Digital multishaping for white noise reduction and its implementation on programmable logic

Authors: Regadío , Dr. Alberto, G. Tejedor, Dr. J. Ignacio, Juan, Dr. J. Blanco, Sánchez-Prieto, S, Esteban, Dr. Luis, Regadío , Dr. Alberto

Date: 2025-04-10T18:41:48+00:00

Abstract

One of the most common noise type in particle physics detectors is white noise. Its impact inversely correlates with the shaping time of the signal. An extended shaping time would significantly improve resolution in pulse height measurement. However, prolonged shaping times elevate the risks associated with pile-up and detector saturation, particularly under conditions of high particle arrival rates. Therefore, it is necessary to reach a compromise on the shaping time in order to achieve a balance between these opposing effects. The originality of this study is to process pulses of a single particle detector with multiple shapers in parallel instead of one, enabling independent pulse analysis for each shaper. Subsequently, a Finite State Machine (FSM) is employed to select the optimal pulse height based on the current pulse arrival rate. The proposed method was implemented and evaluated using both Python and programmable logic. The results show how effective this technique is at reducing noise, which is shown through the analysis of energy histograms. Thus, this strategy offers a promising approach to enhancing particle detector performance, especially at fluctuating particle arrival rates.

Full Text

Preamble

Digital multishaping for white noise reduction and its implementation on programmable logic∗

Alberto Regadío,1,† J. Ignacio García Tejedor,1 Juan José Blanco,1 Sebastián Sánchez-Prieto,1 and Luis Esteban2
1Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain
2EERE Department, Oregon Institute of Technology, Oregon State University, Campus Dr SE 97601, Klamath Falls, USA

One of the most common noise types in particle physics detectors is white noise. Its impact inversely correlates with the shaping time of the signal. An extended shaping time would significantly improve resolution in pulse height measurement. However, prolonged shaping times elevate the risks associated with pile-up and detector saturation, particularly under conditions of high particle arrival rates. Therefore, it is necessary to reach a compromise on the shaping time in order to achieve a balance between these opposing effects. The originality of this study is to process pulses of a single particle detector with multiple shapers in parallel instead of one, enabling independent pulse analysis for each shaper. Subsequently, a Finite State Machine (FSM) is employed to select the optimal pulse height based on the current pulse arrival rate. The proposed method was implemented and evaluated using both Python and programmable logic. The results show how effective this technique is at reducing noise, which is shown through the analysis of energy histograms. Thus, this strategy offers a promising approach to enhancing particle detector performance, especially at fluctuating particle arrival rates.

Keywords: Digital filters, Digital shapers, Noise mitigation, Pile-up, Pulse height

Introduction

In particle spectroscopy, particle detector chains determine the energy of incoming particles while they have to cope with noise. In this scenario, the shaping stage is a fundamental component.

The use of digital electronics in particle detection chains has replaced the use of analog equivalents thanks to the development of integrated circuits because of their advantages (multistage integration within a single integrated circuit, reduced volume, and increased reconfigurability, among others). Digital pulse shaping aims to achieve the same objective as analog shaping, namely signal filtering to enhance system resolution. The versatility of digital shaping allows the implementation of configurations such as triangular, trapezoidal, cusp-like, or flat-topped cusp-like shaping [1] that would be more challenging to achieve using analog electronics.

When considering digital shaping as a filtering process, two fundamental implementations of digital shapers must be considered: Infinite Impulse Response (IIR) filters and Finite Impulse Response (FIR) filters [2, 3]. The types of noise mitigated by both can be effectively analyzed in the frequency domain. However, in particle detection, it is also common to analyze them in the time domain [4–7]. Although there are generalizations [8, 9], three noise components—white noise, $1/f^2$-noise (also called Brownian), and 1/f-noise—are usually dominant in a signal processing chain for acquisition of the signals from capacitive sources.

