Regadío Carretero, A., Sánchez Prieto, S. & Esteban, L. 2019, "Filtering of pulses from particle detectors by means of Singular Value Decomposition (SVD)", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 922, pp. 257-264.

Available at https://doi.org/10.1016/j.nima.2019.01.028

Filtering of pulses from particle detectors by means of Singular Value Decomposition (SVD)

Alberto Regadío[a,*], Sebastián Sánchez-Prieto[b], Luis Esteban[c]

[a]*Electronic Technology Area, Instituto Nacional de Técnica Aeroespacial, 28850 Torrejón de Ardoz, Spain*
[b]*Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain*
[c]*Universidad Antonio de Nebrija, C/Pirineos, 55 28040 Madrid, Spain*

Abstract

This paper presents a novel methodology to filter pulses coming from particle detectors. It is based on variable-in-time convolutions in which one of the operands is the input pulse and the other is a vector that changes with every convolution step. This is equivalent to multiply every incoming pulse by a filtering matrix. The coefficients of this matrix are computed by applying a Singular Value Decomposition (SVD) factorization over a set of training pulses. A detailed explanation of this SVD-filtering methodology, a noise filtering analysis, simulations and filtering of pulses coming from a neutron monitor were carried out to verify its feasibility.

Keywords: Digital pulse processing, Pulse filtering, Noise, Dimensionality reduction, SVD

1. Introduction

The indirect or direct interaction of particles with the appropriate particle detector produces a charge build up or if accelerated, pulses of current. These pulses are converted to voltage by means of trans-impendance amplifiers and fed into subsequent processing stages. These voltage pulses carry useful information about the incident particles such as type, energy or angle of impact [1]. These outgoing pulses are always mixed with noise, limiting the accuracy of the measurement.

In particle detection systems the dominating noise is generated both in the detector and in the detector readout electronics, specially in the analog front-end. The readout electronics is typically implemented with discrete analog components (resistors, diodes, field-effect transistors, etc.) and by hence affected by parasitic effects. This noise causes the masking of information present in the signal.

Part of the noise is the result of fundamental physical processes and quantities such as the discrete nature of electric charge and therefore it cannot be avoided [1]. However, its effects can be reduced by using proper

*Corresponding Author
Email addresses: `regadioca@inta.es` (Alberto Regadío), `sebastian.sanchez@uah.es` (Sebastián Sánchez-Prieto), `lesteban@nebrija.es` (Luis Esteban)

noise filtering strategies. A key to successful noise filtering is to accurately model the noise that appears in the system. This noise model enables the design of filters that maximize the Signal-to-Noise Ratio (SNR).

Filtering of detector pulses is crucial in most nuclear pulse processing applications. A generic pulse filter attenuates or even removes a certain frequency interval of frequency components from a pulse. These frequency components are typically noise though there are situations in which a filter is used to remove some part of the pulse as well.

With the development of integrated circuits, digital electronics has also been used for particle detectors, even replacing the use of analog electronics in some detection stages such as shaping. The use of digital electronics provides many advantages, for instance, the inclusion of several stages in a single integrated circuit, lower volume and consumption or reconfigurability when implemented in Field Programmable Gate Arrays (FPGAs). However, this change could increase the complexity of particle detector backends and increase the number of noise sources that appear due to the new elements added such as Analog to Digital Converters (ADC) [2].

The two digital systems commonly used in particle detection systems are filters and pulse shapers. There are two types of digital filters, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). The FIR filtering of any pulse can be analyzed from the perspective of the convolution of a vector, which represents the filter. In this paper, the convolution is replaced by a multiplication matrix which is denominated filtering matrix. As exposed in Section 2, the multiplication by a filtering matrix is equivalent to making a convolution in which one of the elements is the input pulse and the other is a value that changes with every convolution step. To achieve an optimal filtering it is necessary to find out the matrix entries that optimize the SNR.

Singular Value Decomposition (SVD) is a non-parametric factorization of real (or complex) matrices. SVD is explained in more detail in Section 3. It has been already used in areas related to particle spectroscopy such as histogram creation [3] and pulse unfolding [4]. In [3] SVD is also used for filtering but, unlike the proposed method of this article, it obtains all the SVD values of the pulses at the same time (an autofiltering) without distinguishing between training pulses and real pulses. It has also been used to unfold entire histograms [5, 6]. In this work, the elements of the filtering matrix are obtained using the method explained in Section 4. This method is suitable for real-time implementation using either hardware or software. Finally, a filtering noise analysis is explained in Section 5 and the results of applying this method on pulses are explained in Section 6.

2. Convolution as a matrix multiplication

As stated in Section 1, the result of filtering a digital input pulse $\mathbf{x} = x[n]$ with a Finite Response Filter (FIR) in time-domain can be represented by a convolution. This operation, in turn, can be represented as a matrix multiplication whose impulse response $\mathbf{m} = m[n]$ is a Toeplitz matrix [7] (i.e. its descending diagonals from left to right are constant). For example, the convolution of $\mathbf{x}$ and $\mathbf{m}$ can be rewritten as

$$\mathbf{y} = \mathbf{m} * \mathbf{x} = \mathbf{M}\mathbf{x} = \begin{pmatrix} M_1 & 0 & \dots & 0 & 0 \\ M_2 & M_1 & \dots & \vdots & \vdots \\ M_3 & M_2 & \dots & 0 & 0 \\ \vdots & M_3 & \dots & M_1 & 0 \\ M_{l-1} & \vdots & \dots & M_2 & M_1 \\ M_l & M_{l-1} & \ddots & \vdots & M_2 \\ 0 & M_l & \dots & M_{l-2} & \vdots \\ 0 & 0 & \dots & M_{l-1} & M_{l-2} \\ \vdots & \vdots & \vdots & M_l & M_{l-1} \\ 0 & 0 & 0 & \dots & M_l \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \tag{1}$$

For instance, let define the following normalized low-pass FIR filter whose impulse response in the z-domain is

$$m(z) = \frac{1}{3}\left(1 + z^{-1} + z^{-2}\right) \tag{2}$$

Consequently, the associated matrix to filter a pulse of length $l = 6$ is

$$\mathbf{M} = \frac{1}{3}\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{3}$$

In the case that $\mathbf{M}$ was not a Toeplitz matrix but a generic matrix, $\mathbf{M}$ will actually be a variable-in-time convolution. It implies that the convolution coefficients (i.e. the filter) are changing for every $\mathbf{x}$ element, which is useful to improve the filtering results. In this paper, a novel procedure to obtain an efficient $\mathbf{M}$ against noise using SVD decomposition is proposed. Thi method is explained in Sections 3 and 4.

