Nuclear Instruments and Methods in Physics Research A 736 (2014) 88–98

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

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

Comparative analysis of digital pulse processing methods at high count rates

J. Kamleitner*, S. Coda, S. Gnesin, Ph. Marmillod

École Polytechnique Fédérale de Lausanne (EPFL) – Centre de Recherches en Physique des Plasmas, Association EURATOM – Confédération Suisse, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 12 June 2013
Received in revised form
7 October 2013
Accepted 10 October 2013
Available online 19 October 2013

Keywords:
Digital
Pulse detection
Signal processing
Spectrometer
CdTe

ABSTRACT

An extensive study of digital pulse processing methods is presented. Existing methods, both traditional and more recent, are compared with original advanced techniques within an appropriate modeling and benchmarking framework. This comprehensive approach ensures general applicability to the broad field of pulse processing, even though the focus lies on hard X-ray spectrometers operated at high count rates. In this regime, pile-up is the main issue and the individual pulse shape characteristics play a minor role, although they remain important for the algorithm parameter optimization.

The digital implementation of double-differentiating analog filters and trapezoidal FIR filter methods results in excellent performance that is second only to that of optimum digital FIR filters. Several more complex methods involving increased computational effort are found not to meet the expectations.

1. Introduction

A novel hard X-ray tomographic spectrometer (HXRS) apparatus [1,2] is being developed for TCV, an experimental nuclear fusion device of the tokamak family [3]. It uses $(2\ \mathrm{mm})^3$ cube CdTe detectors to measure photons with energies of 10 keV to over 300 keV. The detectors are followed by fast integrators that act as charge collectors, the charge being proportional to the impinging photon energy. The pulses have a characteristic rise time of 400 ns, and the integrators are discharged with a decay time of 4 μs. The collected signal is amplified to the digitizer range and acquired at 12M samples/s.

The detection and analysis of the single photons from these time traces is a challenging task at the desired high count rates of several 100 kcps (counts-per-second) and in view of the significant noise level of a tokamak experiment environment.

To find the best-suited algorithm for the specific HXRS requirements, several standard and advanced pulse processing techniques have been implemented. The accompanying algorithm benchmarking suite was deliberately designed with broad flexibility, permitting a general study of the pulse detection and analysis problem at high count rates that transcends the specific requirements of the HXRS system. The results of this study are reported in this paper.

1.1. Analog versus digital pulse processing

Practically every particle counting spectrometer consists of an analog and a digital part with an analog to digital converter (ADC) in between. In analog pulse processing the digital part only performs a histogram. This was the only solution available till the early 1990s, when digital systems became fast enough to restrict the analog part to the charge collection and preamplification. In the latter scheme, the preamplified signal is directly sampled by a high-resolution ADC that records the time history of each individual pulse, and the pulse processing is performed digitally, either by hardware or by software [4]. In the transitional period, when the performance of digital processors was still relatively low, hybrid systems were also used, in which a digital signal processor (DSP) was triggered by an analog pulse height analyzer (PHA) [5].

The main advantages of analog pulse processing lie in robustness, decades of experience and low cost as compared to digital systems. With some additional effort even the pulse shape can be used to a certain extent to aid the pulse recognition, for instance to discriminate between particle types [6].

Nowadays digital pulse processing is used in commercially available spectrometers [7] as well as in highly specialized applications such as spacecraft [8] and magnetic confinement fusion experiments [1]. The available digital solutions continuously decrease in cost and increase in processing speed and storage, enabling real-time applications as well as storage of the entire time traces acquired. Since the shape of each pulse is

* Corresponding author. Tel.: +41 216935129.
E-mail address: josef.kamleitner@alumni.tuwien.ac.at (J. Kamleitner).

0168-9002/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.nima.2013.10.023

known, it can be used to extract more information than just the height of the pulse. Therefore the pulse processing algorithms can be optimized with respect to the pulse characteristics, for instance for neutron - gamma discrimination [9] or to detect the incident position of the particle within a detector [10] and use this information to improve the determination of the particle energy [11].

Furthermore, the time and energy of each detected pulse are available. This allows one to freely choose time and energy bins after the measurement and subsequently to use advanced analysis techniques such as conditional averaging: this is very effective in particular in tokamak diagnostic applications to analyze randomly repeated events such as sawtooth crashes [2].

1.2. Existing techniques

In analog pulse processing the preamplifier signal is shaped by a shaping amplifier. This can be realized either as a single or a double delay line (SDL, DDL) or as a combination of m differentiating (CR) and n integrating (RC) circuits ((CR)^m(RC)^n filter). In addition, tail (pole-zero) cancellation and baseline restoration are regularly applied to improve the signal properties further. Finally the signal is evaluated using a peak-sensing ADC and the detected pulses are digitally stored in a histogram [12].

In digital pulse processing the same analog techniques can be implemented digitally [13]. However, a large variety of additional methods can also be used, comprising more sophisticated techniques that take the whole time history and pulse shapes into account. Typical such techniques are digital finite and infinite impulse response (FIR and IIR) filters [14,15], cross-correlation [16,17] and least-squares difference [18] to template pulses, neural network pulse recognition [19], wavelet transform [20,21] and support vector machine (SVM) pulse sorting [22].

1.3. Outline

The implementation framework of the pulse processing algorithms will be briefly discussed in Section 2. The algorithms are then described in detail in Section 3. The specification of the benchmarking methods in Section 4 is followed by the discussion of the algorithms' performance in Section 5. Finally, conclusions will be drawn in Section 6.

2. Digital implementation

Although a very wide range of different pulse processing algorithms is compared, a large fraction of these algorithms shares the use of a few fundamental steps in the data analysis. This allows not only the implementation of a general framework for data input/output (I/O), storage and benchmarking, but also a generalization of the algorithm implementation itself. This in turn makes it possible to study more algorithms with little additional effort and facilitates their comparison. The common basic steps are, namely, signal treatment, pulse detection (PD) and pulse height analysis (PHA). These components are individually presented in the following. Nonetheless, the implementation can be kept flexible enough to treat algorithms that can be only partially or not at all resolved by this sequence of steps or that require additional post-processing.

For all presented algorithms, the signal treatment, PD and PHA methods are listed in Table 1.

2.1. Signal treatment

The signal treatment processes the raw data to provide an input for the pulse detection and analysis. It keeps the signal's original sampling rate and is often realized by the application of filters. Usually the signal treatment is the same for the pulse detection and pulse height analysis, although there are also a few methods where the signal treatment for analysis differs from that for detection. Since most pulse processing algorithms share the same or similar detection and analysis methods, their main differences lie in the signal treatment. Therefore, the signal treatment parts play the main role in characterizing a pulse processing method.

Table 1
List of signal treatment, pulse detection and PHA methods for all presented algorithms.