The effect of these noise types in a detection chain can be modeled using the Equivalent Noise Charge (ENC) formula presented in [6, 7, 10]:

$$Q_n^2 = \left(F_i \frac{C_i^2}{T_s}\right) i_n^2 + \left(F_v \frac{C_i^2}{T_s^2}\right) e_n^2 + F_{nf} C_i^2 e_{nf}^2$$

where Q_n is the ENC in Coulombs, i_n is the current noise spectral density measured in $A/\sqrt{Hz}$, e_n and e_{nf} are the voltage noise and 1/f-noise spectral density measured in $V/\sqrt{Hz}$, and C_i is the equivalent detector capacitance (in Farads). F_i, F_v, and F_{nf} are the noise indexes for white noise (in units of 1/s), 1/f^2-noise (in s), and 1/f noise (dimensionless), respectively. This formula is applicable to both analog and digital shapers.

The influence of 1/f-noise remains independent of the shaping time. The impact of 1/f^2-noise, characterized by its 1/f^2 noise spectrum, is proportional to the shaping time. In the case of trapezoidal or triangular shaping (Fig. 1 [Figure 1: see original paper]), it is straightforward that the impact of this noise type is $F_i^2 \propto \tau_r + \tau_p$, where F_i^2 is the noise index of the 1/f^2-noise, τ_p is the duration of the plateau of the trapezoid, and τ_r is the rising time of the shaped signal. Notice that if the detection chain is dominated by 1/f^2-noise, the solution is straightforward: pulse shaping with very short shaping times whenever the ballistic deficit allows it and we are interested in losing as few pulses as possible.

In contrast to 1/f^2-noise, white noise tends to have an inverse relationship with shaping time, suggesting that longer shaping times could effectively reduce errors in pulse height measurements. Namely, for trapezoidal or triangular shaping its impact ([4], among others) is:

$$F_v^2 \propto \frac{1}{\tau_r + \tau_p}$$

where F_v^2 is the white noise index.

According to these formulas, there exists an optimal shaping time τ^* that minimizes the ENC. There are algorithms available to optimize pulse shaping with the objective of minimizing noise, including analytical methods (e.g., [8, 11–13]), neural network-based approaches (e.g., [14]), and genetic algorithms (e.g., [15]), among other methods. Nonetheless, in certain situations, the search for τ^* is difficult. In addition, many detectors experience fluctuations in particle arrival rates over time, making this optimization more difficult. Therefore, the optimal shaping time becomes a dynamic parameter, requiring careful consideration and adjustment to maintain optimal performance.

Without aiming to be exhaustive, while 1/f^2 noise remains significant in many detectors due to thermal agitation of charge carriers—particularly in semiconductor-based detectors—and 1/f noise can arise from fluctuations in the electron multiplication process, the main source of white noise is typically the preamplifier electronics [16]. The impact of this white noise is amplified when the event arrival rate is very high, requiring a low shaping time to prevent pile-up and saturation. For the purposes of this manuscript, we will assume that serial noise is the dominant noise factor. Note that a short shaping time does not necessarily cause a ballistic deficit as long as it was high enough to avoid this problem [17].

Pulse processing has been developed for particle detectors to enhance resolution and mitigate pile-up effects at the same time [3]. In addition to classical methods for avoiding pile-up (e.g., [18]), another common approach involves reducing convolution time through windowing [19] and unfolding (i.e., deconvolution) [20–23], including some variations [24], albeit at the expense of increasing white noise. Many of these approaches are implementable in digital electronics. However, the cited works do not employ more than one shaper working in parallel.

In [13], a multi-parametric analysis is conducted using delay-line based shapers to determine pulse heights, enabling high-performance particle energy measurements even at high rates. This method ensures accurate dead time corrections and high-resolution energy spectra. A pile-up correction method is introduced in [25] using several shapers for pulse reconstruction and analysis carried out analytically. In [26], instead of employing different shapers, specific pulse shapes generated by α and β particles are used to precisely measure pulse heights. A real-time algorithmic pile-up correction on pulses of known shape using diverse modules in the detection chain is presented in [27]. In [24], energy recovery for individual pulses in pile-up events is also achieved with algorithms. Both methods achieve good resolution with a small number of rejected pulses.

