}, N_{\max}]$ is the shaper order equal to:

$$N = \frac{\tau_s}{T_s} \tag{1}$$

2. Establish the number of temperature steps $T, \ldots, 0$ and the population P of individuals for each temperature step.
3. For each temperature step:
   (a) Generate a population of P individuals. In this work, and in order to reduce the processing time, we assume that individuals follow a monotonically increasing function until they reach the maximum, and then they follow a monotonically decreasing function. Thus, for each individual:

$$I = \left\{x_1, x_2, \ldots, x_{N/2}\right\} \mid 0 \leq x_1 \leq x_2 \leq \ldots \leq x_{N/2} = 1 \tag{2}$$

   The shaper works as a digital Finite Impulse Response (FIR) filter. Thus x_n are the coefficients of the FIR filter.

   (b) Generate a shaper for each individual. In this paper, only symmetrical shapers are considered. Thus, the generated shaper is equal to:

$$\varsigma = \left\{x_1, x_2, \ldots, x_{N/2} = 1, \ldots, x_2, x_1\right\} \tag{3}$$

   (c) Combine ς with the current best shaper. The result will be S. The weight of ς with respect to the best shaper is proportional to T. If there is no current best shaper, S is not combined and $\varsigma = S$.

   (d) Evaluate S according to a cost function previously selected (see Section 2.2). If the cost function of the new shaper is lower than the cost function of the current best shaper, then the current best shaper is S.
4. At the end of the process, the optimal shaper will be the current best shaper.

For all the shapers considered, the flat-top duration is equal to T_s. When considering flat-tops with a duration of τ_t clock cycles, a number of ones equal to $L = \tau_t/\tau_s$ must be added in the middle of ς when attempting to generate the shaper using the individual. In this case:

$$\varsigma = \left\{x_1, x_2, \ldots, x_{N/2-L/2} = 1, \ldots, x_{N/2+L/2} = 1, \ldots, x_2, x_1\right\} \tag{4}$$

However, it is important to take into account that an increasing of the flat-top of a shaper implies an increase of parallel and $1/f$ noise.

The C-pseudocode of this algorithm is the following

$N_{\text{range}} \Leftarrow \{N_{\min} \ldots N_{\max}\}$;
bestDuration $\Leftarrow$ meanValue(N_{range});
bestShaper $\Leftarrow \{0 \ldots 0\}$;
bestMark $\Leftarrow \infty$;
for $i = 1$ **to** T — for each temperature step
    for $j = 1$ **to** P
        Ntmp $\Leftarrow$ bestDuration + GenerateRandomNumber($N_{\min} \ldots N_{\max}$)$/i$;
        if Ntmp $> N_{\max}$ **then**
            $N \Leftarrow$ Ntmp;
        else if Ntmp < 1 **then**
            $N \Leftarrow 1$;
        else
            $N \Leftarrow$ Ntmp;
        end if;
        $I \Leftarrow$ GenerateRandomIndividual(N);
        $\varsigma \Leftarrow$ Shaper(I);
        $S \Leftarrow$ bestShaper $+\varsigma/i$;
        if CalculateFunctionCost(S)$<$ bestMark **then**
            bestMark $\Leftarrow$ CalculateFunctionCost(S);
            bestDuration $\Leftarrow N$;
            bestShaper $\Leftarrow S$;
        end if;
    end for;
end for;

2.2. Cost functions

In this work, the cost function is the ENC for theoretical examples whereas for the real test, the cost function is the SNR.

2.2.1. ENC

As introduced in Section 1, the following three types of noise are considered for calculating the ENC: series noise, parallel noise and $1/f$ noise. Since the individual noise contributions are random and uncorrelated, they add in quadrature. Therefore, and according to [7], the ENC is equal to:

$$\mathrm{ENC}^2 = \frac{1}{2} i_n^2 N_S^2 + \frac{1}{2} v_n^2 C_i^2 N_\Delta^2 + \frac{1}{2} {v_{fn}}^2 C_i^2 N_F^2 \tag{5}$$

where v_n, i_n and v_{fn} are the spectral density of white series, white parallel and $1/f$ noise, respectively. C_i is the sum of all shunting capacitances of the input. Finally, N_S^2, N_Δ^2 and N_F^2 are the noise indices for white series, white parallel and $1/f$ noise, respectively defined in [7] by $N_S^2 = 2F_i T_s$, $N_\Delta^2 = 2\frac{F_v}{T_s}$ and $N_F^2 = 2F_{vf}$.

Traditionally, the calculation of the white series and white parallel noise versus the output of the shaper is usually performed in time-domain [3, 7] whereas the contribution due to the $1/f$ noise is calculated in frequency-domain. However, in this paper the noise analysis is carried out in time-domain due to its simplicity compared to the frequency domain. This analysis is valid for any detector/preamplifier/analog filtering/ADC/PHA combination.

When analog electronics are used to implement the signal conditioning, the analog noise indices in the time-domain for white serial N_Δ^2 and parallel noise N_S^2 are defined in [3]:

$$N_S^2 = \frac{1}{S^2} \int_0^{\tau_s} w^2(t) dt \tag{6}$$

$$N_\Delta^2 = \frac{1}{S^2} \int_0^{\tau_s} {w'}^2(t) dt \tag{7}$$

where τ_s is the shaping time, S is the maximum amplitude of the shaper and $w(t)$ is the weighting function of the shaper. For time-invariant shapers, $w(t)$ is equal to the step response of the system [3] given by the x_n coefficients of Eq. (2).