Algorithm	Detection		Analysis	
Abbreviation	Signal treatment	Pulse detection	Signal treatment	PHA
Trpz2	Trpz2	Dynamic threshold	Trpz2	Level evaluation
Trpz1a	Trpz1a: $n_r = n_d = 4$	Dynamic threshold	Trpz1a: $n_r = n_d = 4$	Level evaluation
Trpz1as	Trpz1a: $n_r = n_d = 3$	Dynamic threshold	Trpz1a: $n_r = n_d = 3$	Level evaluation
Trpz1	Trpz1	Dynamic threshold	Trpz1	Level evaluation
Trpz	Trpz	Dynamic threshold	Trpz	Level evaluation
CC-LMS	CC	Dynamic threshold	LMS	Level evaluation
LMS	LMS	Dynamic threshold	LMS	Level evaluation
$(CR)^2(RC)$	$(CR)^2(RC)$	Dynamic threshold	$(CR)^2(RC)$	Level evaluation
$(CR)^2(RC)^4$	$(CR)^2(RC)^4$	Dynamic threshold	$(CR)^2(RC)^4$	Level evaluation
$(CR)(RC)^4$	$(CR)(RC)^4$	Dynamic threshold	$(CR)(RC)^4$	Level evaluation
$(CR)\ (RC)$	$(CR)\ (RC)$	Dynamic threshold	$(CR)\ (RC)$	Level evaluation
CIS	Digital band-pass	CIS – rise threshold	Digital band-pass	Rise evaluation
PSD	MA filter	Multiple condition	MA filter	Rise evaluation
Canny	Canny	Dynamic threshold	Canny	Level evaluation
SDL	SDL	Dynamic threshold	SDL	Level evaluation
DDL	DDL	Dynamic threshold	DDL	Level evaluation
opt1na	Optimum filter 1	Dynamic threshold	Optimum filter 1	Level evaluation
opt2na	Optimum filter 2	Dynamic threshold	Optimum filter 2	Level evaluation
opt3na	Optimum filter 3	Dynamic threshold	Optimum filter 3	Level evaluation
opt4na	Optimum filter 4	Dynamic threshold	Optimum filter 4	Level evaluation
i-500 kcps	Idealized algorithm, detects all pulses spaced by ≥ 2 μs correctly			
i- 1 Mcps	Idealized algorithm, detects all pulses spaced by ≥ 1 μs correctly			
i-2 Mcps	Idealized algorithm, detects all pulses spaced by ≥ 500 ns correctly			

2.2. Pulse detection

In the analysis chain comprising signal treatment and pulse detection, care has to be taken not to introduce a time shift in the particle arrival time. Already in the raw signal, and additionally after filtering using only data points from the past, the peak of a pulse is delayed with respect to the photon arrival time. This is why it can be necessary to apply a time index shift in addition to the basic signal treatment in order to detect the photon at its arrival time instead of the time point at which the corresponding pulse appears in the processed signal after the whole chain from charge collection to basic signal treatment.

Threshold: The threshold detection defines the detected pulses as groups of contiguous points exceeding a set threshold value, separated by points below it. The time of each pulse is then defined as the point of maximum value within it. This solution prevents artificial time shifts dependent on the pulse height, such as in the leading edge threshold crossing method, which for this reason is not considered here, in spite of its frequent usage.

Dynamic threshold: The dynamic threshold detection algorithm detects local maxima that lie above a set threshold value and identifies them as pulses if they are separated by a minimum, with the maximum to minimum ratio exceeding a set factor. Especially at high count rates this is expected to be advantageous since the signal between two consecutive pulses only has to go down to the specified ratio and not all the way below the threshold value in order to separate these two pulses.

Rise threshold: Here, the pulse rise is investigated by selecting monotonically increasing segments of the signal. Since the derivative of the signal is positive in these segments and bounded by sign changes before and after them, this selection is realized by detecting the change-in-sign (CIS). If the signal rise within such a segment exceeds a set threshold value, the start or mean time of this segment is taken as the pulse time.

Multiple conditions: Several other detection methods can be grouped under this expression, meaning that they do not look only for maxima but apply additional conditions. For instance the pulse-shape discriminator (PSD) algorithm, operating on a smoothed signal, checks for a signal rise followed by a decrease over three or more consecutive points lying within the integrator decay time [23].

2.3. Pulse height analysis

Level evaluation: To determine the pulse height the signal is simply evaluated at the points detected by the pulse detection. It should be noted that a statistical average over several points can be achieved by including this averaging in the signal treatment directly.

Rise evaluation: The rise of the signal over a certain period, usually defined by the rise threshold detection, is used to determine the pulse height.

3. Overview of signal treatment algorithms

The methods presented in this section, especially the filters, are all time-invariant. This corresponds to the application where pulse shapes and noise characteristics do not change over time. In time-variant systems, however, the method's parameters should be adjusted accordingly over time. Then the results presented in this paper can, to a certain extent, be extended to such systems as well.

In Figs. 1 and 2 the pulse responses of the most important filter methods presented in the following are compared.

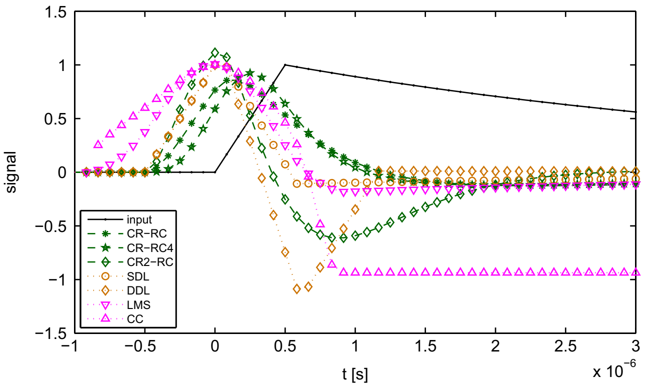

Fig. 1. Pulse response of the presented IIR, delay line, LMS and cross-correlation filters to a simulated clean input pulse.

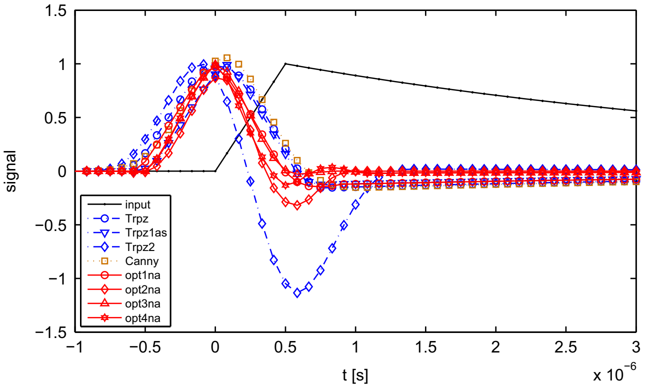

Fig. 2. Pulse response of the presented FIR filters to a simulated clean input pulse.

3.1. IIR filters

Infinite impulse response (IIR) filters are widely used in digital pulse processing, for instance as band-pass filters or to emulate analog filters digitally.

3.1.1. Digital band-pass filters

A tempting approach is the simple use of band-pass filters to remove high frequency noise and low frequency baseline deviation. In the remaining signal only the pre-amplified pulses should appear and be evaluated. However, since the pulses themselves are not sinusoidal but have a wider frequency spectrum, the filter pass band has to be chosen carefully, especially if the pulse and noise spectrum overlap significantly.

CIS method: One of the earliest attempts at digital pulse processing for the HXRS diagnostic was based on a digital band-pass filter. The filtered signal, however, cannot be used for (dynamic) threshold detection and level evaluation analysis. In order to yield a working pulse processing method the rise threshold detection and rise evaluation analysis have to be applied on the band-pass filtered signal. Subsequently, the method as a whole is called a change-in-sign (CIS) method [23].

3.1.2. Analog filters

Classic analog pulse shaping amplifiers use a combination of differentiating CR and integrating RC circuits. Their transfer functions are determined by time constants and can be easily implemented digitally, providing a reference for digital pulse processing. The principle can be understood by considering the simplest type (*CR*)(*RC*) with one CR and one RC circuit. First, the CR circuit determines the slope of the pulse rise and then its output decays over its time constant. Therefore it acts as a high-pass filter. The RC circuit then integrates the CR output and acts as a low-pass filter [12].