3. Singular Value Decomposition

The SVD algorithm is a non-parametric (also called *blind*) factorization of real (or complex) matrices. Additional information about SVD and its implementation can be found in [7]. In this paper, only real numbers are used. Whether we apply SVD **to** an arbitrary matrix $\mathbf{X} \in \mathbb{R}^{l \times n}$, we obtain

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top} \tag{4}$$

where, for $n < l$, $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{S} \in \mathbb{R}^{n\times n}$ and $\mathbf{V}^{\top} \in \mathbb{R}^{n\times l}$.

In this decomposition, $\mathbf{U}$ is an orthogonal matrix (i.e. whose transpose $\mathbf{U}^{\top}$ is also its inverse $\mathbf{U}^{-1}$), $\mathbf{S}$ is the diagonal matrix of eigenvectors and $\mathbf{V}$ (which is also orthogonal) contains the eigenvectors of the decomposition (that is a basis for $\mathbf{X}$).

In contrast to other blind factorization methods such as Non-Negative Matrix Factorization (NNMF) or Sparse Component Analysis (SCA), the use of SVD has the advantage that the obtained eigenvalues are ordered in relation to their significance [7] with respect to $\mathbf{X}$. These eigenvalues provide a basis of n dimensions. According to [3] the most significant dimensions considered are the signal and the lowest noise and therefore they can be removed.

This filter mitigates any type of noise including white, brownian or $1/f$ following the same procedure described above.

The variability of the pulses, both in height and shape, is learned by the filter in the same way than a neural network. In fact, the SVD method is equivalent to using an autoencoder neural network with a linear hidden unit [8]. The difference is that the main components of SVD are represented on an orthogonal basis while those of the neural network do not have to be necessary orthogonal.

4. Procedure

It is known that the value of a vector (actually a tensor) can be expressed in terms of a basis and its components and that it is independent from the chosen reference system. Keeping this in mind, the main idea of this algorithm is to calculate an alternative basis (of reduced dimensions) using training pulses and fit the incoming pulses in it using a change of basis (multiplying by $\mathbf{V}$). As a result of the fitting of the incoming pulses in the alternative basis a series of components have to be obtained. Afterwards, these components are transformed to the original basis again (by multiplying by $\mathbf{V}^{-1}$) in order to obtain the filtered incoming pulses.

For this purpose, let $\mathbf{X_t}$ be a set of n different training pulses of length equal to l clock cycles (dimensions) each, $\mathbf{X_t} \in \mathbb{R}^{n\times l}$. These pulses can be obtained from a real detector or can be generated automatically without noise (the difference in results is explained in Section 6). In order to obtain $\mathbf{V}$, when $n < l$, the value of $\mathbf{X_t}$ is factorized using SVD (4)

$$\mathbf{X_t} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top} = \mathbf{H_t}\mathbf{V}^{\top} \tag{5}$$

where $\mathbf{H_t} \in \mathbb{R}^{n\times n}$ are the components of the training pulses and $\mathbf{V} \in \mathbb{R}^{l\times n}$ are their basis. $\mathbf{H_t}$ is not used for this algorithm. If $n > l$, $\mathbf{V}$ it must be replaced by $\mathbf{U}$ and vice versa. However, in this work, it is assumed that a sufficient number of samples is taken from the pulses to filter them correctly compared to the number of training pulses and therefore $n < l$.

On the other hand, a number of k digital pulses coming from detectors of length l each are embedded into a matrix $\mathbf{X}$. These pulses contain noise that can be reduced if they are analyzed according to the previously obtained basis $\mathbf{V}$. Thus, whenever a new pulse ($k = 1$) or a set of new k pulses are detected, their components are obtained by applying $\mathbf{H} = \mathbf{US}$ on Eq. (4). It yields

$$\mathbf{H} = \mathbf{V}^\top \mathbf{X} \tag{6}$$

where $\mathbf{H} \in \mathbb{R}^{n\times k}$ are the components of the test pulses according to the $\mathbf{V}$ basis.

In order to clear up the pulse, the first s eigenvectors of the diagonal matrix $\mathbf{S}$ are chosen and the others are set to zero. Since $\mathbf{US} = \mathbf{H}$, this is equivalent to keep the first s rows of $\mathbf{H}$ and set the others to zero. The result of this operation is represented by $\mathbf{H}_s \in \mathbb{R}^{n\times k}$ matrix

$$\mathbf{H}_s = \begin{pmatrix} H_{11} & H_{12} & \dots & H_{1k} \\ H_{21} & H_{22} & \dots & H_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ H_{s1} & H_{s2} & \dots & H_{sk} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \tag{7}$$

Finally, to obtain the filtered pulse $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{V}\mathbf{H}_s \tag{8}$$

where $\mathbf{Y} \in \mathbb{R}^{l\times k}$.

Logically, $\mathbf{H}_n = \mathbf{H}$ and implies that $\mathbf{Y} = \mathbf{V}\mathbf{V}^\top \mathbf{X}$ as explained in Section 5.3.

Distinguishing between training and real pulses implies a significant improvement of the results. To sum up, a scheme of the filtering performed in this section is shown in Fig. 1. Note that with this filtering scheme, both a single pulse (case $k = 1$) or a set of k pulses in parallel can be filtered. In the next section, the efficiency of this filtering approach against noise is analyzed.

