Nuclear Instruments and Methods in Physics Research A 735 (2014) 297–303

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

# Implementation of a real-time adaptive digital shaping for nuclear spectroscopy

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

ARTICLE INFO

*Article history:*
Received 4 January 2013
Received in revised form
21 August 2013
Accepted 10 September 2013
Available online 4 October 2013

*Keywords:*
Spectroscopy
Shaping
Adaptive
Signal processing
FPGA
Nuclear detectors

ABSTRACT

This paper presents the structure, design and implementation of a new adaptive digital shaper for processing the pulses generated in nuclear particle detectors. The proposed adaptive algorithm has the capacity to automatically adjust the coefficients for shaping an input signal with a desired profile in real-time. Typical shapers such as triangular, trapezoidal or cusp-like ones can be generated, but more exotic unipolar shaping could also be performed. A practical prototype was designed, implemented and tested in a Field Programmable Gate Array (FPGA). Particular attention was paid to the amount of internal FPGA resources required and to the sampling rate, making the design as simple as possible in order to minimize power consumption. Lastly, its performance and capabilities were measured using simulations and a real benchmark.



## 1. Introduction

Nuclear spectroscopy is the term used to describe the electronic systems employed to study elemental particles and nuclear physics. These techniques have become extensively used in many fields, such as radiation detection or space particle telescopes. Typical elements of these devices include high-speed detectors to measure rates and energy, preamplifiers, shapers, discriminators, counters and pulse height analyzers [1]. From the point of view of signal transformation, the weak and fast pulse provided by the detectors is amplified and filtered to optimize detection. This process has traditionally been implemented by means of analog electronic elements such as operational amplifiers, comparators and peak detectors [2].

The basic detection chain for a particle detector using analog electronics is shown in Fig. 1(a). The pulse provided by the detector is integrated into the preamplifier to obtain a fast slope signal followed by a long tail. This signal is typically filtered to obtain a quasi-Gaussian shaped output, a procedure usually known as pulse shaping, the purpose of which is to increase the Signal-to-Noise Ratio (SNR) in order to optimize detection [3]. In several detectors, the pulse charge is proportional to the energy of the detected particle. Also, the charge of the pulse is proportional to its amplitude. In these situations, a peak detector may be used to obtain a value proportional to the charge.

Nowadays, digital signal processing is also used in particle detectors [4,5]. The proposed detection chain for a particle detector using digital signal processing is shown in Fig. 1(b). When the same detector and preamplifier from Fig. 1(a) are used, an Analog-to-Digital Converter (ADC) should be added after these stages to digitalize the signal. In addition, an optional analog filtering stage before the ADC could also be added. One objective of this study was to use this algorithm in a future satellite payload, an environment in which power requirements are very restrictive. Thus, one main objective was to reduce sampling frequency as much as possible in order to reduce power consumption. With this goal in mind, a passive low-pass filter was included between the preamplifier and the ADC. This was an anti-aliasing filter, which is typically used before an ADC, and its inclusion made it possible to fulfill the Nyquist criteria. The pole generates a low-pass filter that avoids the aliasing effect and thus, the sampling frequency could be decreased [6]. After the ADC, all processing, including shaping and peak detection, is performed using digital elements.

The ideal shaping for a detector depends on the input signal and the predominant noise [7]. Thus, specific techniques are used to synthesize various pulse shapers and optimize detection by maximizing the SNR [8,9]. However, the noise spectrum can vary

* Corresponding author at: Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain. Tel.: +34 669013583.
*E-mail addresses:* aregadio@srg.aut.uah.es, alberto.regadio@edu.uah.es, aregadio@isdefe.es (A. Regadío), ssanchez@srg.aut.uah.es (S. Sánchez-Prieto), mprieto@srg.aut.uah.es (M. Prieto), taberogj@inta.es (J. Tabero).

0168-9002/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.nima.2013.09.063

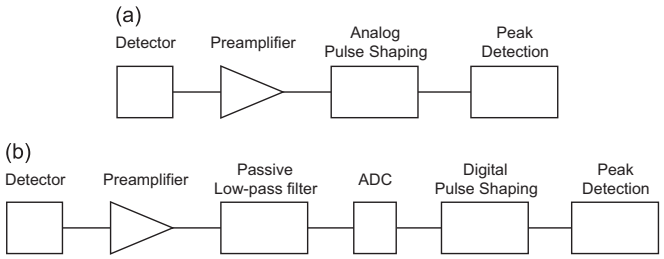


**Fig. 1.** (a) Typical particle detection chain based on analog electronics. (b) Particle detector chain based on mixed analog-and-digital electronics.

over time for several reasons (e.g. accumulated radiation on the detector and damage to the detector structure [1]), whereas these techniques do not usually consider noise variations in real-time [4,5]. To avoid this drawback and allow the use of self-adjusting transfer functions, the use of adaptive shapers was proposed.

Several algorithms exist for performing adaptive shapers (e.g. Least-Mean-Square (LMS), Digital Penalized LMS (DPLMS), Wiener algorithm and Discrete Fourier Transform (DFT) method [10,11]). The LMS algorithm finds the filter coefficients associated with producing the least mean squares of the difference between the desired and the actual signal. The DPLMS is an improvement on the LMS algorithm proposed in Ref. [10]. The Wiener algorithm minimizes the error by equating each partial derivative to zero. Finally, the DFT method is based on dividing the desired signal by a given input signal. The shape can also be adapted by modifying any of its parameters (e.g. the algorithm presented in Ref. [12] searches for the optimum shaping time).