A noise index for $1/f$ noise N_Δ^2 could be found using fractional derivatives, which can be calculated as proposed in [8]. That is,

$$w^{(1/2)}(t) = g(t) * w'(t) \tag{8}$$

where

$$g(t) = \begin{cases} 1/\sqrt{\pi t} & \text{if } t > 0, \\ 0 & \text{if } t \leq 0, \end{cases} \tag{9}$$

Therefore,

$$N_F^2 = \frac{1}{S^2} \int_0^{\infty} \left(\frac{1}{\sqrt{\pi t}} * w'(t) \right)^2 dt \tag{10}$$

These noise indices are only suitable for analog shaping. To discretize these formulae it must be noted that the overall weighting function performed by the measuring setup is the convolution of the analog weighting function with the digital one. According to [9], the "digital" weighting function $w[n]$ has a time continuous counterpart, a delay line tapped at suitable positions. In the digital domain, $w[n]$ can be observed only at the sampling time interval. Following this method, it is clear that, when sampling the preamplifier output, $w[n]$ is just a staircase with steps lasting one sampling interval and in general with variable height. The effects of the shaping on white series and parallel noise are therefore rapidly evaluated. Thus, the noise indices for white parallel N_S^2 and series N_Δ^2 noise are

$$N_S^2 = \frac{1}{S^2} \sum_{n=0}^{\tau_s/T_s} w^2[n] \, T_s \tag{11}$$

$$N_\Delta^2 = \frac{1}{S^2} \sum_{n=1}^{\tau_s/T_s} \left(w[n] - w[n-1]\right)^2 \frac{1}{T_s} \tag{12}$$

These formulae indicate that the shaping time τ_s is directly proportional to the white parallel noise and inversely proportional to the white series noise. It occurs in both analog and digital shaping.

In the same way, a noise index for $1/f$ noise could be obtained discretizing Eq. (8), that yields:

$$N_F^2 = \frac{1}{S^2} \sum_{n=1}^{\infty} \left(\frac{1}{\sqrt{\pi n T_s}} * (w[n] - w[n-1]) \right)^2 T_s \tag{13}$$

Although T_s and τ_s are part of Eq. (13), N_F^2 is independent of τ_s. It occurs in both analog and digital domains. Furthermore, T_s is proportional to N_F^2 in both analog and digital formulae. This effect was evaluated for analog filters in [10]. When $T_s \to 0$, digital noise indices are equal to analog indices.

To check that Eq. (13) is correct, the value of N_F^2 for several shapers were calculated in Fig. 1. These shapers include triangular, peak, cusp-like and optimum for $1/f$ noise [11]. As we can see, N_F^2 does not depend from τ_s. Besides, the noise indices are in the order of the values presented by [13]. However, the noise indices of this figure and the cited article differ as alternative shaper parameters were used to calculate them. Furthermore, as can be observed, when the peak shaper is not considered, the lowest value of N_F^2 is achieved with the optimum shaper for $1/f$ noise [11]. The peak shaper has a lower N_F^2 since $\tau_s = T_s$. Finally, the small differences between N_F^2 when T_s changes is because signals at the output of shapers vary with T_s.

These formulae (11, 12, 13) are used to obtain the three noise indices of Eq. (5) and then use the ENC to calculate the cost function for theoretical experiments. However, it is important to take into account that, for digital systems, the ENC could be increased by sampling effects. That is, the sampling of any signal different to the step pulse, implies an additional increment of ENC depending on the shape of the digitalized signal and T_s [12].

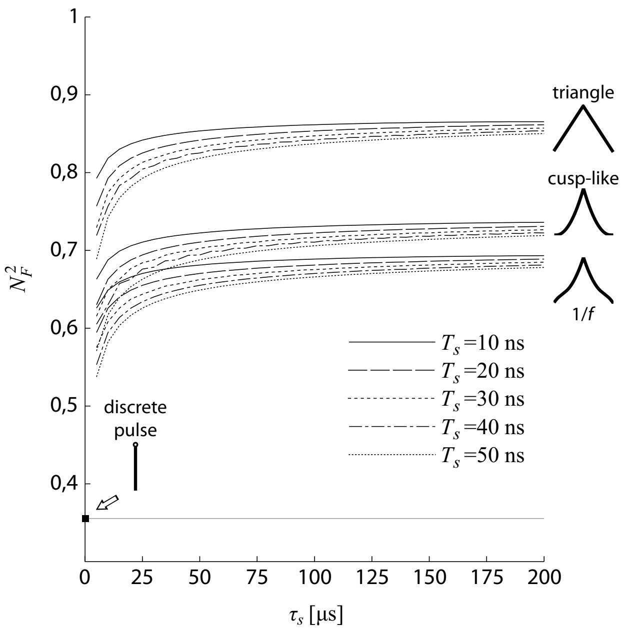


Figure 1: N_F^2 vs. τ_s and T_s for triangle, cusp-like, optimal for $1/f$ noise and step shapers.