Pile-up correction has also been tackled using evolvable algorithms [28] and deep-learning [29, 30]. With respect to the latter, although highly effective, they rely on nonlinear processing and require prior training and a certain level of computing.

The objective of this method is not to rely on a single detector chain, but rather to utilize multiple detection chains concurrently, further minimizing pile-up and saturation effects. An implementation of a similar method was performed in [31], although in this case the shaping is digital and at the end we add additional electronics for pulse selection as we will see later. Each chain employs its own shaper, facilitating independent measurement of pulse height and allowing for the selection of the most suitable height based on the pulse arrival rate. One advantage of the method outlined in this article is its flexibility: the employed shapers can be changed without needing a redesign of the rest of the circuit elements. This versatility also lends itself to ease of implementation in digital logic. The approach aims to optimize noise influence within the detection chain, particularly in scenarios dominated by white noise.

Fig. 2 [Figure 2: see original paper] illustrates an example of the optimal shaping time τ^*. In this context, pile-up and saturation effects broaden the noise spectrum, i.e., increase the ENC, represented in this Figure as additional serial white noise. When this happens, it is necessary to adjust the shaping time to

$\tau^\dagger$ to mitigate these effects, minimize the Equivalent Noise Charge (ENC), and enhance resolution. Pile-up and saturation, which can be modeled as temporally correlated noises, were plotted as $1/f^2$ noise although the exponent value could change as long as it is greater than 1.

The rest of the paper is organized as follows: In Section II the proposed method is explained. An analysis of this method within its expected efficiency is exposed in Section III. The results are presented in Section IV. Finally, Section VI summarizes the conclusions of this work.

II. Method

To implement this method, we depart from the conventional approach of employing a single detection chain with a shaping stage preceding the Pulse Height Analysis (PHA) stage. For this setup, M parallel detection chains are used and the pulse height analysis is determined by a Finite State Machine (FSM). The general scheme for $M = 3$ detection chains is depicted in Fig. 3 [Figure 3: see original paper].

Each detection chain has its own digital shaper. These shapers $(0, \cdots, M-1)$ are FIR and time-invariant and have the particularity that they provide the same pulse shape but with shaping times $\tau_0, ..., \tau_{M-1}$ such that $\tau_0 < ... < \tau_{M-1}$.

To validate the concept, synthetic pulses with a height $h = 0.5$ were generated. Noise was then added to the signal using a random normal distribution with a standard deviation N, so the noise power P_N is N^2. Observe that P_N is directly proportional to e_n^2 of Eq. (1). In triangular shaping, the energy of a triangular pulse with height h and duration τ is $h^2\tau/3$. Thus, the signal power P_S is $h^2/3$. For this manuscript, triangular shapers (i.e., $\tau_p = 0$ in Fig. 1) were chosen for their simple implementation as Finite Impulse Response (FIR) filters in digital electronics, and because a long plateau time increases the $1/f^2$-noise according to Eq. (2).

Throughout this article, $M = 3$. Note that while reducing the number of detection chains to two may lead to a decrease in the efficiency of this method, increasing the number of chains would require additional hardware resources and complicate the FSM. We used triangular shapers with different shaping times. However, other shapers could be chosen. An exhaustive comparison between combinations of shapes is beyond the scope of this paper. Fig. 4 [Figure 4: see original paper] shows such shapers, which have been passed through the three amplification stages. In this figure and in the following ones, all pulse weights are normalized, with 1.0 being the saturation level.

The Pulse Height Analysis (PHA) stage comes after the shaping stage. In this study, only pulses with heights above a certain threshold (0.1) are detected, and the detection of the next pulse height cannot occur until the signal falls back below this threshold. The threshold can be adjusted based on the noise conditions of the detection chain. While more advanced versions of PHA exist [32], including dynamic threshold level, the aim of keeping the PHA stage as simple as possible is to emphasize the effect of shapers on pulse height measurements.