In this paper, we propose a technique for automatic synthesis of digital pulse shapers for high-resolution spectroscopy using an adaptive algorithm based on the DFT and LMS methods. The proposed algorithm is suitable for real-time implementation and only requires simple hardware and an example pattern for training. The main advantage of this technique is that it has the capacity to transform any input signal into any output signal with the desired features (including typical shapes such as trapezoidal, triangular or cusp-like outputs).

In Ref. [10], algorithms for adaptive digital shaping with a good performance were presented; however, implementation of most of these algorithms is quite complex. This complexity would also affect power consumption in a real implementation, which should be kept as low as possible. In addition, in the majority of these algorithms, the entire input signal must be known before calculating the filter coefficients, whereas the algorithm proposed here has the capacity to start adjusting the shaper coefficients as soon as the first pulse sample arrives. To sum up, the proposed design is a synthesizable algorithm that takes into account area, power consumption and above all, functionality.

## 2. Proposed adaptive algorithm

As explained in the Introduction, the utilization of different shaping techniques makes it possible to improve resolution in particle detectors [13]. Consequently, the aim of this study was threefold: to develop an automatic desired pulse shaping operation, to reduce input noise and to solve the pile-up effect. Of the design possibilities for obtaining the desired results, adaptive algorithms were selected because they have the capacity to automatically adjust the coefficients of a linear system and because they are easy to implement in digital hardware.

The proposed shaper works as an adaptive digital Finite Impulse Response (FIR) filter. The output of a FIR filter is given by the following expression:

$$y[n] = \sum_{k=0}^{N-1} a_k x[n-k] = h[n] * x[n] \tag{1}$$

where $x[n]$ is the input signal, $y[n]$ is the output signal, $a_k$ are the coefficients of the filter, $h[n]$ its impulse response and $N$ its order.

Two parameters related to digital filters are the shaping time $\tau_s$, which is defined as the duration of the shaper output signal $y[n]$, and the ADC sampling period $T_s$. These are related to $N$ as follows:

$$N = \frac{\tau_s}{T_s} \tag{2}$$

One method to adjust the $a_k$ coefficients, which is described in Ref. [10], uses an iteration method to search for the solution that minimizes the error according to this formula:

$$h(z) = \frac{x(z)}{d(z)} \tag{3}$$

where $x(z)$ and $h(z)$ are the $z$-transform of $x[n]$ and $h[n]$, respectively, and $d(z)$ is the $z$-transform of the desired output signal $d[z]$. This is one of the simplest methods for adapting a shaper. Thus, implementing it in an FPGA yields a low occupation area and low power consumption, one of the objectives of this study. However, several drawbacks of this method are described in Ref. [10], including output signal undershoots and the possibility that the filter order is infinite (i.e. it does not belong to the FIR class). To minimize the effects of these drawbacks and obtain better results, the lengths of $x[n]$ and $d[n]$ should be equal or the length of $d[n]$ should be slightly smaller than the length of $x[n]$. This is because when $x[n]$ is equal to zero during several cycles at the end of the pulse, the shaper needs $a_k$ coefficients with high values to obtain $d[n]$ at the output. Therefore, the coefficients at the end are much greater than those at the beginning, producing an increase in the undershoot and a poor adaptation depending on $x[n]$.

One method to mitigate noise and phase shift is to replace $x[n]$ by the mean value of $\Lambda$ input signals; however, this does not work properly unless the input pulses have the same shape and duration. Furthermore, the sampling frequency should be sufficiently high to ensure a negligible phase shift between pulses. Thus, there are many applications in spectroscopy where this procedure cannot be applied, and $\Lambda$ shall be equal to one.

## 3. Design

As indicated in the Introduction, the proposed design was conceived with two goals in mind: simplicity and low power consumption. Therefore, it was important to reduce the functional units and working frequency as much as possible. Furthermore, both the shaper and peak detector were implemented in the same FPGA.

In this design, the output pulse $y[n]$ is converted into a desired pulse $d[n]$ by adjusting the shaper coefficients in real-time as a function of an input signal $x[n]$.

Both $x[n]$ and $d[n]$ are introduced into the shaper at the same time. According to Eq. (2), both $N$ and $T_s$ must be set manually before the shaper adjusts its coefficients. The correct $T_s$ depends on the duration of the input pulses.

In order to synchronize $x[n]$ and $d[n]$, this last signal is provided when $x[n]$ is greater than a predefined learning threshold, which should be equal to or greater than the pulse-height analysis threshold. Thus, the shaper only "learns" when a desired input pulse arrives, avoiding possible false triggers due to noise and undesired pulses.