5. Noise filtering analysis

The objective of this analysis is to calculate how the SVD-filtering improves the SNR compared to other FIR filters. As stated in Section 2, these FIR filters are modeled as Toeplitz matrices. The noise filtering study is carried out in time-domain in a similar way to [9–11] for analog shaping. In these works, the assumption that an average number of delta functions (white noise) and step functions (brownian noise) are

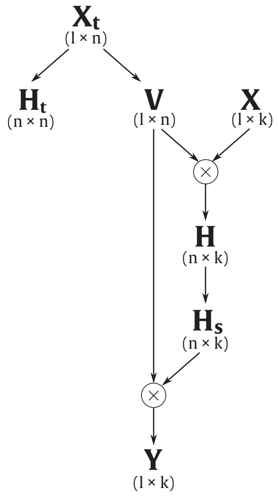

Figure 1: Diagram of the SVD-filtering.

produced in the input circuit by the noise sources is made. Both indices are inversely proportional to the SNR. In accordance with [11], white and brownian noise are represented by F_v and F_i respectively and they are equal to:

$$F_i = \frac{1}{S^2} \int_0^\infty W^2(t)\, dt \tag{9}$$

$$F_v = \frac{1}{S^2} \int_0^\infty \left(\frac{dW}{dt}\right)^2 dt \tag{10}$$

where S is the signal amplitude (in case of filters, it is assumed that $S = 1$ because it is not need it to amplify the signal, just to filter it). $W(t)$ represents the residual effect of a single unit noise element. $W(t)$ can be determined analytically because the filter is known. For time-invariant pulse shaping, $W(t)$ is the system's impulse response for a short input pulse with the peak output signal normalized to unity. For time-variant systems (e.g. gated integrators), $W(t)$ can be also calculated with the method described in [9]. The impact of more general noise types such as those outlined in [12] are beyond of the scope of this paper.

Eq. (9, 10) are applicable to both analog and digital shapers taking into account that integrals and derivatives must be changed to summations and subtractions in the digital shapers also the residual effect must be discretized

$$F_i = \frac{1}{S^2} \sum_{n=0}^{\infty} (W[n])^2 \Delta t \tag{11}$$

$$F_v = \frac{1}{S^2} \sum_{n=0}^{\infty} \left(\frac{W[n] - W[n-1]}{\Delta t}\right)^2 \Delta t \tag{12}$$

where Δt is the clock period of the discrete filter.

In the same way that the cited works and in order to quantify how the SVD-filtering improves the SNR compared to more traditional FIR filters, these two last formulas were adapted to replace $W[n]$ by $\mathbf{M}$.

5.1. Response to white noise

As it is pointed out by [9], a unit of white noise (also called delta noise in the cited reference) can be modeled as a discrete Dirac delta function. In discrete signal processing, this function can be represented by $\delta = (1\ 0\ 0\ \ldots)^{\top}$. However, when an unit of white noise modifies a pulse, it can happen along all the pulse duration. For this reason, let us also define δ_i as δ delayed i cycles, where i can take any discrete value from 0 to the length of the pulse l. The impact of white noise in the pulse measurement is the mean value of all of these probabilities. Therefore

$$F_v = \frac{1}{l}\sum_{i=0}^{l}\sum_{j=0}^{k}\left(M_{ij}\ \delta_i\right)^2 \tag{13}$$

where M_{ij} are the entries of $\mathbf{M}$.

This equation can be simplified as

$$F_v = \frac{1}{l}\sum_{i=0}^{l}\sum_{j=0}^{k}\left(M_{ij}\right)^2 \tag{14}$$

Lets compare this last equation with (12) using the FIR filter defined in (2). Using (12) and normalizing $S = 1$ and $\Delta t = 1$ the white noise index is obtained as

$$F_v = 3\left(\frac{1}{3}\right)^2 = \frac{1}{3}$$

On the other hand, using Eq. (14) with the matrix (2) associated to the example filter for input signals of length $l = 6$, the same result is obtained:

$$F_v = \frac{1}{6}\sum_{i=0}^{5}\sum_{j=0}^{5}\begin{pmatrix} \frac{1}{3^2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3^2} & \frac{1}{3^2} & 0 & 0 & 0 & 0 \\ \frac{1}{3^2} & \frac{1}{3^2} & \frac{1}{3^2} & 0 & 0 & 0 \\ 0 & \frac{1}{3^2} & \frac{1}{3^2} & \frac{1}{3^2} & 0 & 0 \\ 0 & 0 & \frac{1}{3^2} & \frac{1}{3^2} & \frac{1}{3^2} & 0 \\ 0 & 0 & 0 & \frac{1}{3^2} & \frac{1}{3^2} & \frac{1}{3^2} \end{pmatrix} = \frac{1}{3}$$

5.2. Response to brownian noise

As pointed out by [9], an unit of brownian noise (also called step noise in the cited reference) can be modeled as a step signal.

Thus, this function can be modeled simply as $\mathbf{u} = (1\ 1\ 1\ 1\ \ldots)^{\top}$. Thereby, to calculate F_i, we have to divide the effect of filters on brownian noise by l in the same way than F_v

$$F_i = \sum_{i=0}^{l} \left(\sum_{j=0}^{k} M_{ij} \right)^2 \tag{15}$$

Note that, unlike white noise and assuming that the number of events multiplied by the length of the pulse is small compared to the total past time, we can consider that all brownian noise pulse turned up before the pulse is captured. For this reason, delaying $\mathbf{u}$ has no sense because it gives the same result and therefore no mean is worked out for this equation.

Following the previous example, lets compare this last equation with (11) using the FIR filter defined in (2). If we feed this filter with $\mathbf{u}$, the output pulse is $(\frac{1}{2}\ \frac{2}{3}\ 1\ 1\ 1\ 1)$. Using (11) and normalizing $S = 1$ and $\Delta t = 1$. The brownian noise index is obtained

$$F_i = \frac{1}{2^2} + \frac{2}{2^2} + 1 + 1 + 1 + 1 = 4.556$$

On the other hand, using Eq. (15) with the matrix (2) associated to the example filter for input signals of length $l = 6$, we obtain the same result

$$F_i = \sum_{i=0}^{l} \left(\frac{1}{3} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \right)^2 = 4.556$$

Using (14, 15) it is observed that, as explained in [9], the effect of white noise is inversely proportional to the length of the pulse l whereas brownian noise is proportional to l.