2.2.2. Noise/signal ratio

In a real benchmark experiment, the spectral densities of each noise type are not available unless they are calculated. However, a pulse sample $S[n]$ and a noise sample $N[n]$, that is, the value at the output of the shaper when no events are produced, can be easily captured. Using this pair of samples, the noise/signal ratio R can be estimated by the following expression

$$R^2 = \frac{1}{S^2} \sum_{n=0}^{Z} N^2[n] \tag{14}$$

where S is the amplitude (maximum) of $S[n]$ at the output of the shaper and Z is the length of the noise signal captured in cycles.

3. Platform design for FPGA

The simulated annealing algorithm is executed in a microprocessor embedded in an FPGA. Fig. 2 illustrates the block diagram of the platform used in this experiment. The system, which was implemented upon the Xilinx Virtex-4 ML410 platform, has the following components: a PowerPC 405 processor embedded in a Virtex-4 XC4VFX60-11FF1152 that executes the simulated annealing algorithm; a memory system; a serial port for the communications with an external computer; and a custom Intellectual Property (IP) module to capture raw data coming from a propietary data acquisition board whose core is an ADC (a 14-bit Linear LTC2171). Both boards transmit and receive data through LVDS signals.

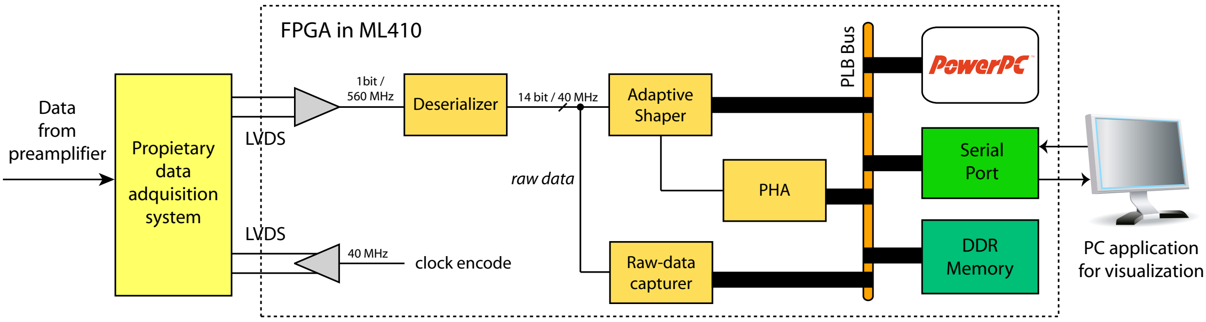


Figure 2: Block diagram of the system.

The ADC works at a dynamically selectable frequency of up to 40 MS/s. The embedded system works at 100 MHz and occupies 2,512 out of 25,280 (10%) slices in the FPGA included in the platform. The simulated annealing was implemented in C and the rest of the system was developed using VHDL and IP cores.

The entire system was designed using the Xilinx ISE Design Suite 10.1 and Xilinx Embedded Development Kit 10.1. It was debugged using ModelSim SE 6.3 and Chipscope 10.1 via JTAG. Finally, the software for this system, including the algorithm, was created and debugged using Xilinx Embedded Development Kit 10.1.

3.1. Hardware modules

3.1.1. Adaptive Digital Shaping Module

This module performs the shaping of the signal coming from the ADC. It works as a FIR filter whose output response is given by the convolution between the input pulse and x_n coefficients. This module implements a Processor Local Bus (PLB) interface to be controlled by the PowerPC. Thus, its coefficients can be adapted when the simulated annealing algorithm calculates the optimal shaper.

3.1.2. Pulse Height Analysis Module

This module is the last stage of the data processing. It receives data coming from the adaptive digital shaper as input. When this module signals a trigger event, it starts calculating the corresponding pulse height that is determined by the difference between top value and the estimated baseline of the signal, according to algorithm presented in [11].

To create the histogram, the data coming from this module was captured using the serial port and exported to Matlab.

3.1.3. Raw Data Capturer Module

By means of this module, the acquired data is stored in local memory. This module, which implements a PLB interface, works in two operating modes: (i) raw data — all acquired data is stored in local memory;

(ii) pulse data — only pulses are stored in local memory.

The first mode is used to capture noise $N[n]$ whereas the second mode is used to capture a pulse $S[n]$. These captures are used to calculate the noise/signal ratio presented in Section 2.2.2.

The length of the capture Z can be configured with the microprocessor as it is limited by the amount of memory controlled by the FPGA.

3.1.4. PowerPC 405 processor

PowerPC 405 is the microprocessor that Xilinx provides in their Virtex-4 FX models. In this implementation, it executes a simulated annealing algorithm that evolves a population of shapers, trying to minimize the cost function. The optimal shape is the one that produce the minimum value for this cost function.

3.2. Software components

There are three software components included in this platform: (i) the simulated annealing algorithm executed by the PowerPC embedded processor, (ii) the debug module to test that this algorithm works properly and (iii) the software module used to display the evolution of the annealing process in a PC using the serial port.