Fig. 2 shows a detailed diagram of the shaper. The *den* signal is used to enable the learning process. The error $e[n]$ is obtained by subtracting the reference signal $d[n]$ from the output signal $y[n]$

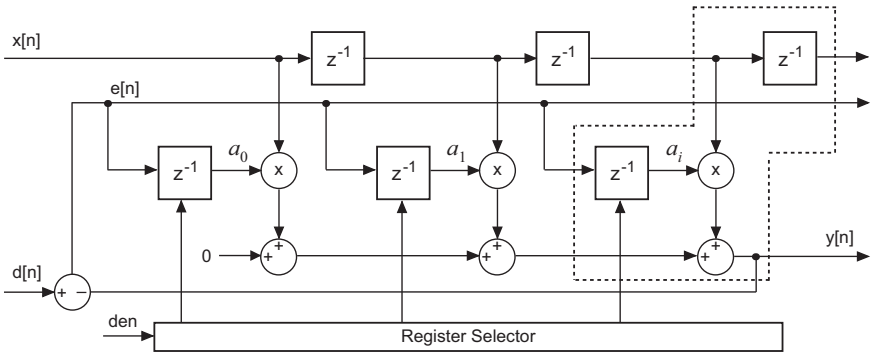


**Fig. 2.** Block diagram of the third order adaptive shaper using one learning cycle.

and is used to adjust the coefficients signaled by the Register Selector which enables one register each clock cycle while the *den* signal remains at high-level. The structure inside the box must be repeated $N$ times, where $N$ is the order of the shaper. For each sample from the input signal, a coefficient $a_i$ is adjusted. The process ends when all the filter coefficients have been adjusted. In this moment, the *den* is disabled and the overall shaper works as a filter. If so desired, the learning process can repeated by once again enabling the *den* signal.

## 4. Implementation

The overall design was implemented in VHDL. The order of the shaper $N$ and the input/output signal width $W$ are configurable parameters. The structure proposed in Fig. 2 can be translated into VHDL code for implementation in the FPGA using the following pseudocode:

```
a(1..N) ⇐ 0;
for i=1 to N –learning process
   x(1) ⇐ x;
   for j=2 to N –shift register
      x(j) ⇐ x(j−1);
   end for;
   if i=1 then
      a(1) ⇐ d(1)/FP;
   else
      y ⇐ Σ_{j=1}^{N} x(j)·a(j);
      e ⇐ d(i)−y(i);
      a(i) ⇐ e/FP;
   end if;
end for;
```

where *FP* is the fixed-point parameter used to adjust the scale of the values used. The lower the value of this parameter, the better the adaptive shaper fits to the desired signal; however, a low value of *FP* could cause an overflow. Therefore, the *FP* should be adjusted specifically for each detection chain. It is strongly recommended that this value is a power of two to simplify the division operation.

In accordance with Fig. 2 and the pseudocode presented, for each order of the shaper, an adder, a multiplier and two flip-flops, all of them with width $W$ are synthesized. In this way, the resources used for a given device can be estimated. As explained in Section 3, both shaper and detector are synthesized in the same digital chip.

## 5. Test results

Two types of test were performed: simulation tests and real experimental tests using a NIM module. In both cases, it was assumed that when a particle is detected, the preamplifier stage will generate the following pulse:

$$x(t) = A\exp\left(\frac{-t}{\tau_1}\right) \tag{4}$$

where $A$ is the pulse amplitude and $\tau_1$ the adjustable constant decay. As stated in Section 1, in traditional spectroscopy systems, an analog shaper that produces a semi-Gaussian pulse usually follows the detector–preamplifier configuration. As described in the Introduction, in order to reduce $T_s$, the preamplifier is followed by a low-pass RC filter as an analog shaper. Thus, when it is fed with the pulses (4), the CR–RC output becomes

$$x(t) = A\frac{\tau_1}{\tau_1-\tau_2}\left(\exp\left(\frac{-t}{\tau_1}\right)-\exp\left(\frac{-t}{\tau_2}\right)\right) \tag{5}$$

Pulses similar to the one described in Eq. (5) were processed with the adaptive shaping presented in this paper. Finally, at the end of the chain, a peak-detector captured the maximum values of the signals generated by the digital shaper, and these values were used to measure chain detection performance.

### 5.1. Simulation tests

These tests consisted of simulating the adaptive shaper in a digital design tool, using the Mentor Graphics ModelSim SE 6.2c software package using CR and CR–RC as input signals. Although CR–RC shaping is usually carried out using $\tau_1=\tau_2$ [1], we selected $\tau_1=100$ μs and $\tau_2=1$ μs as the time constants in order to obtain signals with a similar shape (but with different shaping time $\tau_s$) to the signals coming from the facility described in Section 6. The input/output signals were processed with 14-bit resolution at 2.5 MHz because this is the resolution of the ADC used in the experimental test.

Although the proposed shaper has the capacity to synthesize a wide range of output signals, in this study the simulation tests were performed using trapezoidal and cusp-like shaping because these shapers are used extensively [5,9]. In addition, triangular shaping was also used in this test because this is the optimum type for white noise [7]. To simplify the tests, the rising and falling edge of the trapezoidal shaper was $N/4$ cycles long and the flat-top was $N/2$ cycles long in all tests (simulation and experimental) for a given order filter $N$. In triangular and cusp-like shaping, the edges were also symmetrical.