Unfortunately, when $s < n$ the filtering operation is not linear, there is not a matrix $\mathbf{M}$ from which to calculate F_i nor F_v but they have to be calculated in three steps (6, 7, 8). Next section, is focused on the analysis of the case when $s = n$ (that is $\mathbf{H}_s = \mathbf{H}_n$, which is the only case where the filtering operation is linear, to calculate the noise indexes (see Section 6.1) and then calculate the effect on noise of varying s.

5.3. Particular case for $\mathbf{H} = \mathbf{H}_n$

As stated in Section 4, when all the eigenvalues are taken to filter the signal, that is $\mathbf{H} = \mathbf{H}_n$, Eq. (8) is rearranged as

$$\mathbf{Y} = \mathbf{V}\mathbf{V}^{\top}\mathbf{X} \tag{16}$$

The key to analyze the impact of noise is to find out what represents $\mathbf{V}\mathbf{V}^{\top}$. It is known that $\mathbf{V}$ is an orthogonal matrix i.e. $\mathbf{V}^{\top} = \mathbf{V}^{-1}$ but it is also known that $\mathbf{V}$ is not (always) symmetric. Therefore, it can be concluded that $\mathbf{V}^{\top}$ is the Moore-Penrose pseudoinverse [13, 14] of the orthogonal matrix $\mathbf{V}$.

Pseudoinverse matrices have specific properties depending of their dimension. Let define $\mathbf{I_L} = \mathbf{V}^\top\mathbf{V}$ and $\mathbf{I_R} = \mathbf{V}\mathbf{V}^\top$ where, as stated in (5), $\mathbf{V} \in \mathbb{R}^{l\times n}$. Then,

$$\mathbf{I_L} = \mathbf{I} \Leftrightarrow n < l \tag{17}$$

$$\mathbf{I_R} = \mathbf{I} \Leftrightarrow n > l \tag{18}$$

$$\mathbf{I_L} = \mathbf{I_R} = \mathbf{I} \Leftrightarrow n = l \tag{19}$$

where $\mathbf{I}$ is the identity matrix.

Then, according to Eq. (16), when $n > l$, $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$ and the system does not filter at all. This is also applicable when $s < n$, i.e. following the steps (6, 7, 8).

From all of this it can be concluded that SVD-filtering only works when $n > l$ because, otherwise, in Eq. (4), $\mathbf{U} \in \mathbb{R}^{n\times l}$, $\mathbf{S} \in \mathbb{R}^{l\times l}$ and $\mathbf{V}^\top \in \mathbb{R}^{l\times l}$. Therefore, $\mathbf{V}$ is orthogonal and symmetric and $\mathbf{V}\mathbf{V}^\top = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$. In conclusion, the constraint $n < l$ is mandatory for the filter to work.

A property of $\mathbf{I_R}$ and $\mathbf{I_L}$ is that $\mathbf{I_R}^2 = \mathbf{I_R}$ and $\mathbf{I_L}^2 = \mathbf{I_L}$ hence the same output is obtained whether we apply the filter once or several times in cascade. This is also applicable when we filter $\mathbf{X}$ is filtered when $s < n$.

In the case of white noise, when $\mathbf{H} = \mathbf{H}_s$, $\mathbf{M}$ value of Eq. (14) is replaced by $\mathbf{V}\mathbf{V}^\top$. Elaborating this equation, it yields

$$F_v = \frac{1}{l}\sum_{\text{entries}} \mathbf{V}^\top\mathbf{V} \tag{20}$$

where *entries* are every entry of the matrix $\mathbf{V}^\top\mathbf{V}$.

According to (17), when $n < l$, which is mandatory as stated before, $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$, this identity matrix has dimensions $n \times n$. Therefore,

$$F_v = \frac{n}{l} \tag{21}$$

In case of brownian noise, when $\mathbf{H} = \mathbf{H}_n$, Eq. (15) leads up to

$$F_i = \frac{1}{l}\sum_{\text{entries}} \mathbf{V}\mathbf{V}^\top \tag{22}$$

In this case, the product $\mathbf{V}\mathbf{V}^\top \in \mathbb{R}^{l\times l}$.

With these two indexes it can be concluded that, since $n < l$, F_i is greater than F_v. In addition, in the same way that common filters and shapers [9, 12], the effect of white noise is inversely proportional to the length of the pulse l whereas brownian noise is proportional to l. In the next section, these assumptions are verified.

6. Results

A set of simulations and tests to check the SVD-filtering in a real environment has been performed with the aim of checking that it works efficiently.

The SVD-filter is designed to work on adquisition chains at the output of the preamplifier or at the output of the shaping stage because at this point there is still noise left: in Section 6.1, it was placed at the output of the shaping stage to highlight the shape of the shaper, especially how flat is the output of the trapezoidal shaper (although its height varies). This flatness can help to the pulse height analyzer to measure the height of the pulse. In Section 6.2, it was placed at the output of the preamplifier, which is a more realistic scenario.

6.1. Results with simulated pulses and noise

In this test, a set of triangular, trapezoidal and cusp-like pulses of random heights and without noise were created ($\mathbf{X}^*$). Then, white, brownian and $1/f$ noises were added to these pulses to yield $\mathbf{X}$. These pulses were filtered using the SVD-filtering method to verify how the SNR was improved. The detection chain scheme was detector→shaper→filter, consequently $\mathbf{X}$ are the pulses at the output of the shaper. Despite that the brownian noise is generated mainly at the detector, we suppose that both noise types are generated before the filtering for comparison purposes.

Two examples of filtering using SVD-filtering are shown in Figure 2 and 3. The length of all the pulses of these two figures are $l = 100$ and the number of learning pulses were $n = 3$. For all cases, $\mathbf{H}$ was transformed to $\mathbf{H}_1$. However, for this concrete experiment, similar results were obtained with both $\mathbf{H}$ and $\mathbf{H}_1$.