3.2.1. Simulated annealing algorithm

The algorithm proposed in Section 2.1 is programmed in this module. It is implemented in C to run in the embedded PowerPC 405 using floating-point numbers. The input parameters for this algorithm are the pulses and the noise captured with the Raw Data Capturer Module. The output of this software component is the coefficients to adapt the Adaptive Digital Shaping Module.

3.2.2. Debug module

The debug module is a workaround to establish numerical spectral densities of noise. In this case, the input parameters for the simulated annealing algorithm is the spectral densities for each supported type of noise (series, parallel and $1/f$ noise). When using this module, the cost function is the value of ENC according to Eq. (5).

3.2.3. Display module

This module sends data to a standard PC via the serial port. Data includes coefficients of the resulting optimal shaper, or the value of the cost function for each value of T.

4. Experiments

To validate the robustness of this algorithm, three groups of experiments were performed. The first attempts to reach a known target shaper applying this algorithm. The second one applies the algorithm to a known experiment to compare results. Finally, the third one validates the entire design using real data.

Results of the first and second experiments show that the simulated annealing algorithm works properly. Results of the third experiment prove that the algorithm works even in a real environment and that a real embedded design based on simulated annealing is perfectly feasible.

In every experiments, the platform presented in Section 3 was used to test the system. The cost function varies as explained in Section 2.2.

4.1. Simulated annealing using specific noise types

The aim of the first experiment is to obtain the optimal shaper for series noise and $1/f$ noise according to [11] to check that the algorithm works properly. Besides, according to [14], when the effect of series and parallel noise are equal in a spectroscopy system, the optimal shaper is a cusp-like shaper.

Fig. 3 shows the result of the application of the algorithm. The first column shows the resulting shaper for each temperature step (dashed lines imply higher temperatures). The second column depicts the final shaper. The third column represents the evolution of the function cost (in this case, Eq. (5)). As a result of this test, it can be observed the optimal shapers for each type of noise: (a) parallel noise, (b) series noise, (c) equal influence of series and parallel noise (cusp-like shaping) and (d) shaper for $1/f$ noise [11].

The test was carried out using 25 temperature steps and different population of individuals P. This value has influence on the shape, depending on the type of shape. Thus, in the (a) case, only $P = 10$ is required to archive the optimal shaper for parallel noise. However, in the (c) case, $P = 100$ is not enough to obtain a defined cusp-like shape. In the rest of the cases, the shapes achieved with $P = 100$ and $P = 500$ are similar.

The execution time of the algorithm using the FPGA platform was 11 ± 1 seconds in the case of $P = 100$ and 46 ± 1 seconds in the case of $P = 500$.

4.2. Simulated annealing using a known experiment

In this example, which was obtained from [1], an equivalent input capacitance of $C_i = 0.3$ pF is used. An additional constraint imposed to the shaper is that it has to guarantee a shaping time $\tau_s = 4$ μs. If a filter with $N = 100$ is used, according to Eq. (1), $T_s = 40$ ns.

In this example the following noise spectral densities are assumed:

- $v_n^2 = 1.5 \cdot 10^{-18}$ V^2/Hz
- $i_n^2 = 1.6 \cdot 10^{-31}$ A^2/Hz
- $v_{fn}^2/\,|f| = 1.5 \cdot 10^{-12}/\,|f|$ V^2/Hz

The result of the application of the algorithm is shown in Fig. 4. As in Fig. 3, the first column, the shaper for each temperature step is shown (dashed lines implie higher temperatures). The second column