The duration of the CR input pulse (Fig. 3(a) and (b)) is just over 12 μs. This time is similar to the duration of the signals used in this section. According to Eq. (2), if this example was carried out using shapers with $N=32$, the sampling frequency $f_s=1/T_s$ should be approximately 2.5 MHz. Note that when the duration of a pulse is low with respect to the duration of the pulse used to adjust the coefficients (fast pulses), the pulse will not be captured correctly unless the sampling frequency changes and the shaper adjusts its

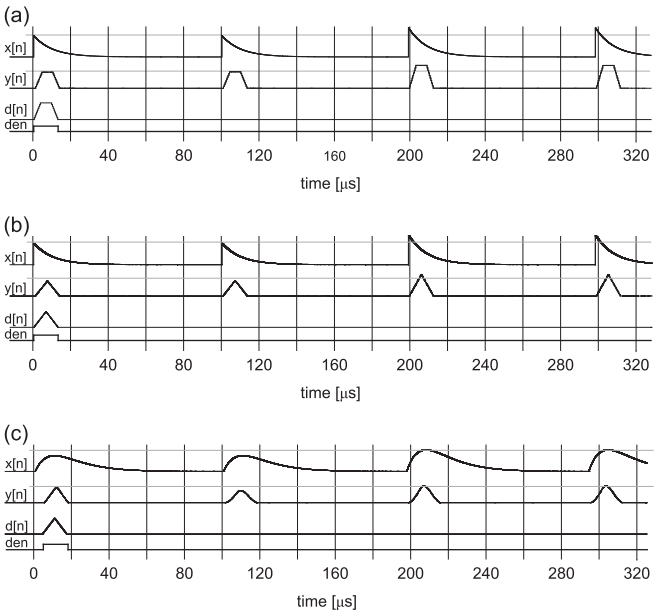


**Fig. 3.** Simulation of the learning process (first pulse) and shaping for 32-order trapezoidal (a), triangular (b) and (c) shapers.

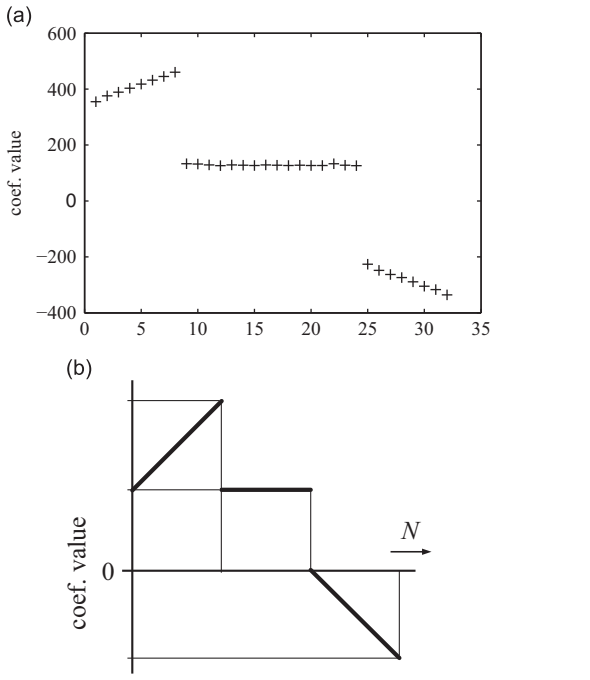


**Fig. 4.** Coefficients of an automatically synthesized $a_k$ trapezoidal shaper ($N=32$) (a) vs. a trapezoidal shaper impulse response (adapted from Ref. [14]) (b). In both cases, the input signal is a CR pulse.

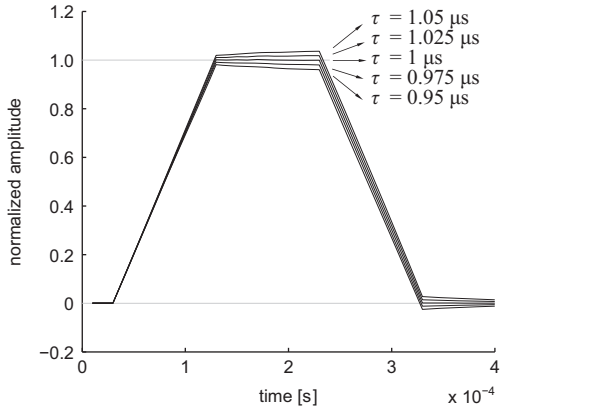


**Fig. 5.** Effect on shaping when the input signal varies in time duration. This shaper is adjusted to obtain a trapezoidal shape when the input signal is a CR pulse with $\tau_1 = 1$ μs.

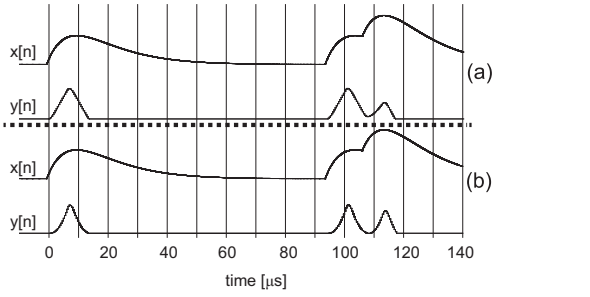


**Fig. 6.** Simulation of the pile-up effect for a triangular (top), and cusp-like (bottom) using trained 32-order shapers.

coefficients again. Although this seems a low clock frequency for an actual digital system, on the one hand the ultimate objective was to use this algorithm in a satellite payload where power requirements are very restrictive, and thus the sampling frequency should be reduced as much as possible, and on the other hand, according to Ref. [6], if a Gaussian signal is sampled at 2.5 MHz, the energy variation estimation is negligible compared to another noise sources.