The noise indexes F_v and F_i defined on (21) and (22), respectively are listed on Table 1. In this Table, the noise indexes for a generic low-pass filter were also added for comparison purposes. The transfer function of this filter in the z-domain is

$$h(z) = \frac{1}{5}\left(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}\right) \tag{23}$$

For this filter, the filtering matrix is not an orthogonal matrix but a Toeplitz matrix, as explained in Section 2. Therefore, to calculate the noise indexes, Eq. (14, 15) had to be used instead of (21, 22).

Table 1: Noise indexes value for the tests shapers and for a generic FIR low-pass filter for s = k.

Shaper	F_v	F_i
Triangular	0.030	0.748
Trapezoidal	0.030	0.840
Cusp-like	0.030	0.694
Low-pass	0.196	0.972

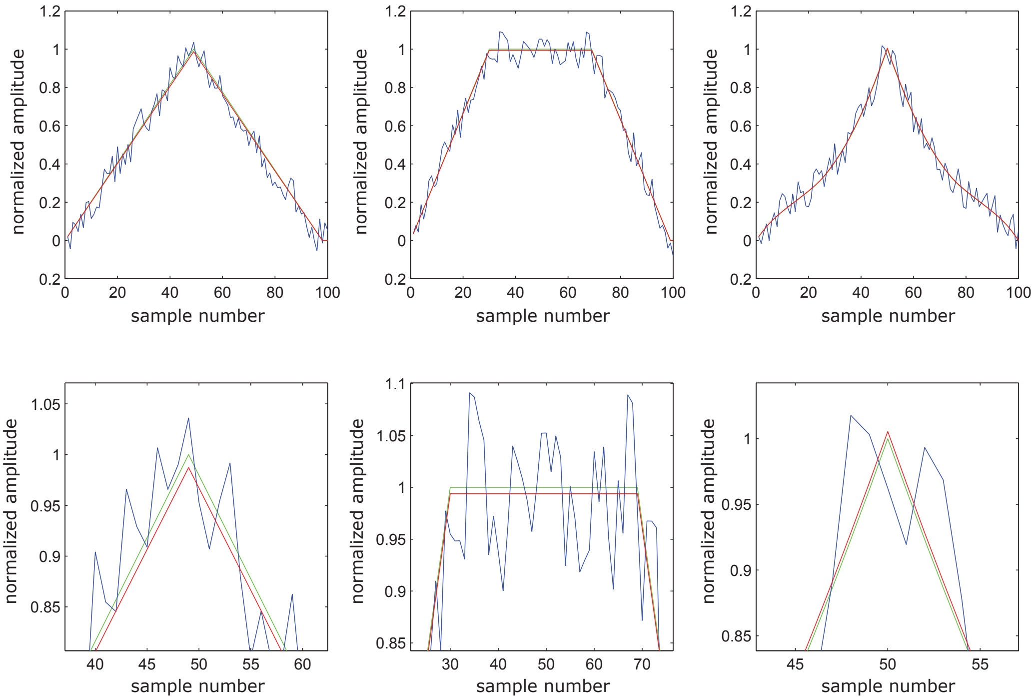

Figure 2: Example of white noise filtering for a triangular, trapezoidal and cusp-like pulse. The green pulse is the original one without noise $\mathbf{X}^*$ whose amplitude is equal to 1, the blue pulse is the input with noise $\mathbf{X}$ and the red pulse is the filtered one $\mathbf{Y}$. The panels below show a detailed view of the peak of the above panels.

When $s < n$, F_v remains constant but F_i changes. These changes are illustrated in Figure 4. It can seen that F_i oscillates around the values given in Table 1 because F_i is proportional to the squared area of the pulse.

It can observed that F_v is lowered to a constant value n/l, which is independent of the pulse shape, as predicted in (21), whereas F_i indicates that the brownian noise is harder to filter using this method. The low-pass filter gives worse results in both indexes than the other filters.

These two indexes support the assessment that the SVD-filtering filters noise in a more efficient way than FIR filters. However, when the pulses from the particle detector are processed using Pulse Height Analysis (PHA), a more realistic way to obtain the system resolution is to measure the relative error Δ of each pulse $\mathbf{x}$ contained in $\mathbf{X}$, defined as

$$\Delta = \text{mean}\left(\frac{|\max(\mathbf{x}) - \max(\mathbf{x}^*)|}{\max(\mathbf{x}^*)}\right) \tag{24}$$

where *mean* stands for the mean value of every pulse contained in $\mathbf{X}$. In the case of trapezoidal shaping, the maximum value of the pulse is the mean value of its plateau. To calculate this mean value, the number of filtered pulses were $k = 200$. Using this method instead of the one explained in Section 5 we obtain similar