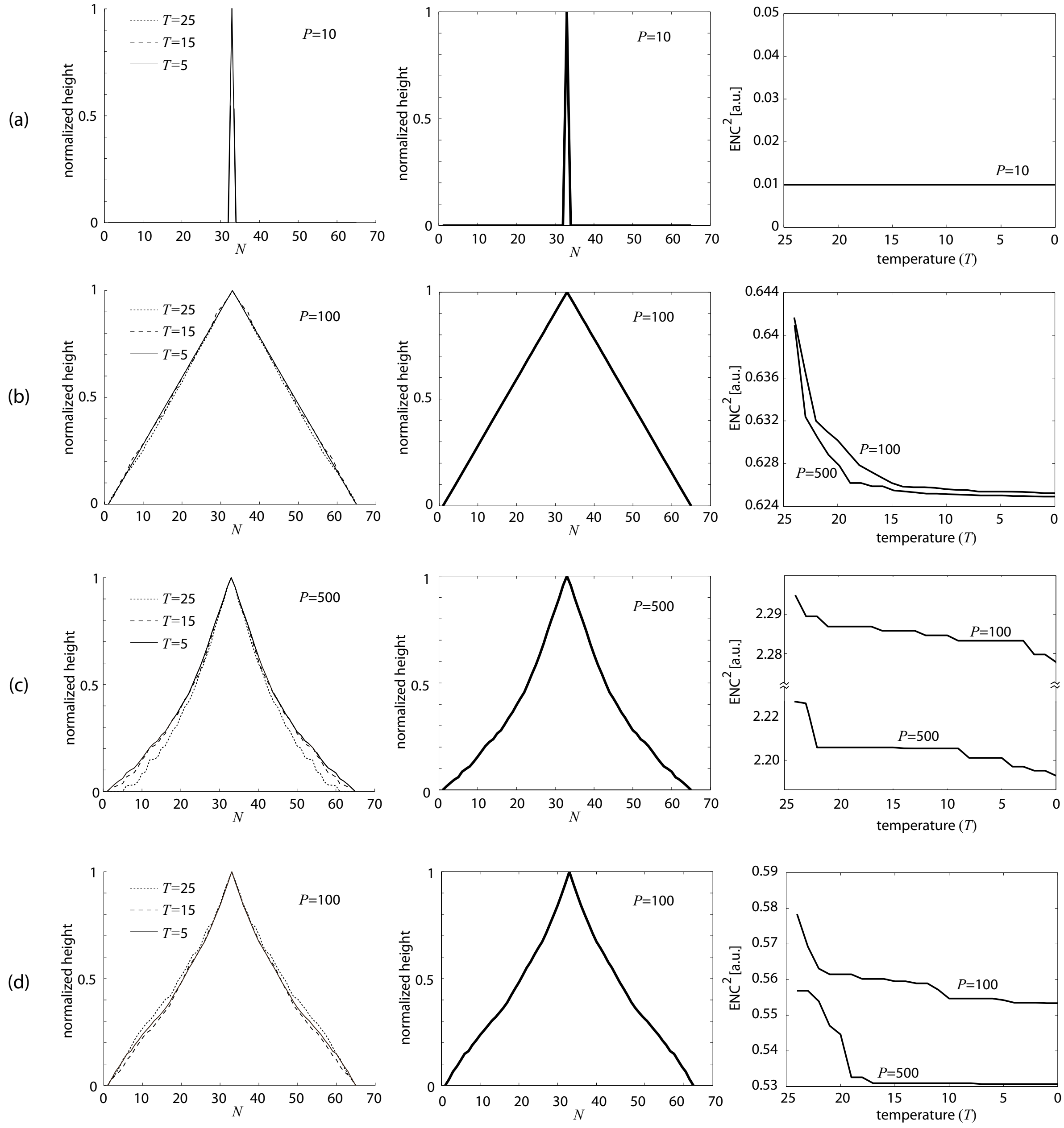


Figure 3: Algorithm results for (a) $v_n = v_{fn} = 0, i_n > 0$. (b) $i_n = v_{fn} = 0, v_n > 0$. (c) $v_{fn} = 0, C_i v_n = i_n$. (d) $i_n = v_n = 0, v_{fn} > 0$.

depicts the final shaper. The third column represents the evolution of the cost function (in this case, Eq. (5) to obtain the ENC).

According to Eq. (5), using the resulting shaper, an ENC equal to 4.106 e^- with $P = 100$ is obtained. In case of $P = 500$, an ENC equal to 4.407 e^- is obtained. It implies an improvement of 6.8 % increasing the population (and the execution time) fivefold.

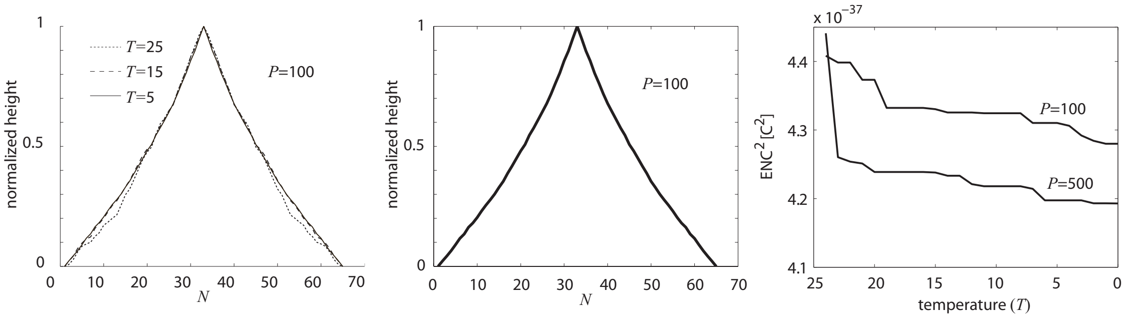


Figure 4: Algorithm results.

5. Experimental results using a neutron monitor

Lastly, a test to check the proposed annealing algorithm in a real environment was performed. The main objective of this test is to obtain similar results to those obtained in experiments carried out without using the annealing algorithm to adjust the shaper.

This test was performed in the Castilla-La Mancha Neutron Monitor (CaLMa) located in Guadalajara, Spain. The instrument consisted of fifteen proportional gas counter tubes. More information about features, setup and results of this instrument can be found in [15]. In both, the cited experiment and the present test, a gas tube (LND2061) connected to a Canberra ACHNA98 amplifier followed by an AmpTek 8000A Multichannel Analyzer were used. A complete setup of the experiment is shown in Figs. 5 and 6. The Data Acquisition Board (DAQ) was a proprietary design, whereas the Digital Processing Board was an evaluation board (Xilinx ML410).