$(CR)(RC)^n$ *method*: If the RC integrating step is repeated n times the high-frequency noise is further reduced and the resulting pulse shape approaches a Gaussian. To keep the peaking time constant, the time constants of the n RC circuits have to be $1/n$ times the time constant of a single RC circuit. This shortening of the time constants shortens the decay after the peak as well. However, adding more circuits increases the complexity of the system and the improvement is only significant for low n. Typically 2–4 RC circuits are used [13].

$(CR)^2(RC)^n$ *method*: In the limit of high count rates any pulse detection method is affected by pile-up. To reduce this effect a second differentiating CR circuit is added, which makes the output signal bipolar. This was, for instance, done for the hard X-ray camera (HXRC) that was formerly installed on TCV and used $(CR)^2(RC)$ analog pulse processing with a filter time constant of 0.3 μs [24]. As before, $n>1$ is expected to yield a slight improvement in performance.

3.2. FIR filters

Finite impulse response (FIR) filters have, in contrast to IIR filters, an impulse response of finite duration. An analog implementation of these filters is not trivial but can be realized using delay lines or surface acoustic wave (SAW) filters. The digital implementation of discrete-time FIR filters, however, is very simple. The output signal $\mathbf{r}$ is the convolution of the input signal $\mathbf{s}$ and the filter coefficients $\mathbf{c} \in \mathbb{R}^{(N+1)\times 1}$

$$r_i = \sum_{k=0}^{N} c_k s_{m+i-k} \quad \forall i = 1, \ldots, n \tag{1}$$

where $\mathbf{s}$ is zero for all indices that are out of range. The common FIR filter definition ($m=0$) is extended here by a time index shift $m \in \mathbb{Z}$, in order to detect photons at their arrival time, as described in Section 2.2. The map $f : \mathbf{s} \mapsto \mathbf{r}$ defined by Eq. (1) is linear $f \in L(\mathbb{R}^{n\times 1})$. Another important property is that the filter coefficients represent the impulse response of a Nth order discrete-time FIR filter, which is of length $N+1$ samples.

The impulse responses of the more complicated FIR filters presented in the following, corresponding to their filter coefficients, are compared in Fig. 3. Their step responses are shown in Fig. 4.

3.2.1. Delay lines

The delay line technique splits the input signal onto several paths of different lengths where each sub-signal is also multiplied by a different weighting factor. The outputs from all the paths are then added to obtain a weighted sum of delayed versions of the raw signal.

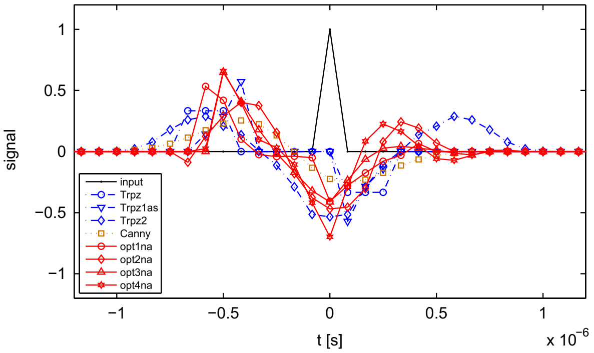

Fig. 3. Impulse response of the presented FIR filters, being equal to the filter coefficients.

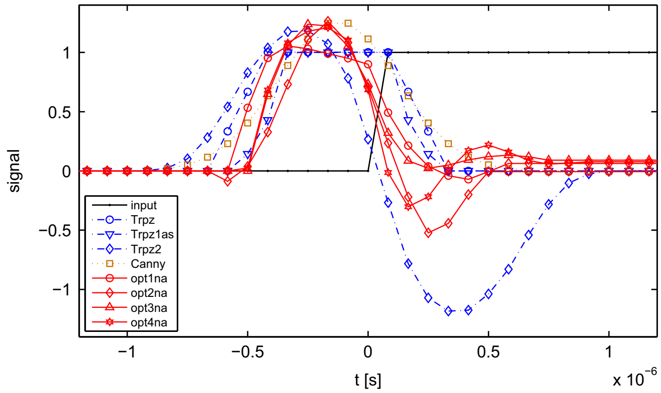

Fig. 4. Step response of the presented FIR filters.

SDL method: The single delay line (SDL) method uses a single delay line to subtract a delayed and slightly downscaled version of the signal off the signal itself. This leads to a short, nearly trapezoidal pulse with a length of the order of the delay followed by a fast reset to zero. The delay of the line should in no case be smaller than the rise time of the input pulse, otherwise parts of the pulse would already be subtracted while it still rises. To minimize the flat top of the trapezoidal-like pulse the delay should be close to the rise time of the input pulse. This results in a quasi-triangular output pulse without visible flat top, as can be seen in Fig. 1. Also, the downscaling factor has to be adjusted according to the pulse decay in order to restore the baseline after the pulse properly.

While noise at the exact delay time frequency is damped, this method includes no low-pass filter to get rid of general high-frequency noise.

DDL method: The double delay line (DDL) is a series of two SDL circuits to obtain a bipolar pulse instead of a trapezoidal peak. The resulting zero-crossing leads to a better pulse separation whereby even higher count rates can be resolved than with the SDL.

3.2.2. MA filter

A moving average (MA) filter of span S is a FIR filter with S coefficients, all being equal to $1/S$. We restrict S to odd positive integers, meaning that the order of the FIR filter $N=S-1$ is even ($N \in 2\mathbb{N}$), and we set the time shift to $m=N/2$. Following this definition, the filtered signal at each point in time is the arithmetic mean of the raw signal at that time and the $N/2$ consecutive points before and $N/2$ consecutive points after that point in time.

PSD method: The moving average filter is used to smooth the signal for the multiple condition detection in the pulse-shape discriminator (PSD) method.

3.2.3. Trapezoidal filters

Here, the trapezoidal (Trpz) filter family encompasses the basic Trpz filter and several FIR filters closely related to it.

Trpz method: The step response of the trapezoidal (Trpz) filter is a trapezoidal peak as shown in Fig. 4. The filter is specified by the number of samples in the rise (n_r), the flat top (n_f) and the decay (n_d) phases of this peak. It computes the difference between the mean value of n_r samples on the future side of a gap of n_f samples and the mean value of n_d samples on the gap's past side [12]. The response to a ramp from 0 to 1 within a certain rise time is a piecewise quadratic function. It is smoothed with respect to the step response and its flat top is shortened by the finite ramp rise time. Without a flat top this response is a quadratic spline approximation to a Gaussian.

In analogy to the SDL, the delay time, corresponding to n_f, should be greater than the rise time of input pulses. The flat top should be as short as possible to obtain Gaussian-like output pulses. However, a short flat top is required in order to account for ballistic deficit [25].

The case $n_r = n_d = 1$ corresponds to a time-discrete SDL method with n_f specifying the delay; for increasing n_r and n_d high-frequency noise is averaged out with increasing effectiveness.

Trpz1/Trpz1a method: Starting from the Trpz filter, the Trpz1 (a) filter is obtained by computing weighted mean values instead of mean values before and after the gap. If one increases the weights for samples closer to the gap, e.g., this technique can narrow the output pulse. In the Trpz1 method the weights increase linearly towards the gap while an exponential increase is used in the Trpz1a method.