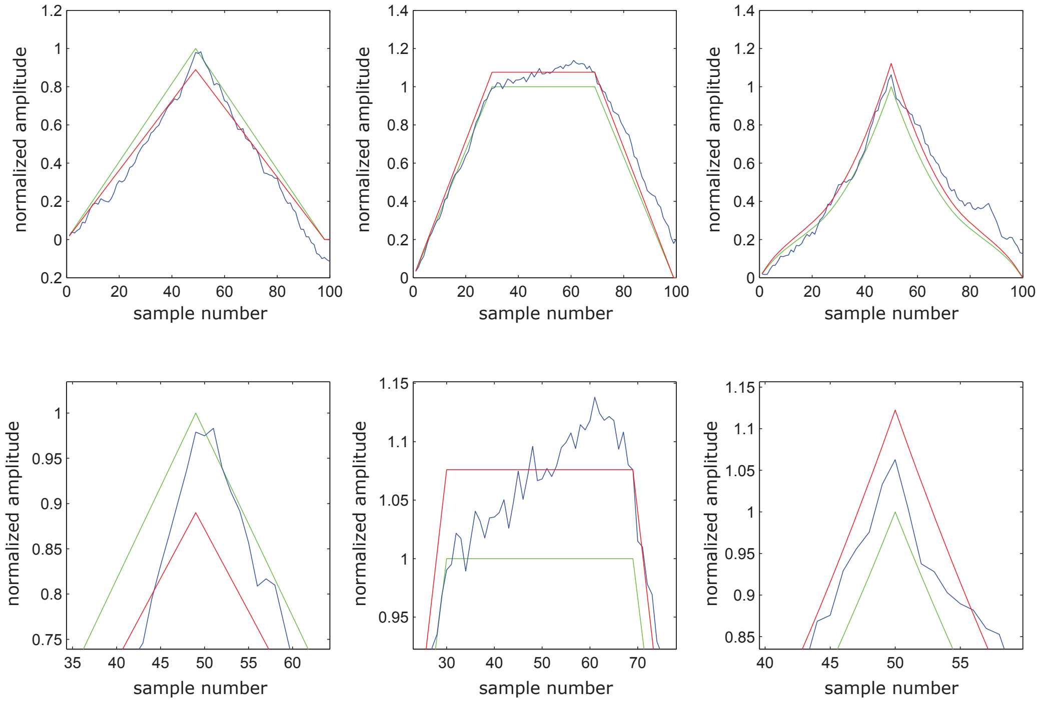

Figure 3: Example of brownian noise filtering for a triangular, trapezoidal and cusp-like pulse. The green pulse is the original one without noise $\mathbf{X}^*$ whose amplitude is equal to 1, the blue pulse is the input with noise $\mathbf{X}$ and the red pulse is the filtered one $\mathbf{Y}$. The panels below show a detailed view of the peak of the above panels.

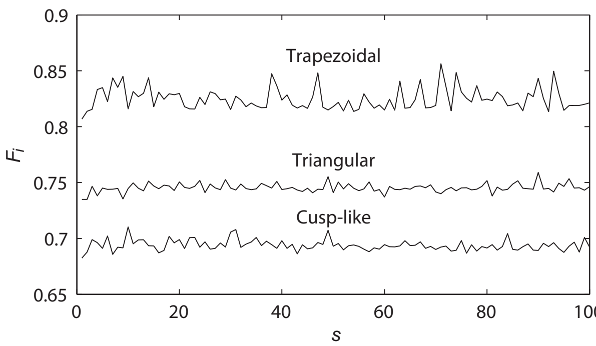

Figure 4: F_i vs s for trapezoidal (upper line), triangular, and cusp-like (lower line) for $l = 100$.

results. They are shown in Figure 5, 6 and 7 for white, $1/f$ and brownian noise, respectively. The straight lines indicate pulses with $l = 100$ whereas the dotted lines indicate pulses with $l = 10$. The $\mathbf{V}$ matrices were calculated using $n = 1$, $n = 10$ and $n = 100$ training pulses (in red, green and blue, respectively). These figures also includes the low-pass filtering (depicted in black for comparison) defined in Eq. (23).

The blue ($n = 100$) dotted ($l = 10$) line does not appear in these figures because n is greater than l and, according to $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$ it is not possible. The black dotted line is not depicted in any figure because the filter of (23) lowers the height of the pulses with length ($l = 10$). In fact, the error value defined in (24) is approximately one order of magnitude higher than the others.

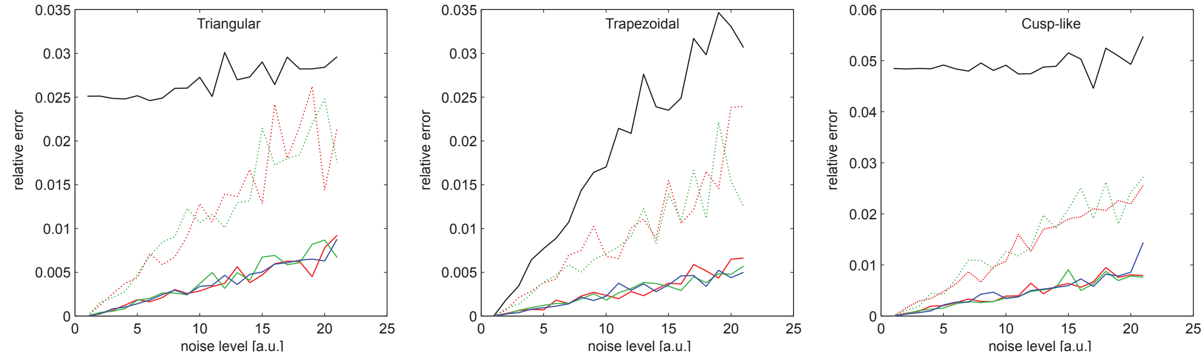

Figure 5: Level of white noise (in arbitrary units) vs. relative error (in percentage) for triangular, trapezoidal and cusp-like pulses.

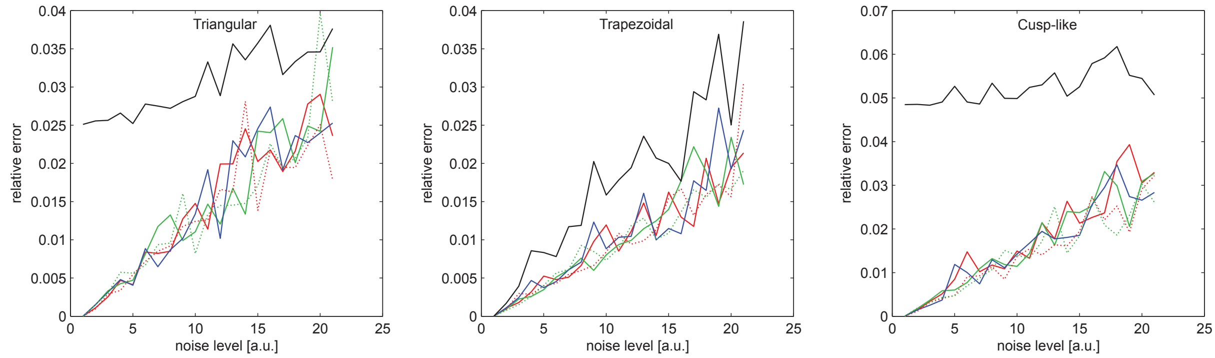

Figure 6: Level of $1/f$ noise (in arbitrary units) vs. relative error (in percentage) for triangular, trapezoidal and cusp-like pulses.