Fig. 3 shows the output of the proposed 32-order shaper working at a $f_s =$ 2.5 MHz for a CR input (Fig. 3(a) and (b)) and a CR–RC input (Fig. 3(c)). In all the figures, the first and second $x[n]$ pulses are smaller than the third and fourth to show the effect of scaling on $y[n]$. The signals $d[n]$ and *den* are the desired signal and the learning enable, respectively. Thus, the learning process is active when *den* is active-high. It can be observed in Fig. 3(c) that the pulses are worse shaped than those shown in Fig. 3(a) and (b). This is because when $x[n]$ has a high maximum, $a_k$ coefficients are also adjusted with high values, minimizing the effect of rounding to the nearest integer. Minimizing this effect implies a greater similarity between $y[n]$ and $d[n]$.

After each learning iteration using the input/output signal in Fig. 3(a), the coefficients were automatically adjusted as shown in Fig. 4. As can be seen, the coefficients of the shaper were very similar to the static coefficients of the trapezoidal shaper presented in Ref. [14], leading us to confirm that the shaper had been synthesized correctly.

The effect on shaping when the input signal varies in duration is similar to the effect produced using non-adaptive shapers since equal shapers produce equal results independently of the method applied to calculate their coefficients. Fig. 5 shows the effect on an adaptive trapezoidal shaping when the time constant $\tau_1$ of the CR pulse varied. The shaper used was the same as the one whose coefficients are depicted in Fig. 4(a). It produced a trapezoid when the input signal was a CR pulse of $\tau_1 = 1$ μs. As can be seen, these variations affected pulse height and consequently, resolution. Therefore, time variations should be minimized.

Independent of the coefficients obtained, the shaper is linear and time-invariant; thus, it can facilitate the detection of pile-up events. Fig. 6 shows a pile-up event, in which the first signal has twice the amplitude of the second one. For pile-up management, a pile-up event processing stage should be included between the digital shaper and the peak detector [15].

### 5.2. Experimental results using a reference pulser

After the shaper had been successfully simulated, it was tested using a real benchmark. In this scenario, the pulses explained at the beginning of this section were generated using a Canberra Model 1407 Reference Pulser of a Nuclear Instrumentation Module (NIM). This NIM generates a group of periodic pulses of equal amplitude that can be fed into the test input of the preamplifiers or the shapers. It generates an exponential pulse identical to those produced by the detector–preamplifier, which can be fed in at the ADC stage (see Fig. 1(b)).

The reference pulse generator was adjusted to obtain a time constant $\tau_1 = 100\,\mu s$ and $\tau_2 = 1\,\mu s$. No more analog electronics were required after the analog shaper, except for the ADC (see Fig. 1), a Texas Instruments 14-bit (ADS5474) with up to 25 MS/s. With these values, Eq. (2) can be used to obtain the sampling frequency for a given shaper order $N$ (Table 1). Following the ADC, an FPGA was used to perform the rest of the processing (pulse shaping and pulse-height analysis). Besides, to adjust $f_s$, the FPGA drives the ADC input clock. FPGA logic was implemented in a Xilinx Virtex-4 FPGA (XC4VFX60-11FF1152). Finally, a Xilinx Chip-Scope Pro 10.1 was employed to capture the pulse amplitudes using the JTAG port.

**Table 1**
Shaper order $N$ and the corresponding sampling frequencies $f_s$ used in the experimental tests. $M$ is the flat-top number of samples of the trapezium and $L = N - M/2$ its number of edge samples.

| $N$ | $L$ | $M$ | $f_s$ (kHz) |
|---|---|---|---|
| 16 | 4 | 8 | 625 |
| 32 | 8 | 16 | 1250 |
| 40 | 10 | 20 | 1562.5 |
| 56 | 14 | 28 | 2187.5 |
| 64 | 16 | 32 | 2500 |
| 80 | 20 | 40 | 3125 |

Once the input signal and the shaping type have been selected, the shaper learns with the method explained in Section 3. To test how the shaper worked, a trapezium was synthesized. The results for each of the shapers shown in Table 1 are given in Fig. 7, where it can be seen that the trapezoidal shape was obtained from $N = 32$.

One key parameter to measure the quality in spectroscopy of the energy resolution is the Full Width at Half Maximum (FWHM). Thus, the proposed shaper can be characterized by determining the FWHM of a group of identical signals produced with a precision pulse generator. The FWHM results for one pulse and the mean of five pulses ($\Lambda = 1$ and $\Lambda = 5$, respectively) are shown in Fig. 8. For this statistical analysis, 10,000 pulses of equal height were used throughout the test. The FWHM values were calculated from the pulse height histograms. These calculations were performed using Matlab, and the histograms were not fitted with any kind of function.