The method operates as follows: the pulse heights obtained from the three chains (C0, C1, and C2) are fed into the FSM of Fig. 5 [Figure 5: see original paper], as explained at the beginning of this section. This FSM, which works as a Mealy machine, serves as the core of the proposed method, and its state transition diagram is depicted in Fig. 5. The FSM comprises three states (S0, S1, and S2), with the initial state being S0.

During periods of low pulse rates, the proposed method operates in the following manner: it starts in the S0 state, and when a pulse arrives, due to $\tau_0 < \tau_1 < \tau_2$, C0 calculates the pulse height first, followed by C1 and finally C2. The output $y_2[n]$ from C2 is then displayed as the output $y[n]$. This sequence of the FSM (0/-, 1/-, 2/2) ensures that another pulse has not been detected in between, and the resulting height will have the least noise according to (3), in this case, from C2.

In scenarios with high pulse rates, a C0 or C1 event may occur before the sequence of events from all three chains concludes. In such cases, the height calculated in these chains, $y_0[n]$ or $y_1[n]$, respectively, will be displayed as the output $y[n]$ instead of $y_2[n]$. In summary, when a sequence of events from the three chains is interrupted, the output $y[n]$ will display the event captured by the chain where the sequence has been interrupted, typically the one that provides greater resistance to white noise due to its higher shaping time.

Finally, it is possible that when a single pulse is detected, the heights measured by different PHAs may vary significantly due to pile-up or saturation effects. In this case, the output of C0 $y_0[n]$ is displayed at $y[n]$ because the pulse height calculated with C0 is more reliable, as its shaper is less prone to saturation or pile-up effects compared to the others. This comparison is shown in Fig. 5 with asterisks ($*$) at the right of the output. The discrepancy level is a parameter that has been set at 10% for the tests.

To perform a proof of concept and to evaluate the results, the code was first coded in Python using the Numpy library [33]. An example of output is shown in Fig. 6 [Figure 6: see original paper] where the input signal $x[n]$ and the output signal $y[n]$ are observed. We can see that when there is no overlap, the sequence S0, S1, S2 is produced, returning the captured pulse of the chain that offers less noise.

The objective is to adapt to the varying frequencies of pulse arrivals without losing any of them. When the arrival rate is very low, the detection chain that effectively filters the noise, typically C2, is prioritized due to its high τ compared to the others. In contrast, when pile-up and/or saturation occurs, the chains that return the pulses are C1 or C0. This minimizes the loss of pulses due to a low τ by the shaper while improving the resolution of the detection chain, similar to one with a shaper featuring a high τ.

All of this is achieved by configuring the digital shaper, replicating the Pulse Height Analysis (PHA), and implementing the described FSM. Note that this method assumes that the arrival time of each detected event can be delayed within an interval between 0 and $\tau_{M-1} - \tau_0$. While this does not pose a significant issue for creating histograms, it may impact measurements that rely on precise timing, such as Time Of Flight (TOF) and (anti)coincidence measurements. Nevertheless, this effect can be corrected if we know which detection output $y_i[n]$ is selected at the output $y[n]$ to return the pulse height.

III. Efficiency of This Method

As stated in Section I, the main objective of this method is reducing the noise index F_v^2. With this goal in mind, we define P_k as the probability that the detection chain k is the one that performs the PHA of the event.