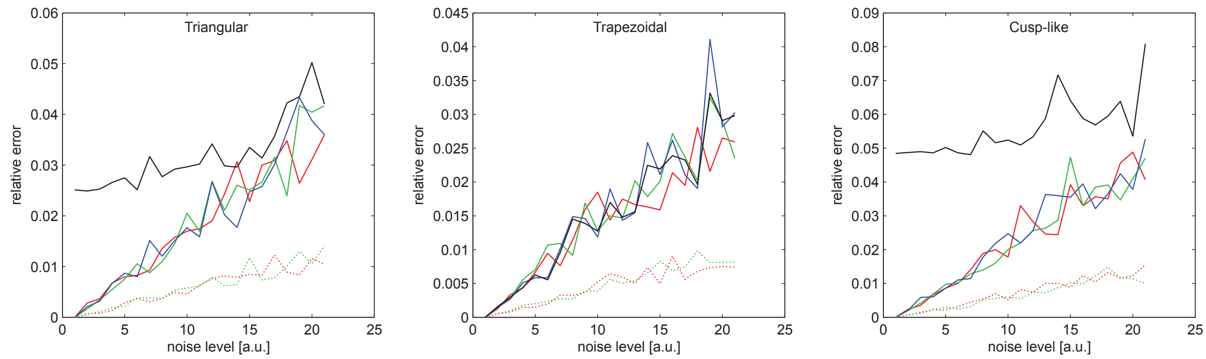

Figure 7: Level of brownian noise (in arbitrary units) vs. relative error (in percentage) for triangular, trapezoidal and cusp-like pulses.

These figures confirm that the pulse length is inversely proportional to the white noise effect, proportional to the brownian noise effect and independent from the $1/f$ noise effect, in accordance with Section 5 and [9, 11, 12]. Likewise, it can be seen that brownian noise has more impact on the SVD-filtering than white noise, which were predicted by Eq. (21, 22).

It can be seen that this method filters the noise efficiently. In fact, when a high noise level is present, the key issue is to establish a correct threshold algorithm such as the one presented in [15, 16].

6.2. Results with pulses from a neutron monitor

Finally, a test to check the proposed filtering was performed. The main objective of this test was to obtain similar results to those obtained in the experiments done without SVD filtering.

This test was performed in the Castilla-La Mancha Neutron Monitor (CaLMa) located in Guadalajara, Spain. This instrument consists of 15 proportional gas counter tubes. More information about features, setup and results of this instrument can be found in [17]. In both the cited experiment and the present test, an LND206 tube connected to a Canberra ACHNA98 preamplifier was used.

The raw data fed out from the preamplifier was digitized using a Data Acquisition system (DAQ) at sampling period of $T_s = 20$ ns and storing it in a PC. Pulses stored in a text file can be used multiple times without recapturing new data. In addition, it ensures that possible changes in the obtained results during the test are exclusively due to digital pulse processing. The total raw data length was of 46105 pulses $\times$ 1002 samples per pulse (i.e. $l = 1002$) captured during over 5 hours (from 10:20 UTC to 15:23 UTC on November, 16th 2018). To separate the input pulses, a trigger threshold of 1 V without any previous digital filtering was used.

A difference with the previous test is that the training pulses are obtained from the same source than the test pulses. Thus, a conclusion obtained from this is that the more noise the training signal has, the more pulses are required to reduce the noise, the larger is $\mathbf{V}$, and therefore more computing time is necessary when processing a pulse. For this reason, a trade-off between learning pulses and computation time was carried out and finally $n = 100$ and $s = 10$ were used. In Figure 9, the output of one pulse processed with the SVD-filter compared to raw pulses and a low-pass FIR filter are shown. Note from this figure that the low-pass filtering can remove the high frequency components which prevents the peak from reaching its maximum. Therefore, altering the histogram.

In Figure 9, the output of six pulses processed with the same SVD-filter are shown. We can see that, regardless of the shape of the pulses, the pulses are correctly filtered.

Since Eq. (24) cannot be used to evaluate the obtained results, because the value of $\mathbf{X}^*$ is not known, the filter quality was measured using the Full Width at Half Maximum (FWHM) which is defined as the width of the distribution at a level that is just half the maximum value of the peak divided by the location of the peak maximum [1].

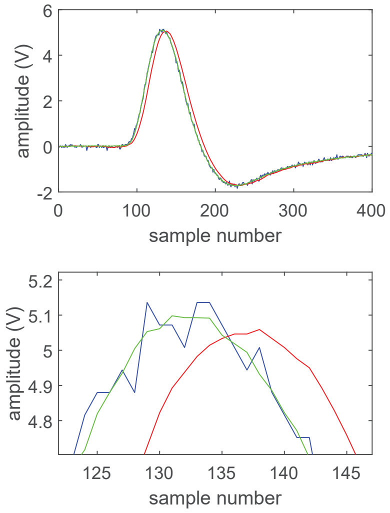

Figure 8: Example of pulse filtering. The blue pulse is the original unfiltered, the red pulse is the same pulse filtered with a low-pass filter such as (23) with order 10, the green pulse is the pulse filtered with the SVD method setting $s = 10$. The bottom panel shows a detailed view of the one above.

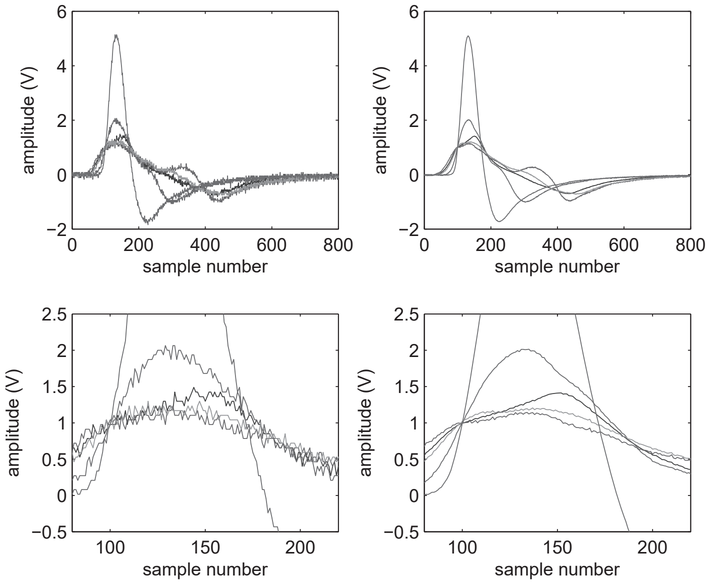

Figure 9: Example of pulse filtering on the neutron monitor. Left: Pulses from the neutron monitor without filtering. Right: Pulses filtered with the SVD-filter.