Although the energy error of the generated pulse should ideally be zero, it can be seen that the FWHM was inversely proportional

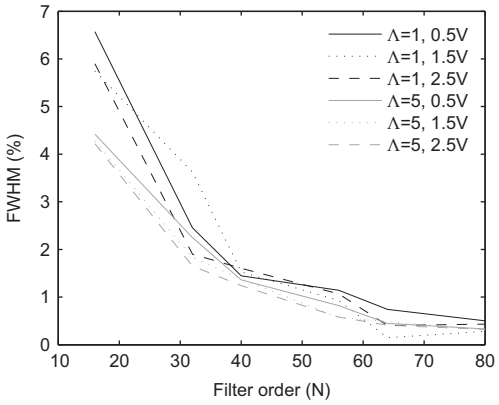


**Fig. 8.** Results of FWHM with $\Lambda = 1$ and $\Lambda = 5$ for different shaper orders and amplitude levels of the signal that arrives at the digital shaper after the ADC stage.

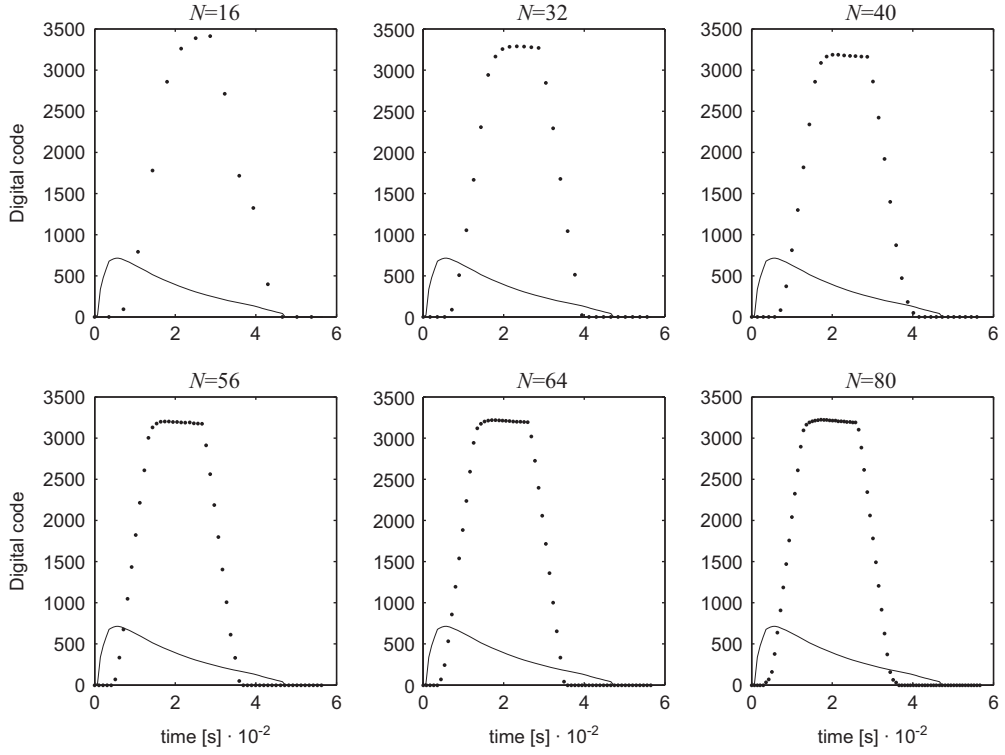


**Fig. 7.** Input signal (solid line) and its corresponding trapezoidal shape (dotted line) for different orders of the shaper $N$ using one learning iteration.

to the order of the shaper. Thus, the FWHM increased because of a phase shift between pulses due to the sampling frequency. For this reason, the gap between the shapers trained with $\Lambda = 1$ and $\Lambda = 5$ became smaller when $N$ was increased. Therefore, for a high $N$, adjusting the shaper with just one pulse may be sufficient.

As stated at the beginning of this section, both shaping and peak detection stages were synthesized in the same FPGA using a Xilinx ISE 13.1 developing tool. Table 2 shows the resources used and the power consumption, measured using current probes. As can be seen, power consumption rose slowly as the order of the shaper increased.

## 6. Experimental results using a neutron monitor

Lastly, a test to check the proposed shaper was performed. The main objective of this test was to obtain similar results to those obtained in experiments carried out with a non-adaptive shaper. This test was performed in the Castilla-La Mancha Neutron Monitor (CaLMa) located in Guadalajara, Spain. The instrument consisted of 15 proportional gas counter tubes. More information about features, setup and results of this instrument can be found in Ref. [16]. In both the cited experiment and the present test, an LND2061 tube 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. 9 and 10. The Data Acquisition Board (DAQ) was a proprietary design, whereas the Digital Processing Board was an evaluation board (Xilinx ML410).