The value of P_k is calculated when given an event, what is the probability that in a given interval the next event will occur again. If that interval is within the shaping time, the pile-up will happen. P_k can be modelled as a Poisson distribution. According to it, the probability that given an average interval of events λ per second, the rate that an event occurs in a time interval $[\tau_k, \tau_{k+1}]$ is:

$$P_k = P(\tau_k < t \leq \tau_{k+1}) = p(t \leq \tau_{k+1}) - p(t \leq \tau_k) = (1 - e^{-\lambda \tau_{k+1}}) - (1 - e^{-\lambda \tau_k})$$

Given M detection chains ordered by the shaping time, where 0 is the one with the minimum time and $M-1$ the one with the maximum time. The probability of no pile-up for all detection chains is:

$$P_{M-1} = p(\tau_{M-1} < t \leq \infty) = 1 - (1 - e^{-\lambda \tau_{M-1}})$$

In this case, the shaper that carries out the shaping is $M-1$. Similarly, if $P(\tau_{M-1} < t \leq \tau_{M-2})$ then the shaping is performed by $M-2$ and so on until the shaper of C0.

The probability defined as P_{-1} that two events were so close together that even the shaper of C0 could not avoid the pile-up is:

$$P_{-1} = P(0 < t \leq \tau_0) = 1 - e^{-\lambda \tau_0}$$

Therefore, we can calculate a new term F^{*2} that stands for the new effective white noise index and replaces F_v^2. This new term is equal to the weighting mean that the detection chain k calculates the pulse height $F_{v,k}^2$ defined as the value of F_v^2 for the shaper k. This weighting mean is calculated with the probability P_k in this way:

$$1 - P_{-1}\text{(cid:88)}P_{kF}^2$$

Namely, with this method, F_v^2 can be replaced by:

$$\frac{1}{1 - P_{-1}} \sum P_{kF_{v,k}}^2$$

The term prior to the summation is the inverse of the probability that the pulse height can be calculated with no pile-up and can be considered as the partition function. This equation replaces F_v^2. The particular value for trapezoidal and triangular shapers was shown in Eq. (3).

With this equation, given τ_0, that is, the percentage of pulses lost, the main objective is to minimize F^{*2}. For this, the following formula can be used:

$$\frac{\partial F^{*2}}{\partial \tau_k} = 0, \forall k \in \{0 \dots M - 1\}$$

and then solve the system of equations. Our experience tells us that if $M > 2$, the system is complicated to solve or approximate analytically and other methods should be employed.

To optimize Eq. (7) as a function of τ_k as discussed in (8), the gradient descent algorithm was programmed in Python using the Tensorflow package [34]. For this purpose, a fixed value of τ_0 was set before running the optimization as a function of the number of pulses one is willing to lose according to (6). The results for the new noise index are shown in Fig. 7. For clarity in the results shown, optimizations were carried out for $M = 1, 2, 3, 4$. The case of $M = 1$ (the case of a classical detection chain) was also calculated for comparison purposes; this particular case does not require optimization, it is sufficient to use (3).

The first observed in this Figure is that F^{*2} decreases as we are willing to lose pulses. Reductions of around 65% are also observed when we switch to using 2 chains instead of 1. These reductions continue to occur, although to a lesser extent as we increase M beyond 4. Notice that these values are the theoretical minimums for white noise and that other factors such as saturation voltage, which for the sake of clarity was not modeled in this section, can increase it.

As already explained, the optimization of (7), fixing τ_0, also yields the optimal value of the shaping times. These values are shown in Fig. 8 [Figure 8: see original paper] for $M = 2, 3, 4$. Note that in all three cases, if one is willing to lose more pulses, the shaping time of the other shapings $(\tau_1, \dots, \tau_{M-1})$ is lengthened to produce the results shown in Fig. 7. Recall that the values shown in the last two figures are for white noise and for a triangular shaping. If the parallel white noise had been considered, another equivalent parallel white noise F^{*2} would be calculated in the same way as in (7) and would cause an increase in the values of the noise index in Fig. 7 for high shaping times yielding curves similar to those in Fig. 2.

IV. Results

To validate the performance of the presented method, a series of simulations and tests were performed. The results presented in this study were also obtained using Python, employing the same packages enumerated in Section II for ease of result handling. It is important to note that both the Python and VHDL implementations exhibit the same behavior. This ensures the consistency and reliability of the findings across different implementation platforms.