Trpz2/Trpz2a method: Applying to the Trpz1(a) method the same step that was taken from the $(CR)(RC)$ to the $(CR)^2(RC)$ and from the SDL to the DDL method to the Trpz1(a) method yields the Trpz2(a) method. This subsequent application of two Trpz1(a) and an integrating window can be represented by a single FIR filter whose coefficients are a convolution of two Trpz1(a) filters and an integrating FIR filter. The latter is similar to a MA filter, but uses numerical integration method coefficients (such as $(7, 32, 12, 32, 7)/90$ from Boole's rule) instead of the arithmetic mean.

As is the case in going from SDL to DDL, this leads to a bipolar shape and is expected to increase the throughput at high count rates.

3.2.4. Canny method

The problem of detecting a pulse is similar to edge detection. The Canny edge detector [26], the first derivative of a Gaussian, can therefore be used as a FIR filter.

3.2.5. Optimum filters

The FIR filters presented above represent only a small, very specific set within the space of FIR filters. To overcome the restrictions to filters that follow strict design rules and are determined by only a few parameters, one can look for an optimum FIR filter with respect to well-defined requirements. This can be realized by linear least squares minimization of a functional containing information about the noise and the desired output pulse shape [27,28].

The basic idea is to optimize the FIR filter coefficients under several constraints in a least squares sense. The two main constraints taken into account are the following: for a given input pulse a desired output should be obtained; and for noise input the output should be zero. Furthermore, specific input frequencies can be suppressed and the pulse response area can be specified. Since all these constraints form an over-determined system of equations there is usually no exact solution. However, a best approximation in a least squares sense, where weighting factors reflect the importance of each constraint, can be found.

The two main constraints, optimum pulse response and noise rejection, can be formulated as follows: the signal $\mathbf{s}$ is written as the sum of a clean pulse signal $\mathbf{s}_c$ and a noise component $\mathbf{s}_n$. This results, according to Eq. (1), in the FIR filter output

$$r_i = \sum_{k=0}^{N} c_k s_{c,m+i-k} + c_k s_{n,m+i-k} \quad \forall i = 1, \ldots, n. \tag{2}$$

For noise of mean zero this yields an average output

$$\bar{r}_i = \sum_{k=0}^{N} c_k s_{c,m+i-k} \quad \forall i = 1, \ldots, n \tag{3}$$

with variance

$$\sigma_{\mathbf{r}}^2 = \mathbf{c}^T V \mathbf{c}. \tag{4}$$

Here, V is the auto-covariance matrix of the noise $\mathbf{s}_n$. The variance of the clean signal $\mathbf{s}_c$ that does not originate from noise, such as pulse shape changes due to different interaction locations in the detector, can be taken into account in the same manner.

The ideal result would be an average output identical to the request, and with zero variance. Generally, this cannot be obtained exactly but a best approximation can be found by minimizing the sum of the weighted difference between the average output $\bar{\mathbf{r}}$ and the desired output $\mathbf{r}_d$ and the variance $\sigma_{\mathbf{r}}^2$

$$\sum_i \alpha_{r,i} \|\bar{r}_i - r_{d,i}\|^2 + \alpha_n \sigma_{\mathbf{r}}^2 \to \min \tag{5}$$

$\boldsymbol{\alpha}_r$ being the weighting factors for the request and α_n being the noise weighting factor, all greater or equal to zero.

The expression in Eq. (5) can be written in matrix form. It adopts its minimum at the zero of its gradient, yielding the normal equation

$$\left[\left(\sum_i \alpha_{r,i} (\mathbf{s}_c \mathbf{s}_c^T)_i\right) + \alpha_n V\right] \mathbf{c} = \sum_i \alpha_{r,i} (\mathbf{s}_c)_i r_{d,i}. \tag{6}$$

Its solution is the vector of the $N+1$ optimum FIR filter coefficients $\mathbf{c}$. Additional constraints can be imposed by adding corresponding terms to Eqs. (5) and (6) respectively.

The choice of the weighting factors is essential in order to obtain a well-behaved optimum filter. In particular, a low α_n can lead to over-fitting, yielding unacceptably poor performance if the signal $\mathbf{s}$ deviates only slightly from the clean signal $\mathbf{s}_c$.

3.3. Least squares and cross-correlation methods

3.3.1. LMS method

The least squares (LMS, from least mean squares) difference from a template pulse can be used to specify a digital filter. The idea is to find the pulse amplitude a such that the template pulse $\mathbf{t}$ of height 1 is optimally scaled to the signal $\mathbf{s}$ (both column vectors of N samples) in a least squares sense

$$\|\mathbf{t}a - \mathbf{s}\|^2 \to \min \tag{7}$$

with $\mathbf{t}, \mathbf{s} \in \mathbb{R}^{N \times 1}$. The obvious choice for the template pulse $\mathbf{t}$ is the clean signal $\mathbf{s}_c$. To reduce the effect of baseline shifts both the signal and the template are shifted by their mean value before the least squares optimization is effected ($\mathbf{u}$ is a vector with all values equal to 1 and of same size as $\mathbf{s}$ and $\mathbf{t}$)

$$\mathbf{x}_0 = \mathbf{x} - \frac{\mathbf{u}^T \mathbf{x}}{N} \mathbf{u}, \quad \mathbf{x} = \mathbf{s}, \mathbf{t}. \tag{8}$$

The linear least squares problem (7) is solved by the solution of the normal equation

$$\mathbf{t}_0^T \mathbf{t}_0 a = \mathbf{t}_0^T \left(\mathbf{s} - \frac{\mathbf{u}^T \mathbf{s}}{N} \mathbf{u}\right). \tag{9}$$

From the definition of $\mathbf{u}$ it follows that $\mathbf{u}^T \mathbf{u} = N$ and, since

$$\mathbf{u}^T \mathbf{x}_0 = \mathbf{u}^T \mathbf{x} - \frac{\mathbf{u}^T \mathbf{x}}{N} \mathbf{u}^T \mathbf{u} \tag{10}$$

subsequently

$$\mathbf{u}^T \mathbf{x}_0 = 0 \tag{11}$$

for vectors $\mathbf{x}_0$ as defined in Eq. (8). On the other hand, vectors $\mathbf{y}$ with $\mathbf{u}^T \mathbf{y} = 0$ remain unchanged by Eq. (8): $\mathbf{y}_0 = \mathbf{y}$. Therefore Eqs. (8) and (11) are equivalent definitions of a zero mean LMS method template pulse $\mathbf{t}_0$.

Using Eq. (11) for $\mathbf{x} = \mathbf{t}$ yields directly $\mathbf{t}_0^T \mathbf{u} = 0$, which simplifies Eq. (9). The pulse amplitude is then given by

$$a = \frac{\mathbf{t}_0^T}{\|\mathbf{t}_0\|^2} \mathbf{s}. \tag{12}$$

Comparison to Eq. (1) reveals that Eq. (12) defines the coefficients of a FIR filter of order $N-1$:

$$\mathbf{c} := \frac{\mathbf{t}_0}{\|\mathbf{t}_0\|^2}. \tag{13}$$

Conversely, FIR filters with coefficient sum zero ($\mathbf{u}^T\mathbf{c}=0$) can be represented by the LMS method, since the template pulse defined by

$$\mathbf{t}_0 := \frac{\mathbf{c}}{\|\mathbf{c}\|^2} \tag{14}$$

fulfills Eq. (11).