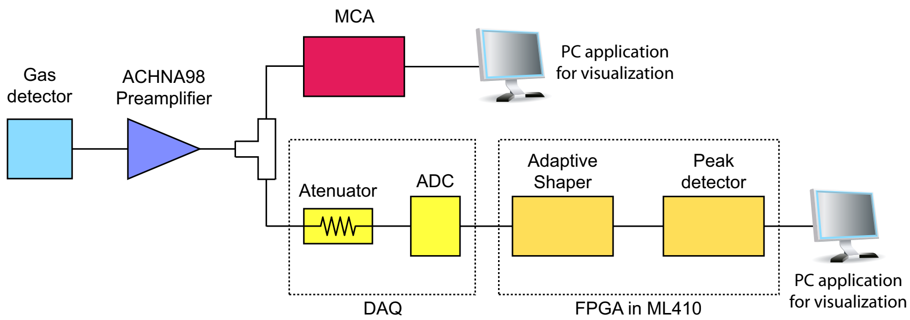


Figure 5: Setup of the neutron monitor experiment.

In real particle detectors, the pulses can have a different duration and amplitude. This is the case of the detector in use at this facility. When this occurs, the user must define a region of interest based on previous experiences to apply this algorithm. Fig. 7 shows a pulse whose height is within the region of interest.

To perform the test, the preamplifier was connected to a proprietary data acquisition system described in section 3. Following this board, a ML410 board, whose core was the digital system explained in Section

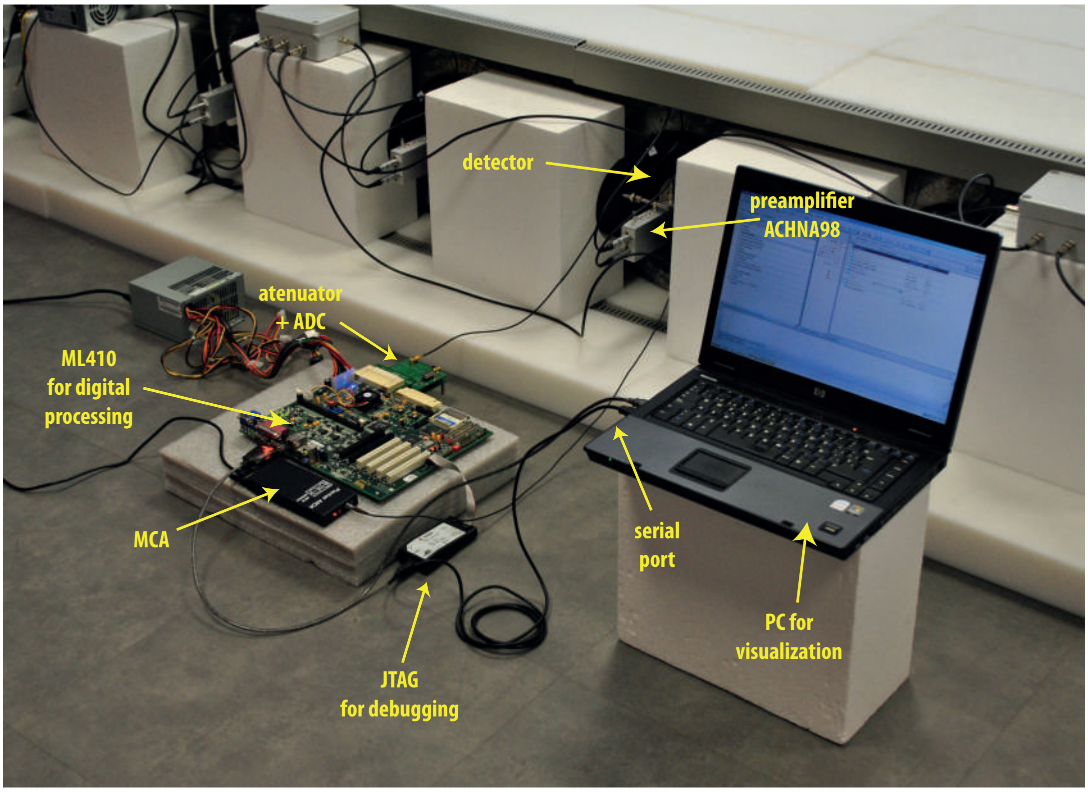


Figure 6: Image of the setup of the neutron monitor for this experiment.

3, included the described adaptive shaper and the peak detector. As said in Section 3, both boards transmit and receive data through LVDS signals. We selected the beginning of the region of interest (channel #265) as the threshold for the shaper.

The pulses coming from the preamplifier have semi-Gaussian shape with a duration of approximately 4 μs. For this reason, the data acquisition board operates at a sampling frequency $f_s = 1/T_s = 10$ MS/s to avoid sampling noise [16]. According to the presented algorithm, the value of $N = 65$ is fixed. However, this value can be automatically changed by the simulated annealing algorithm.

Fig. 8 shows a sample of noise captured using the propietary data acquisition board. For this test, $Z = 8192$.

The results of the experiment are shown in Fig. 9. In the first column, the shaper for each temperature step is shown (dashed lines imply higher temperatures). In the second column the optimal shaper according to this algorithm is depicted. The third column represents the evolution of the function cost, in this case, Eq. (14).

The predominant noise is series and $1/f$ noise. In fact the obtained shaper is similar to the one obtained in section 4.2 for values of in negligible compared to v_n and v_{fn}. Therefore, the shaper is a quasi-triangle shaper.