In this test, pulses were generated with a frequency f measured in events per second over a duration of 2000 seconds. All pulses had identical heights set at 0.5, representing the mean value between 0 (no signal) and 1 (saturation). The reason is that, in the absence of noise and pile-up, the histogram should exhibit a delta centered at 0.5. However, the presence of noise and pile-up will inevitably broaden the noise spectrum, resulting in a typical deviation σ of the measurement. The σ value is directly proportional to the Full Width at Half Maximum (FWHM), namely, FWHM $= 2.35 \cdot \sigma$. Also, remember that ENC presented in (1) and FWHM in turn are also directly proportional [16]. Therefore, disregarding all non-white noise types, we can conclude that σ is proportional to F_v.

For this test, a maximum pulse loss of 9.5% was chosen. These pulses can be recovered with a subsequent preprocessing step, such as the method described in [18], but in this work, we will consider them lost. According to (6), this constraint yields $\tau_0 = 10$ ms. Applying the optimization method of F^{*2} shown in Section III (Figure 8), we obtain that for $f = 10$ events/s, the optimal shaping times to minimize the equivalent white noise are those shown in Table 1 .

In the histograms shown in Fig. 9 [Figure 9: see original paper], the variance difference becomes evident as white noise is increased. Notably, with C0, the width of the histogram, represented by σ, is the highest. Furthermore, it is apparent that C2 exhibits a low σ but fails to capture all the pulses that have occurred. Additionally, when employing the proposed method (Multishaper), it is observed that σ resembles those of C2 but manages to capture more pulses. Finally, notice that the percentage of contribution of each shaper is practically the same as those in Table 1.

We also compute histograms again during 2000 s to examine how frequency influences the histogram due to pile-up effects. In this context, frequency f is defined as the mean number of input pulses per second. When noise is limited but the pulse frequency increases, it becomes evident that not all pulses are captured by the shapers with a higher shaping time, resulting in count losses, as depicted in Fig. 10 [Figure 10: see original paper].

When testing a combination of noise values with frequencies, we observe that the pulse widths have the behavior shown in Fig. 11 [Figure 11: see original paper]. Notice that when a high f is selected, pile-up and saturation effects are very common, and the output is almost always the output of C0 (bottom right panel of Fig. 11). For the same reason, when a low f is selected, the output is almost always the output of C2 reducing σ by approximately half for $N \approx 0.1$ (top left panel of Fig. 11).

Moreover, when testing a combination of noise values with frequencies, we observe that the percentage of captured pulses exhibits the behavior shown in Fig. 12 [Figure 12: see original paper]. We note that this shaper captures as many pulses as C0 does, i.e., a value close to 100%.

To summarize, resolution notably improves with high pulse arrival rates. This involves employing multiple linear shapers with nonlinear treatment (the FSM) applied at the end. Naturally, the extra captured pulses using this method may exhibit more noise, but they are at least captured.

When the arrival frequency of pulses remains constant, employing this method may not be necessary. Instead, a detection chain with an optimal shaping time is sufficient. However, this is not typically the case in most situations. The best results for this method are observed when the detector processes pulses with variable arrival rates.

In this approach, pulse capture has been simplified to its maximum extent. While more complex systems, such as those involving deep learning (e.g., [29, 35]), may yield better results, implementing them on reconfigurable devices poses additional challenges unless implemented in software.

Implementation in Configurable Logic

To test the proposed method in a real detection chain, the scheme depicted in Fig. 3 has been implemented in a programmable FPGA device. Instead of a complete system with several channels as in [36], in this section we will focus specifically on the module that implements the proposed algorithm.

For this project, tools from the Xilinx family were employed to synthesize the design in a Spartan-7 FPGA, specifically a XC7S75FGGA676-1 device. The implementation was executed using generic VHDL without relying on any specific component of the device. As a result, synthesizing this design in any other FPGA from a different manufacturer should not pose any issues.