3.3.2. Cross-correlation detection

Using the notation of Section 3.3.1 the normalized cross-correlation of a signal and a template pulse can be written as

$$c = \frac{\mathbf{t}_0^T\mathbf{s}_0}{\|\mathbf{t}_0\|\,\|\mathbf{s}_0\|} \in [-1,1]. \tag{15}$$

This can be directly used as a non-linear filter for pulse detection [17]. A clear advantage of this filter is that it ideally responds equally to the pulse shape, no matter what its amplitude is. However, since it does not respond to the amplitude of the pulse, a different method has to be used for the pulse height analysis. This could be any of the PHA methods discussed earlier; however, the formal similarity between the cross-correlation detection and the LMS method suggests to combine these two for an optimal result.

3.4. Wavelet transform

A discrete wavelet transform is used to detect patterns in signals that may occur on different timescales. This is for instance successfully applied in sawtooth detection using the Canny edge detector as wavelet [20] to detect sawtooth crashes whose length may vary.

In the present pulse processing application, however, the rise and decay times of the pulses to be detected are practically constant and usually well known. Therefore, pulse detection algorithms can be well-tuned to the pulse characteristics. It is not necessary to scan different timescales using a wavelet transform since the timescale over which the pulses occur is already known. Hence, it is not expected that the use of the wavelet transform would improve well-tuned pulse detection algorithms. However, if the pulse rise and decay times are not known, this method may be advantageous [21].

4. Benchmarking methods

A detailed comparison and benchmarking of the presented algorithms is carried out using experimentally measured as well as simulated signals. The analysis concentrates on simulated signals where the particle arrival time and energy are exactly known and arbitrary count rates and noise can be investigated. Nonetheless the use of experimentally measured noise in the simulation as well as the analysis of full experimental signals are necessary to ensure the validity of the signal simulation models.

4.1. Signal simulation model

The signal simulation model comprises the steps from the particles arriving at the detector via the charge collection in the pre-amplifier and the amplification up to the digitization of the signal.

4.1.1. Particles

The particle arrival times are modeled using a Poisson distribution whose free parameter is determined by the desired average count rate. The particle energy can be modeled using different spectra, e.g. monochromatic or polychromatic, exponentially distributed or uniformly distributed. The vectors of particle arrival times and energies are fed into the pre-amplifier model and kept as reference for the benchmark evaluation.

4.1.2. Detector

Since we are interested in the performance of the pulse detection algorithms, the detector model assumes simply that all modeled particles are detected at their exact incident energy. In a real system this energy may be smaller than the total energy of the particle, e.g. in the case of Compton scattering of photons. However, we do not concern ourselves in this paper with the analysis complications arising from such events.

4.1.3. Pre-amplifier

The pre-amplifier model accepts arbitrary particle arrival time and energy vectors. It is the first step in which the full time trace is modeled: the output pulses are defined by the charge collection time and the integrator decay time.

4.1.4. Amplifier

The amplifier's purpose is to amplify the pre-amplifier output to the digitizer input range without disturbing the pulse shape significantly. Since these requirements are met quite easily in practice, this stage can almost be neglected in the signal simulation model.

4.1.5. Noise

The noise sensitivity is important in the benchmarking; therefore, different forms of noise (white Gaussian, $1/f$ and experimentally measured) can be used and entered at the pre-amplifier as well as after the amplifier stage [29]. In the following analysis the experimentally measured noise is taken from HXRS data on the TCV tokamak discharge #46061, partially shown in Fig. 5.

4.2. Benchmarking figures of merit

4.2.1. Main figures of merit

Detection efficiency: The parameters reflecting the detection performance are true positive, false negative and false positive detection rates. Here, positive/negative stands for the detection result of the algorithm and true/false if this agrees with the simulation, in each case normalized to the number of simulated particles. The true positive rate (a simulated particle is detected) should obviously be as close to unity as possible, while the false

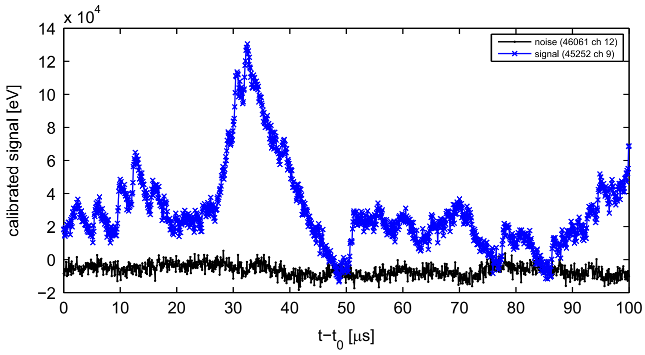

Fig. 5. Experimentally measured time traces for noise only (TCV discharge #46061, HXRS chord 12, $t_0=0.8$ s) and signal with photons arriving at the detector (TCV discharge #45252, HXRS chord 9, $t_0=0.8$ s).

negative (a simulated particle is not detected) and false positive (a particle that was not simulated is detected) rates should be as small as possible.

Energy accuracy: To obtain high energy resolution the energy deviation has to be minimal. At high count rates where significant pile-up occurs the energy deviation is expected to increase accordingly. Note that the energy under discussion here is the energy transferred by the incident particle to the detector, which is assumed to be the total particle energy, as stated already in Section 4.1.2.

4.2.2. *Additional figures of merit*

Time resolution: The time resolution of the detection is not analyzed in detail since the deviation in the particle arrival time is, for all investigated algorithms, in the order of the sampling interval. In specialized applications, such as Positron emission tomography (PET) using time of flight measurements, the time resolution plays a major role and can be enhanced by increasing the sampling rate or applying interpolation around the detection time. This works in a straightforward manner for all methods presented here.

The standard application, however, is to build spectrograms or time traces in spectral channels. Hence, the single pulses are grouped in energy and time intervals. The length of the latter can be effectively shortened by several orders of magnitude using techniques such as conditional averaging, which improve statistics by adding over different realizations of statistically similar events (e.g., periodic phenomena). Still, to get useful statistics, the time grouping interval needs to be much longer than the sampling interval length and is therefore much longer than the detection time error.

As an example, asking only for quite poor statistics (1000 pulses) from a system detecting a pulse in average every 10 samples requires more than 10^4 conditional averaging events in order to get a time resolution (grouping interval length) close to the sampling rate.

Computational performance and complexity: Since digital pulse processing inherently involves high sampling rates (typically in the M samples/s–G samples/s range), the computational performance plays an important role. Execution time and storage requirements are the main issues. Parallelization and its scaling may also play a role in certain applications but can be neglected for the HXRS system in which the number of detectors exceeds the number of available processor cores.

Real-time applicability: Directly related to the computational performance is the question whether or not the algorithms can be implemented in real-time, for instance directly in FPGAs. This is essential if the measurement is used in a control cycle, for systems with low storage capacities relative to input channels and acquisition rate and for continuously operating systems.

5. Results: algorithm benchmarks

The benchmark analysis is presented in this section.

5.1. *Analyzed algorithms*

All algorithms analyzed and presented in this study are listed in Table 1. Therein, for each algorithm abbreviation, the corresponding signal treatment algorithm for detection (as presented in Section 3), the detection method (Section 2.2), the signal treatment for analysis (Section 3) and the PHA (Section 2.3) applied are specified. All algorithms using dynamic threshold detection can also be used with ordinary threshold detection, yielding a slightly degraded performance at reduced computational effort.

5.1.1. *Optimum filter parameters*

In the benchmarking analysis four optimum FIR filters are investigated. They are all optimized with respect to the noise present in the HXRS system but the requested output differs.