In real particle detectors, the pulses can have a different duration and amplitude. This is the case of the detector at this facility. When this occurs, the user must define a region of interest based on previous experiences (according to Fig. 12(a), between channels #260 and #300). If it is not possible to define a region of interest, the adaptive shaper simply operates as an adaptive FIR filter. However, the main objective of a shaper is not to obtain an

**Table 2**
Comparison of FPGA (Xilinx XC4VFX60-11FF1152) performance and power consumption when implemented with different $N$.

| $N$ | DSP48 (%) | Slices (%) | FPGA power consumption (mW) |
|---|---|---|---|
| 16 | 8 (6%) | 622 (2%) | 265 |
| 32 | 16 (12%) | 1013 (4%) | 275 |
| 40 | 20 (15%) | 1209 (5%) | 280 |
| 56 | 28 (21%) | 1519 (6%) | 285 |
| 64 | 32 (25%) | 1811 (7%) | 295 |
| 80 | 40 (31%) | 2220 (8%) | 310 |

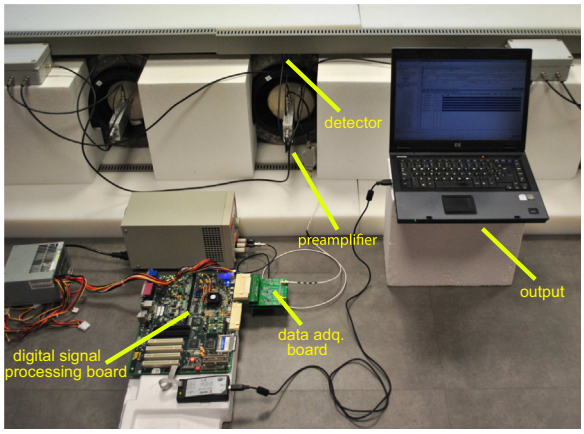


**Fig. 10.** Image of the setup of the neutron monitor for this experiment.

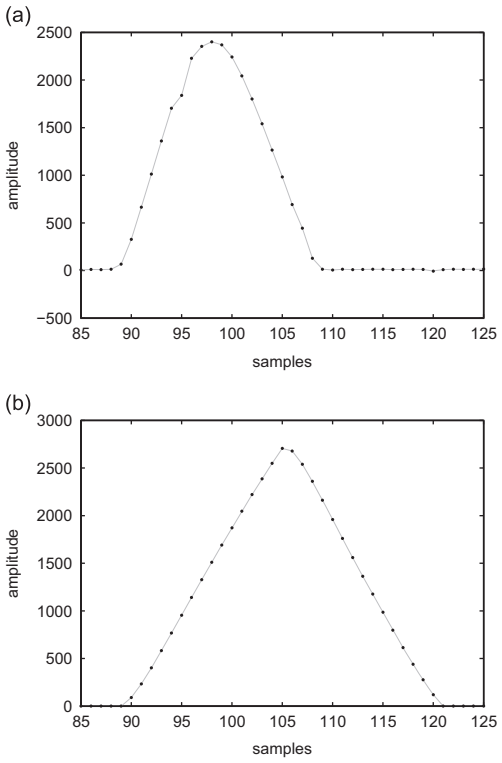


**Fig. 11.** Results obtained with the adaptive shaper inside the FPGA using a pulse coming from the preamplifier. The signal was sampled at 10 MHz. (a) Pulse event. (b) Corresponding triangular shape.

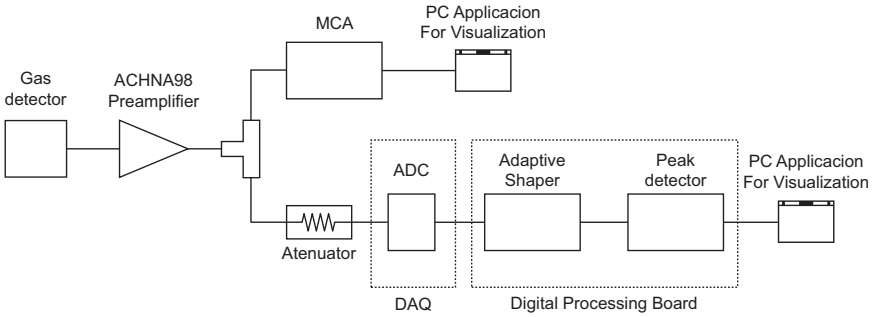


**Fig. 9.** Setup of the neutron monitor experiment.

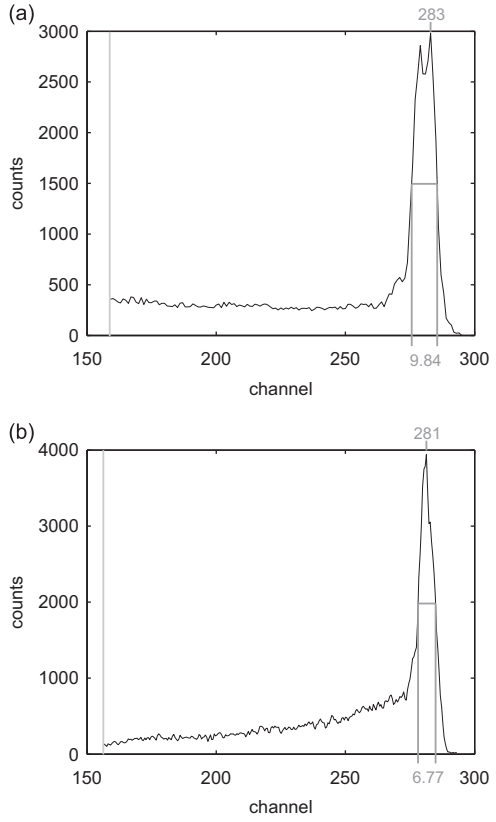


**Fig. 12.** Histogram for pulses obtained using (a) the AmpTek 8000A Multichannel Analyzer and (b) the proposed adaptive shaper. The grey lines indicate the threshold level.

output signal with a "perfect" shape, but to obtain the maximum SNR.