The coding has been divided into (a) shaping, implemented as an FIR filter; (b) PHA according to the explanation in Section II; (c) FSM that yields the output pulse heights. Fig. 13 [Figure 13: see original paper] shows a general schematic of the connections between the shapers, PHA and FSM. As can be seen, it is similar to the one in Fig. 3.

All the interfaces have in common the resolution in bytes W of the signals. Apart from signals CLK (clock signal) and RST (reset signal), which are common for all the proposed interfaces, in the case of the shaper, the input signals are: X: input signal from the ADC; XEN: input signal enable. The output signals are: Y: shaped signal; YEN: shaped signal valid. The shaping type and duration are configured internally.

The input signals of the PHA are: X: input signal from shapers; XEN: input signal enable. The output signals are: Y: pulse height (zero if not pulse detected); YEN: pulse height detected; LAST: height of the last pulse detected. The threshold level (see Section II) is a parameter of this component selected during synthesis.

Finally the input signals of the FSM are: IN0, IN1, IN2: pulse heights detected; IN0EN, IN1EN, IN2EN: pulse heights detected valid; LAST0, LAST1, LAST2: height of the last pulses detected by PHAs. The output signals are: Y: pulse height (zero if not pulse height detected). Table 2 shows the components used for $W = 14$ in a Xilinx XC7S75FGGA676-1. This FPGA is one with more limited resources within the Spartan-7 family. However, the proposed design fits perfectly. Note that specific resources from this architecture or from Xilinx, such as Digital Signal Processors (DSPs), were not used. Instead, generic resources have been employed to make the design transferable to other programmable devices, even those different from those provided by Xilinx. If an FPGA with more resources had been used, and specific resources had been employed, it would allow replicating the proposed method for multiple detectors using a single device.

VI. Summary

The evaluation of pulses coming from particle detectors using several shapers at a time instead of one and evaluating the results of all of them at a later stage improves the resolution and the number of captured events. This improvement stems from the ability to adapt shaping times based on the particle arrival rate: longer shaping times mitigate white noise, while shorter shaping times help avoid pile-up and saturation. The originality of this study is to use multiple shapers in parallel with a single detector, enabling independent pulse analysis for each shaper, and minimizing the pulse loss and improving the resolution of the detection chain. Despite this scheme can be implemented for an arbitrary number of shapers, in this article we focused on 3 shapers. While adjusting the number of shapers may complicate the presented finite state machine, changing the shaping time is practically instantaneous, allowing for adaptation to each specific situation. The results were presented. Finally, we demonstrated that the approach is fully implementable in both software and configurable logic.

References

[1] J. Liu, J. Yang, G.-Q. Zeng, K.-Q. Zhang, S.-L. Wei, and L.-Q. Ge, Nuclear Science and Techniques 28, 1 (2017).

[2] V. T. Jordanov and G. F. Knoll, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 345, 337 (1994).

[3] J.-X. Li, H.-L. Hou, Y.-F. Huang, M.-S. Cheng, and Z.-M. Dai, Nuclear Science and Techniques 34, 165 (2023).

[4] F. Goulding, Nuclear Instruments and Methods 100, 493 (1972).

[5] F. S. Goulding and D. A. Landis, IEEE Transactions on Nuclear Science 29, 1125 (1982).

[6] V. Radeka, "Signal processing for particle detectors," in Particle Physics Reference Library: Volume 2: Detectors for Particles and Radiation, edited by C. W. Fabjan and H. Schopper (Springer International Publishing, Cham, 2020) pp. 439–484.

[7] P. Zyla et al. (Particle Data Group), PTEP 2020, 083C01 (2020).

[8] E. Gatti, A. Geraci, and G. Ripamonti, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 381, 117 (1996).

[9] A. Regadío, J. Tabero, and S. Sánchez-Prieto, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 811, 25 (2016).