The outputs requested for filters 1 and 2 are based on the clean signal responses of the Trpz1as and $(CR)^2(RC)^4$ method respectively. For the filters 3 and 4 cusp responses are requested. In the case of number 3 the cusp rises within 6 points, defined by

$$r_{\mathrm{d},i} = \frac{1}{2}\left[\frac{i}{6} + \left(\frac{i}{6}\right)^3\right], \quad i = 1, \dots, 6 \tag{16}$$

and the cusp decay is symmetric, followed by 23 zeros. For number 4 the rise is requested to take place within only 4 points and the linear contribution is replaced by a quadratic one, both in order to optimize for a very fast response. After the cusp rise defined by

$$r_{\mathrm{d},i} = \frac{i^2 + i^3}{80}, \quad i = 1, \dots, 4 \tag{17}$$

and the symmetric cusp decay, the subsequent zero is followed by a small negative dip ($r_{\mathrm{d},9} = -\frac{1}{40}$) before the final zeros. It turns out that in the case number 4 the request can never be met since it is on a smaller time scale than the pulse. However, the response is optimized to be as fast as possible in order to get good pulse separation at high count rates.

5.1.2. *Idealized algorithms*

To provide an absolute comparison of the detection efficiency, idealized algorithms are used. These are not real algorithms; rather, they use the particle simulation without processing. They are characterized by a specific detection frequency and detect all pulses that are at least separated by the timescale corresponding to this frequency. Here, the idealized algorithms i-500 kcps, i-1 Mcps and i-2 Mcps are used for the performance comparison.

5.2. *Detection efficiency*

5.2.1. *Count rates*

All presented algorithms are directly compared to each other and to the idealized cases in Fig. 6 with respect to their true positive detection rate. The photon energy of 20 keV and signal noise correspond to the typical HXRS operational point while the count rate range is extended well beyond the HXRS limits (operational point: 100–400 kcps), especially toward higher frequencies, to better visualize the performance limitations of the detection

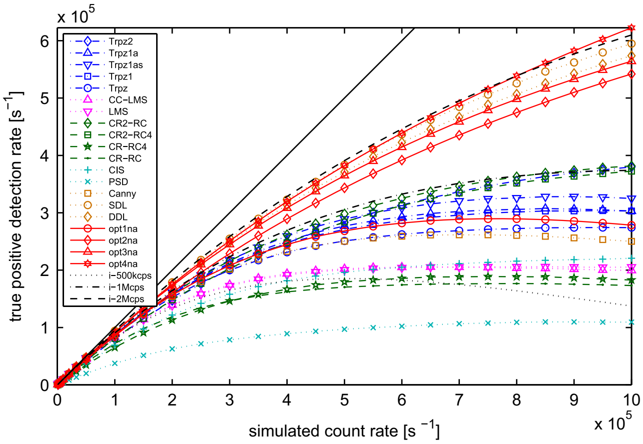

Fig. 6. True positive detection vs simulated count rates for all analyzed algorithms including three idealized reference ones: 20 keV photons with experimental and Gaussian ($\sigma = 1$ keV) noise.

algorithms. In addition to the experimentally measured noise, a white Gaussian noise component of $\sigma = 1$ keV is added in the preamplifier stage.

Very restrictive algorithms aiming to recognize the whole pulse shape, represented here by the PSD, are very limited in the count rate and stay far below the ideal 500 kcps limit. The other investigated algorithms can be classified into three distinct groups; within each group a change of simulation parameters may still yield a reordering.

In the lowest group we find the simple CIS algorithm and the analog methods using only one differentiator. The LMS and CC-LMS show similar performance and are therefore assigned to this group as well, although they approach the second group for lower count rates.

The main, middle group contains mainly FIR filters including the trapezoidal and the Canny algorithms as well as the optimum filter 1 (opt1na). The best performance in this group is achieved by double differentiation IIR and FIR filters coming close to the idealized 1 Mcps case.

The top performance with respect to true positive detection rate is obtained with the other optimum filters 2, 3 and 4, and by delay line algorithms. All of these approach the performance of the idealized 2 Mcps detection.

5.2.2. False detection

Although the true positive detection or throughput at high count rates is an important parameter, the classification made in the previous section is certainly not a general one. One parameter demonstrating this is the percentage of false positive detection, plotted in Figs. 7–9.

In the top group (Fig. 7) it can be clearly seen that the delay line algorithms, especially the double one, are unusable in the present parameter range since the noise leads to an unacceptable fraction of false positive detection.

Within a selection of the middle group with good throughput (Fig. 8), the FIR filter algorithms based on one differentiation suffer from practically no false positive detection while those using a second differentiator (Trpz2 and $(CR)^2(RC)^4$) do have a finite false positive fraction, albeit still arguably fairly low.

For the algorithms in the bottom throughput rung (Fig. 9), there is a very similar subdivision. Here, the CC-LMS and LMS algorithms select the pulses quite restrictively and are therefore very resistant to false positive detection. In the case of the CIS and PSD algorithms, a significant false detection level is observed, attributable to the limited noise filtering.

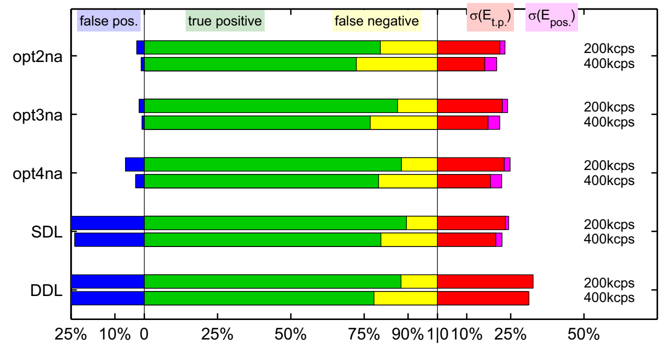

Fig. 7. Comparison of figures of merit for the 5 algorithms achieving the best count rate performance for 20 keV photons at 200 and 400 kcps with experimental and Gaussian ($\sigma = 1$ keV) noise. From left to right the 5 figures of merit are the false positive detection rate, the true positive detection rate, the false negative detection rate, the energy standard deviation derived from true positive detection and the energy standard deviation including all (true and false) positive detection. The true positive and false negative detection rate bars are stacked since they add up to one. In contrast, the two energy deviation bars are not stacked but overlaid, starting from the same origin, labeled at the horizontal axis.

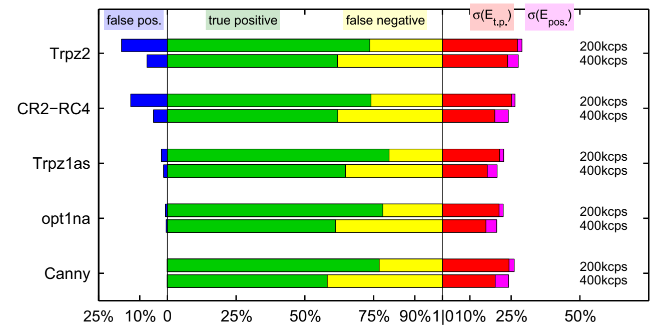

Fig. 8. Comparison of figures of merit as in Fig. 7 for 5 algorithms with intermediate count rate performance for the same modeled signal as in Fig. 7.