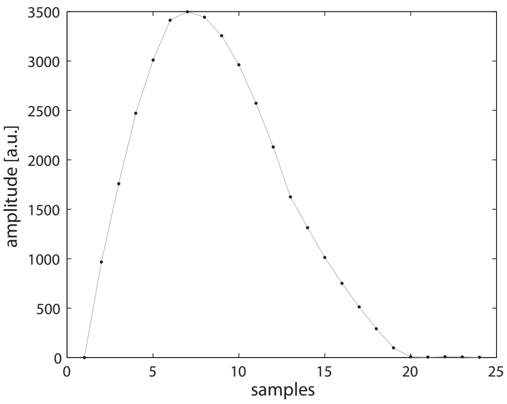


Figure 7: Pulse event ($S[n]$) coming from the preamplifier. The signal was sampled at 10 MHz.

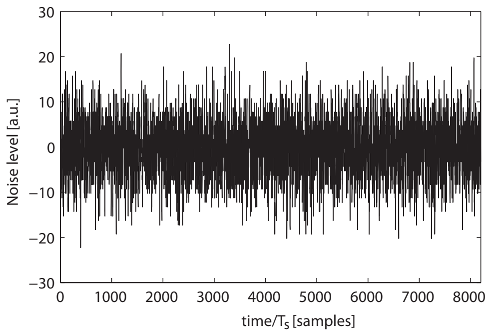


Figure 8: Example of noise captured $N[n]$ with the data acquisition board with no radiation events.

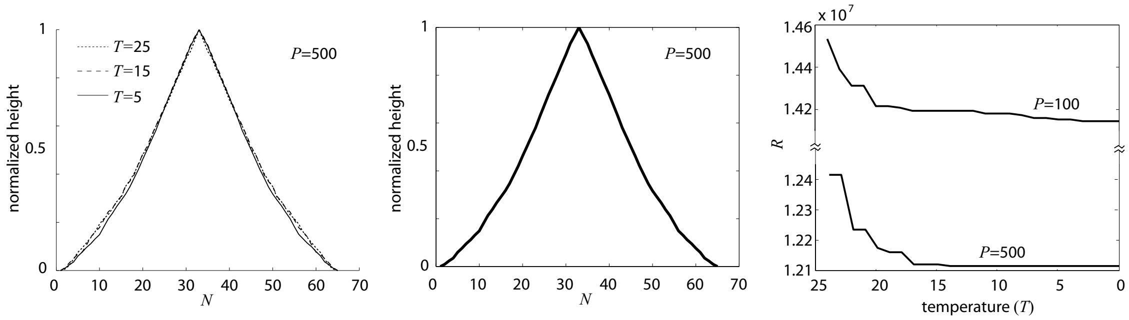
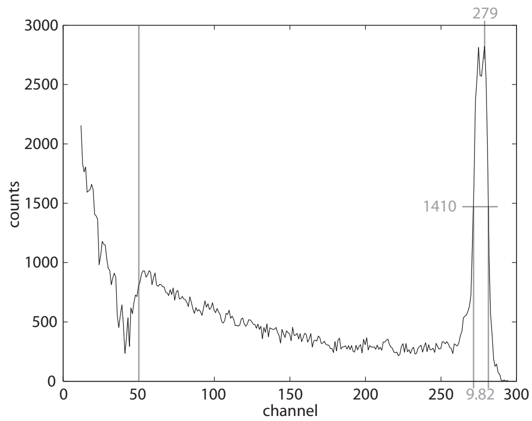


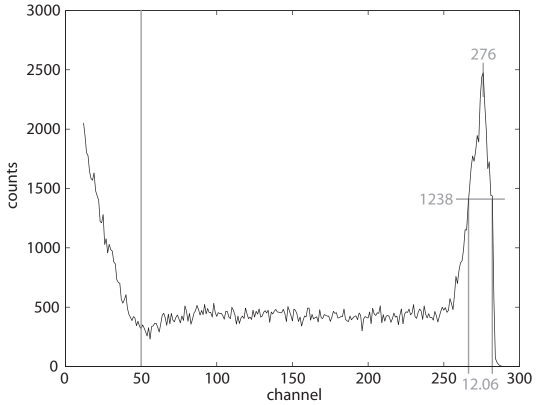
Figure 9: Results of the algorithm for the neutron monitor.

Once obtained the optimal shaper, two histograms are obtained. The first one has been obtained using a shaper adjusted with the coefficients of Fig. 9. The second one has been obtained using the Multichannel Analyzer. Both histogram were generated simultaneously using the setup depicted in Fig. 5.

The obtained histogram is shown in Fig. 10(b) whereas the obtained histogram with the Multichannel Analyzer is shown in Fig. 10(a). The first one was created using 192160 samples and the second one was created using 194190 samples. The measured energy spectrum was 2.31 MeV (channel #300). The duration



(a)



(b)

Figure 10: Histogram for pulses obtained using (a) the AmpTek 8000A Multichannel Analyzer and (b) the shape of Fig. 9 obtained from the annealing algorithm. The grey lines indicate the threshold level.

of the test was 10 hours. Thus, the average event rate considered was 5.3 per second.