[10] H. Spieler, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 636, S149 (2011), 7th International "Hiroshima" Symposium on the Development and Application of Semiconductor Tracking Detectors.

[11] V. Radeka, IEEE Transactions on Nuclear Science 15, 455 (1968).

[12] E. Gatti and P. F. Manfredi, La Rivista del Nuovo Cimento (1978-1999) 9, 1 (1986).

[13] G. Gerardi and L. Abbene, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 768, 46 (2014).

[14] Q. Wang, H. Huang, X. Ma, Z. Shen, C. Zhong, W. Ding, W. Zhou, and J. Zhou, Applied Radiation and Isotopes 190, 110515 (2022).

[15] A. Regadío, S. Sánchez-Prieto, J. Tabero, and D. M. González-Castaño, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 795, 115 (2015).

[16] Glenn G. Knoll, John Wiley & Sons, Inc. (John Wiley & Sons, 2010).

[17] B. Loo, F. Goulding, and D. Gao, IEEE Transactions on Nuclear Science 35, 114 (1988).

[18] C. Imperiale and A. Imperiale, Measurement 30, 49 (2001).

[19] M. Wang, J. Zhou, X. Wang, H. Wang, Y. Liu, X. Hong, and T. Yang, Applied Radiation and Isotopes 196, 110772 (2023).

[20] V. T. Jordanov, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 805, 63 (2016), special Issue in memory of Glenn F. Knoll.

[21] X.-Y. Wang et al., Nuclear Science and Techniques 33, 46 (2022).

[22] V. T. Jordanov and K. V. Jordanova, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 1044, 167421 (2022).

[23] A. Regadío and S. Sánchez-Prieto, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 888, 228 (2018).

[24] Z. Deng, X. Zhao, H. Ding, Y. Deng, X. Cao, and K.-V. Yuen, in 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC) (2020) pp. 1–6.

[25] M.-R. Mohammadian-Behbahani and S. Saramad, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 897, 1 (2018).

[26] S. Miłosz, K. Tudyka, A. Walencik-Łata, S. Barwinek, A. Bluszcz, and G. Adamiec, IEEE Transactions on Nuclear Science 64, 2536 (2017).

[27] M. Bolic, V. Drndarevic, and W. Gueaieb, IEEE Transactions on Instrumentation and Measurement 59, 122 (2010).

[28] H.-Q. Huang, X.-F. Yang, W.-C. Ding, and F. Fang, Nuclear Science and Techniques 28, 12 (2017).

[29] B. Jeon, S. Lim, E. Lee, Y.-S. Hwang, K.-J. Chung, M. Moon, IEEE Transactions on Nuclear Science 69, 1344 (2022).

[30] A. Regadío, L. Esteban, and S. Sánchez-Prieto, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 1005, 165403 (2021).

[31] H. Sakai, A. Uritani, Y. Takenaka, C. Mori, and T. Iguchi, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 421, 316 (1999).

[32] V. T. Jordanov and D. L. Hall, in 2002 IEEE Nuclear Science Symposium Conference Record, Vol. 1 (IEEE, 2002) pp. 140–144.

[33] C. R. Harris et al., Nature 585, 357 (2020).

[34] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," (2015), software available from tensorflow.org.

[35] L. Tang, K. Shi, H. Shen, and Y. Soh, IEEE Transactions on Nuclear Science 70, 2479 (2023).

[36] H.-Y. Shen, J.-L. Zhang, J. Zhang, and J.-H. Zhou, Nuclear Science and Techniques 35, 136 (2024).

∗ Supported by the project SBPLY/21/180501/000170, cofunded by the Junta de Comunidades de Castilla-La Mancha and by the European Regional Development Fund, FEDER. Supported by the project PID2019-107806GB-I00, funded by Ministerio de Ciencia e Innovación (Spain Government).

† Corresponding author, alberto.regadio@uah.es

The Python and VHDL code used for this method is available at https://github.com/arc140181/multishaping.

Note: Figure translations are in progress. See original paper for figures.