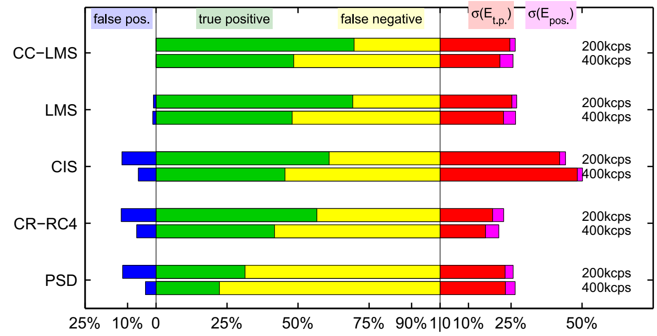

Fig. 9. Comparison of figures of merit as in Fig. 7 for 5 algorithms with relatively poor count rate performance for the same modeled signal as in Fig. 7.

5.2.3. Noise sensitivity

The comparisons of the algorithms in the count rate scan, especially the false detection analysis, lead one to expect significant differences in the algorithms' noise sensitivity. This is now studied, again around the HXRS operational point defined in Section 5.2.1, by varying the noise level from 0% to 200% at a fixed simulated count rate of 200 kcps. This noise level scan is equivalent to a photon energy scan since both result in scans of the signal-to-noise ratio.

True positive detection: The resulting true positive detection fraction is shown in Fig. 10. Substituting the experimentally measured noise by white Gaussian noise with $\sigma = 5$ keV yields essentially the same result, shown in Fig. 11.

One of the main observations is that at a low noise level the classification into 3 groups breaks down. All investigated algorithms except (CR) (RC) and PSD detect about 75–90% of the simulated pulses if the noise stays below 40% of the reference level. However, as the noise rises above 50% of the reference level, the optimum and single Trpz FIR filters, as well as the $(CR)^2(RC)^n$ analog methods are hardly affected while the other methods are quite significantly degraded, especially the $(CR)(RC)^4$ and CIS methods. The delay-line algorithms represent a special case: their performance appears, paradoxically, to increase with noise, but this is an artifact as discussed in the following.

False positive detection: The increase in the true positive detection performance of the delay line algorithms with increasing noise can be understood by looking at the false detection rate, shown among the other figures of merit in Fig. 12. It is evident that the very limited noise rejection of the delay line algorithms leads to an extremely high false positive detection rate at a significant noise level, reaching in fact over 300% for the DDL (well beyond

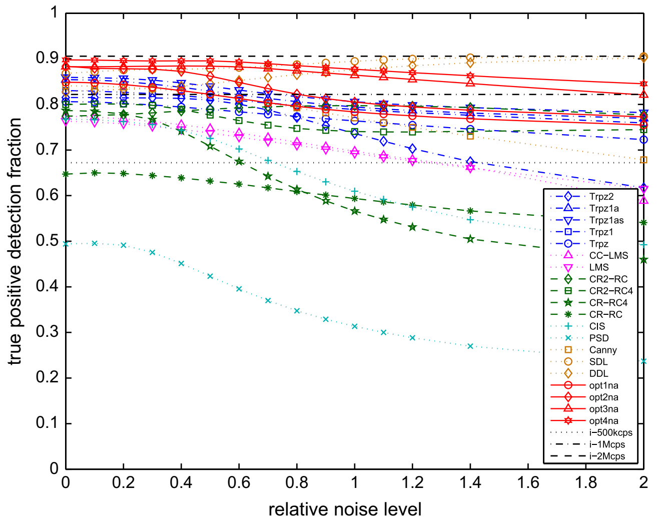

Fig. 10. True positive detection vs relative noise level (w.r.t. the noise used in Fig. 6) for all analyzed algorithms including three idealized reference ones for 20 keV photons.

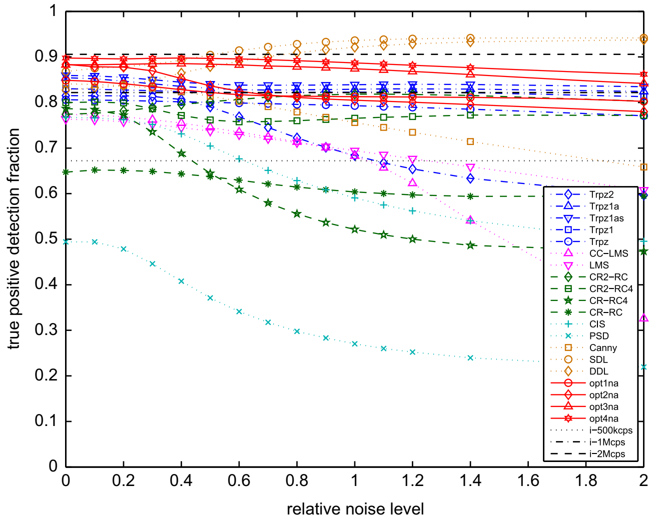

Fig. 11. True positive detection vs relative noise level (w.r.t. pure white Gaussian noise with standard deviation of 1 keV at the pre-amplifier stage and 5 keV after the amplifier stage) for all analyzed algorithms including three idealized reference ones for 20 keV photons.

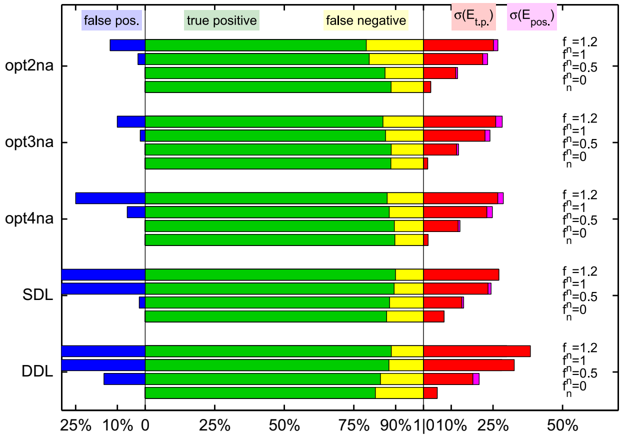

Fig. 12. Comparison of figures of merit as in Fig. 7 for the 5 algorithms achieving the best count rate performance for 20 keV photons at 200 kcps for several relative noise levels $f_n \in \{0, 0.5, 1, 1.2\}$ (selected from Fig. 10).

the range accommodated by Fig. 12). This results in spurious false positive detections that coincide but are uncorrelated with real pulses, leading them to be wrongly classified as true positive. As a consequence, the use of delay lines should be avoided if there is significant noise in the signal.

Again, the main algorithms of each group are shown in Figs. 12–14, for four different relative noise levels. The algorithms with strongly noise-degraded true positive detection rate (PSD, $(CR)^m(RC)^4$, CIS, Trpz2) also show the highest noise sensitivity regarding false positive detection. Hence, their usage should be limited to low noise applications too. In contrast to that, the false positive detection rate of the optimum and single Trpz FIR filters is also barely affected by the increasing noise level. Only the opt4na filter, which is most optimized for fast response, should be used with some caution at significantly higher noise levels.

5.3. *Energy accuracy*

Regarding the energy accuracy the most salient observation is that all algorithms exhibit a significant energy deviation of at least 15% at the HXRS operational point. This is, however, reasonable under the demanding conditions of high count rates and pulse heights in the range of the noise level. Figs. 7–9 indicate that most algorithms lie near the high energy accuracy limit achieved by several FIR and the $(CR)(RC)^4$ IIR filters. Only the CIS and the DDL algorithms are highly disturbed due to their poor noise filtering.