To perform the test using the adaptive shaper, the preamplifier was connected to a proprietary data acquisition system with an ADC core. This device was a 14-bit Linear LTC2171 working at a dynamically selectable frequency of up to 40 MS/s. Following the ADC, a ML410 board, whose core was the digital system explained in Section 5.2, included the described adaptive shaper and the peak detector. Both boards transmitted and received data through LVDS signals. We selected the beginning of the region of interest (channel #260) as the learning threshold of the shaper.

An optimal shaper depends on the facility's noise type. Thus, it should be calculated by means of a procedure (e.g. [3] or [7]) to enable the adaptive shaper to transform the input signal into the optimal shape.

Using a spectrum analyzer, it was observed that the predominant noise was white noise. Therefore, in accordance with Ref. [7], a triangular shaping for the region of interest with $N=32$ and $f_s = 1/T_s = 10\,\mathrm{MS/s}$ was used for this setup. The histogram obtained is shown in Fig. 12(b) whereas the histogram obtained with the Multichannel Analyzer is shown in Fig. 12(a). Both histograms were created using the 65,000 samples with the highest amplitude from a set of 90,500. The measured energy spectrum was 2.31 MeV (channel #300), and the duration of the test was 8 h. Thus, the average event rate was 3.1 per second.

In the case of using the adaptive shaper, a FWHM equal to 6.77 (2.41%) was obtained, whereas in the case of the histogram created with the MCA, a FWHM equal to 9.84 (3.48%) was obtained. It is important to consider that the MCA does not apply any shaping, whereas in this test, the triangular shaping of Fig. 11(b) was applied to reduce the white noise. 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. 9). In fact, the histogram of Fig. 12(a) was generated using an analog MCA with no previous digital shaping whereas the histogram of Fig. 12(b) was generated using an ADC, digital adaptive shaping and digital peak detection.

## 7. Conclusions

In this study, a new adaptive digital shaper for real-time spectroscopy was designed and implemented in an FPGA. This proposed adaptive shaper is an effective means of generating a desired shape by automatically adjusting the coefficients of the shaper using the input signal during a learning period. A high sampling rate increases power consumption, but a low sampling rate could deteriorate the FWHM; therefore, a good reference point must involve a trade-off between these parameters. The most common pulse shape methods were successfully tested in simulations. As regards the type of noise present in a real benchmark, the optimal shaper for this noise was tested. As with the other non-adaptive shapers, the FWHM of the histograms was reduced when the shaper order was increased, due to sampling effects. Lastly, we have shown that the increase in resolution when shapers are trained using five signals instead of one is negligible for high order shapers. The proposed shaping method allows for the generation of histograms with a resolution comparable to classical analog shaping alternatives. Thus, this represents a promising method, regardless of whether power consumption or simplicity is the main consideration.

## Acknowledgments

This project was funded by the Spanish Administration as part of project ref. AYA2012-39810-C02-02. The authors thank the CaLMa Team for technical support and for providing the CaLMa facility.

## References

[1] G.F. Knoll, Radiation Detection and Measurement, John Wiley & Sons, Inc, 2000.
[2] P.W. Nicholson, Nuclear Electronics, John Wiley & Sons, Ltd, 1973.
[3] E. Gatti, A. Geraci, G. Ripamonti, Nucl. Instrum. Methods Phys. Res. A 381 (1996) 117.
[4] D. Alberto, et al., Nucl. Instrum. Methods Phys. Res. A 611 (2009) 99.
[5] A.M. Fernandes, et al., Fusion Eng. Des. 85 (2010) 308.
[6] R. Abbiati, A. Geraci, G. Ripamonti, IEEE Trans. Nucl. Sci. NS-52 (October (5)) (2005).
[7] F.S. Goulding, Nucl. Instrum. Methods 100 (1972) 493.
[8] M. Nakhostin, IEEE Trans. Nucl. Sci. NS-58 (October (5)) (2011).
[9] V.T. Jordanov, Nucl. Instrum. Methods Phys. Res. A 670 (2012) 18.
[10] E. Gatti, A. Geraci, S. Riboldi, G. Ripamonti, Nucl. Instrum. Methods Phys. Res. A 523 (2004) 167.
[11] A.V. Oppenheim, R.W. Shafer, Discrete—Time Digital Signal Processing, 3rd ed., Pearson Education, 2010.
[12] A. Abba, A. Geraci, IEEE Trans. Nucl. Sci. NS-59 (October (5)) (2012).
[13] J. Beringer, et al., Particle Data Group, Phys. Rev. D86 (2012) 010001 http://dx.doi.org/10.1063/1.1898643.
[14] V.T. Jordanov, G.F. Knoll, Nucl. Instrum. Methods Phys. Res. A 345 (1994) 337.
[15] C. Imperiale, A. Imperiale, Measurement 30 (2001) 49.
[16] J. Medina, et al., Nucl. Instrum. Methods Phys. Res. A (2013) http://dx.doi.org/10.1016/j.nima.2013.06.028i.