In Figure 10, an example of histogram is shown. To evaluate the results, in all the tests the FWHM of the pulse height histograms were similar for three cases: (a) pulses unfiltered; (b) pulses filtered with a low-pass filter whose response function in z-domain is (23); (c) filtered pulse using SVD-filtering. The obtained FWHM was 0.0370, 0.0340 and 0.0375, respectively. We can observe that the FWHM obtained with SVD-filtering is slightly lower than that of unprocessed pulses. On the contrary, as it was advanced, the low-pass filter reduces the height of the pulse, lowering its FWHM.

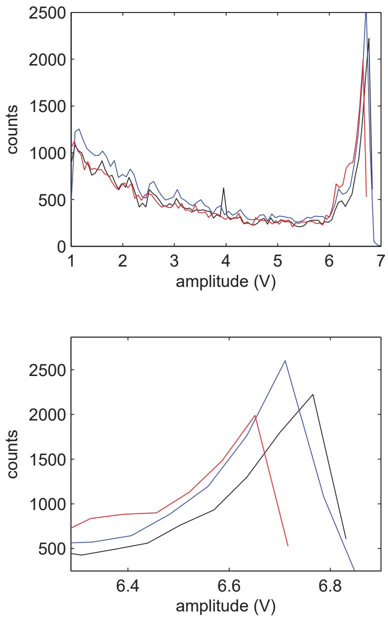

Figure 10: Histogram obtained using different filtering methods. The blue pulse is the original one unfiltered, the red pulse is the same pulse filtered with a low-pass filter of order 10, the green pulse is the pulse filtered with the SVD method setting $s = 800$. The panel below show a detailed view of the peaks.

Additional experiments showed that as s decreases, the filtered pulses (Fig. 9) are smoother. However, a decrease in the FWHM value begins to be noticed. Thus, it can be concluded that a good filtering not always implies a significant reduction of the FWHM. Moreover, for this concrete experiment, oscillations on the FWHM were detected for small variations of either s and l.

To finish, it has been seen from the observations that this method does not work properly with pile-up pulses. This fact has been addressed also in previous works such as [3].

7. Conclusions and future work

A novel filtering technique has been presented in this article. This technique is a time-variant convolution calculated using matrices. These matrices are obtained from SVD and provides a filtering quality that often improves that of traditional filters. This method filters noise as traditional filters do, that is, the length of the pulse is proportional to the brownian noise and inversely proportional to the white noise. Despite being less susceptible to white noise, it is more susceptible to brownian noise. Simulation and tests with pulses coming from a neutron monitor were performed to evaluate its performance. The constraint of this method is that the number of pulses to calculate the **V** used to filter pulses must be lower that the length in cycles of a single pulse. Besides, when the pulses used to calculate **V** are noisy, additional training pulses are needed, increasing the computation time. Noise indexes measurement supports the assessment that the SVD-filtering filters noise in more an efficient way than FIR filters. Furthermore, practical tests demonstrate that lowering the value of s (number of eigenvalues) improves the FWHM. With all this, this method is relatively easy to implement (just matrix multiplication and a threshold mechanism are necessary) and therefore suitable for particle detection spectroscopy.

As future work, alternative linear and non-linear techniques will be implemented to substitute the SVD factorization method used to find out the basis.

Acknowledgments

Thiw work has been supported by the Spanish Administration as part of the project with Ref. ESP2017-88436-R. The authors thank the CaLMa Team for technical support and for providing the CaLMa facility.

Bibliography

References

[1] M. Nakhostin, Signal Processing for Radiation Detectors, Wiley (2018).

[2] V. T. Jordanov, K. V. Jordanova, IEEE International Conference on Advancements in Nuclear Instrumentation Measurement Methods and their Applications (2011) 1.

[3] D. Yan et al., Journal of Low Temperature Physics 184 (2016) 397.

[4] A. Hčker, V. Kartvelishvili, Nuclear Instruments and Methods in Physics Research A 372 (1996) 469.

[5] I. Sandberg, et al., IEEE Transaction on Nuclear Science 59 (2012) 1105.

[6] V. Maget, et al., IEEE Transaction on Nuclear Science 61 (2012) 1687.

[7] G. H. Golub, C. F. Van Loan, Matrix Computations 4th ed, The John Hopkins University Press (2013).

[8] G. D. Clifford, HST582J/6.555J/16.456J Biomedical Signal and Image Processing (2005) 43.

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

[10] F. S. Goulding, Nuclear Instruments and Methods 485 (2002) 653.

[11] C. Patrignani et al. (Particle Data Group), Chin. Phys. C, 40, 100001 (2016) 477.

[12] A. Regadío, J. Tabero, S. Sánchez-Prieto, Nuclear Instruments and Methods in Physics Research A 811 (2016) 25.

[13] E. H. Moore, Bulletin of the American Mathematical Society 26.9 (1920) 394.

[14] R. Penrose, Proceedings of the Cambridge Philosophical Society 51 (1955) 406.

[15] V. T. Jordanov, D. L. Hall, M. Kastner, IEEE Nuclear Science Symposium Conference 1 (2002) 140.

[16] H. Kim, et al., Nuclear Instruments and Methods in Physics Research A 602 (2009) 618.

[17] J. Medina, et al., Nucl. Instrum. Methods Phys. Res. A 727 (2013) 97.