In the case of using the shaper adjusted by means of the simulated annealing algorithm, a FWHM equal to 12.06 (4.36%) was obtained, whereas in the case of the histogram created with the MCA, a FWHM equal to 9.82 (3.58%) was obtained. It is important to consider that the MCA does not apply any shaping and the height of each pulse is captured by means of an analog circuit, whereas in this test, the shaping of Fig. 9 was applied and the height of each pulse is captured using a digital circuit working at 10 MS/s. However, a comparison of both figures indicates the similarity of the FWHM of the histograms from both experiments. It may be reasonable to assume that the differences in the histograms were the result of using different detection chains to generate them (see Fig. 5) and ADC sampling effects. In fact, the Multichannel Analyzer used does not apply any digital shaping and the pulse height capture is performed using analog electronics. However, the inclusion of digital signal processing implies an increase of the algorithm complexity and reusability that improves the efficiency of nowadays spectroscopy systems.

Once the coefficients were adjusted, the architecture of the shaper is linear and time-invariant because, as stated in Section 2.1, it works as a FIR filter. Thus, the maximum event rate of this shaper depends on the shaping time and on the pile-up management selected in the same way than other non-adaptive, linear, time-invariant shapers.

6. Conclusions and future work

In this study, a novel algorithm which uses simulated annealing for calculating optimal filters in presence of arbitrary noise type was designed and implemented in an FPGA. To test the efficiency of this algorithm, theoretical examples were evaluated and one real setup was measured in real radiation facilities. Additional constraints such as shaping time or peak time can be added modifying the parameters of this algorithm. It can be concluded that this algorithm is a promising method to be taking into account in successive digital spectroscopy systems due to its efficiency, simplicity and implementability in programmable logic or embedded processors.

As future work, it is planned to compare the results of the algorithm applying different temperature gradients.

Acknowledgments

This project was funded by the Spanish Administration as part of project ref. AYA2012-39810-C02-02. The authors thank the Guadalab and the CaLMa Team for providing the CaLMa facility supported by *Junta de Comunidades de Castilla-La Mancha* through the project PPII10-0150-6529.

Bibliography

[1] E. Gatti, A. Geraci, G. Ripamonti, "Automatic synthesis of optimum filters with arbitrary constraints and noises: a new method", *Nuclear Instruments and Methods in Physics Research A* vol. 381, (1996) 117–127.

[2] E. Gatti, A. Geraci, G. Ripamonti, "Optimum filter for $1/f$ current noise smoothed-to-white at low frequency (Letter to the Editor)", *Nuclear Instruments and Methods in Physics Research A* vol. 394, (1997) 268–270.

[3] F. S. Goulding, "Pulse-Shaping in Low-Noise Nuclear Amplifiers: A Physical Approach to Noise Analysis", *Nuclear Instruments and Methods* 100, (1972) 493–504.

[4] N. Menaaa, P. D'Agostino, B. Zakrzewski, V. T. Jordanov. "Evaluation of real-time digital pulse shapers with various HPGe and silicon radiation detectors". *Nuclear Instruments and Methods in Physics Research A* vol. 652, (2011) 512–515.

[5] R. A. Rutenbar, "Simulated annealing algorithm: an overview", *IEEE Circuits and Device Magazine* 5, (1989) 19–26.

[6] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, vol. 220, (1983) 671–680.

[7] J. Beringer, et al. (Particle Data Group), *Physical Review D* vol. 86, 010001, (2012) pp. 357–359.

[8] A. Pullia, "Impact of non-white noises in pulse amplitude measurements: a time-domain approach", *Nuclear Instruments and Methods in Physics Research A* vol. 405, (1998) 121–125.

[9] G. Ripamonti, A. Castoldi, E. Gatti, "Multiple delay line shaping: A new class of weighting functions suitable for digital signal processing", *Nuclear Instruments and Methods in Physics Research A* vol. 340, (1994) 584–585.

[10] J. H. Fischer, "Noise Sources and Calculation Techniques for Switched Capacitor Filters", *IEEE Journal of Solid-State Circuits* SC-17, no. 4, (1982) 742–752.

[11] V. T. Jordanov, "Real time digital pulse shaper with variable weighting function", *Nuclear Instruments and Methods in Physics Research A* vol. 505, (2003) 347–351.

[12] L. Bardelli, G. Poggi, "Digital-sampling system in high-resolution and wide dynamic-range energy measurements: Comparison with peak sensing ADCs", *Nuclear Instruments and Methods in Physics Research A* vol. 560, (2006) 517–523.

[13] E. Fairstein, "Linear Unipolar Pulse-Shaping Networks: Current Technology". *IEEE Transactions on Nuclear Science*, vol. 37, no. 2, (1990) 382–397.

[14] P. W. Nicholson, *Nuclear Electronics*. John Wiley & Sons, Ltd., 1973.

[15] J.Medina, et al., "Castilla-La Mancha neutron monitor". *Nuclear Instruments and Methods in Physics Research A* vol. 727, (2013) 97-103.

[16] R. Abbiati, A. Geraci, G. Ripamonti, "Analog Shaping Optimization for Digital Processing of Radiation Detector Signals". *IEEE Transactions on Nuclear Science*, vol. 52, (2005) 1638–1642.