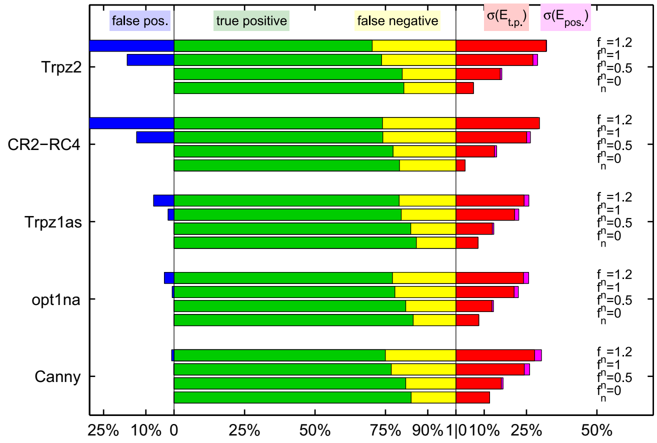

Fig. 13. Comparison of figures of merit as in Fig. 12 for 5 algorithms with intermediate count rate performance for the same modeled signal as in Fig. 12.

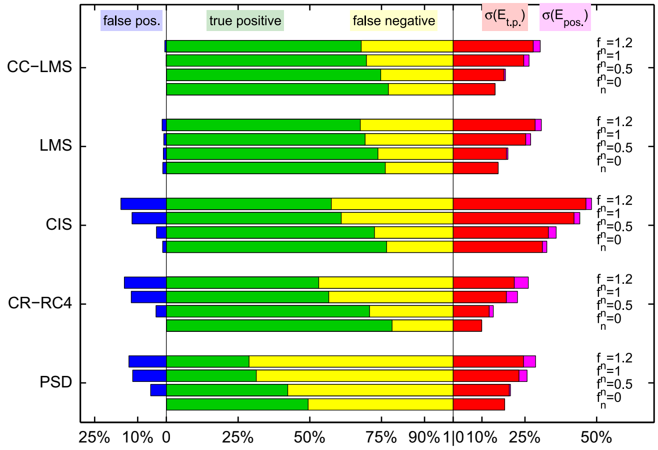

Fig. 14. Comparison of figures of merit as in Fig. 12 for 5 algorithms with relatively poor count rate performance for the same modeled signal as in Fig. 12.

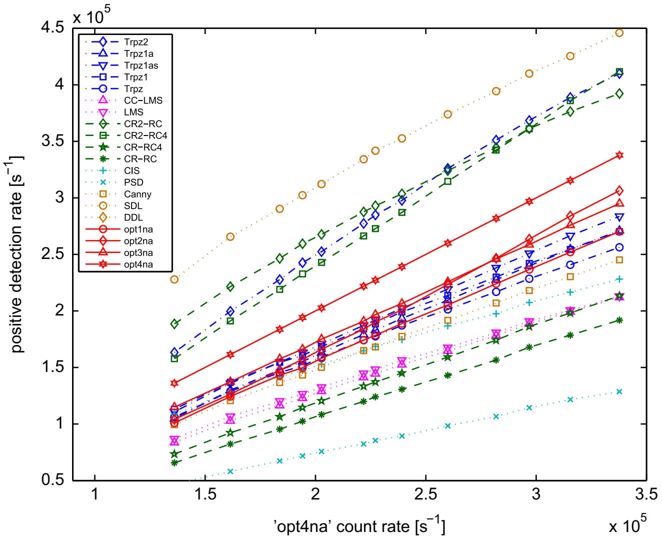

Fig. 15. Experimental positive detection vs the 'opt4na' count rate for all analyzed algorithms and a threshold of 13 keV for the signal of HXRS chord 9, TCV discharge #45252, $t = [0.4, 1.7]$ s, 13 time bins of 100 ms each. Except for CIS and PSD, dynamic threshold detection was used. The DDL trace lies outside the range of the plot due to very high false positive detection.

As for the previously discussed figures of merit, the energy accuracy is also affected by a changing signal-to-noise ratio. As can be seen in Figs. 12–14, the energy deviation, as expected, increases with the noise level. Interestingly, however, several algorithms exhibit a significant energy deviation even when the noise level is equal to zero. This effect is generally stronger for algorithms with low count rate performance. Those suffer mainly from poor pulse separation which causes consecutive pulses to influence both the pulse detection and the pulse height analysis of each other.

5.4. Experimental validation

The simulation-based benchmarking was validated using an experimentally measured signal from the HXRS diagnostic (TCV discharge #45252, time trace part shown in Fig. 5). The positive detection rate was compared to a reference method (opt4na) around the HXRS operational point (100–400 kcps). As can be seen in Fig. 15, the result of the simulation is essentially reproduced. Only the double differentiating methods (Trpz2, $(CR)^2(RC)^n$) and the delay line methods report a significantly higher count rate. This is mainly due to an increased false positive detection.

5.5. Computation/real-time applicability

The computational effort required by most of the algorithms discussed in this paper is quite limited. To give a quantitative measure we list the CPU time required by our implementation on a single core of a present x86-64 CPU (Intel Core i7-2760QM) for 2 s of signal acquired or simulated at 12M samples/s: the signal treatment using digital FIR or digitally emulated analog filters takes about 1 s, only the cross-correlation and digital band-pass filters require significantly higher computational effort (about one order of magnitude). The threshold or rise threshold detection takes about 1.5 s while the use of a dynamic threshold or multiple condition detection increases the computational time as well. The level or rise evaluation analysis is negligible (about 10 ms). Algorithms with significantly higher computational effort (about one order of magnitude) are therefore the CC-LMS, CIS and PSD algorithm. The post-processing to obtain a histogram in time and energy bins takes about 2 s at high count rates and is significantly lower at low count rates. The real-time applicability is already proven for digitally implemented $(CR)^m(RC)^n$ filters in Ref. [13] and digital FIR filters in Ref. [14], at sampling rates well beyond that of the HXRS. It is planned to implement real-time pulse processing for the HXRS system too, using the FPGAs on the digitizer cards (D-tAcq ACQ216CPCI).

6. Conclusions

A complete set of digital pulse processing methods was individually described, jointly implemented and compared within a general benchmarking framework. This ensures the general applicability of the presented results for any kind of digital pulse processing application, far beyond the scope of the specific measuring apparatus to which these results were first applied [2]. Although the focus was initially placed on high count rates and significant noise levels, the extension of the analysis to both low count rates and noise levels was straightforward and seamlessly integrated.

The main message is that the implementational and computational effort of essentially all the methods presented is comparable while the results differ significantly. Therefore the importance of making the right choice of pulse processing method should not be underestimated.

The best overall performance was obtained by optimum FIR filters that were also superior on each of the individual benchmarking figures of merit. Therefore these filters are clearly the top choice. FIR filters constructed on the basis of trapezoidal filters, along with the analog $(CR)^2(RC)^n$ method, lie only slightly behind. Since the filters of the Trpz family are among the easiest to implement, they can also be a reasonable solution if one wants to avoid the optimization process. The $(CR)^2(RC)^n$ is consequently the best-performing method that can be easily implemented as an analog system [24].

The use of delay lines should only be considered for virtually noiseless systems. Other algorithms studied, such as the $(CR)(RC)^n$, CC-LMS, LMS, PSD and CIS methods, are not recommended, especially not in high count rate applications.

Acknowledgments

This work was supported in part by the Swiss National Science Foundation.

References

[1] S. Gnesin, S. Coda, J. Decker, Y. Peysson, Rev. Sci. Instrum. 79 (10) (2008) 10F504, http://dx.doi.org/10.1063/1.2957843.

[2] J. Kamleitner, S. Gnesin, S. Coda, P. Marmillod, J. Decker, A new hard X-ray spectrometer for suprathermal electron studies in TCV, in: Europhysics Conference Abstracts 36F, 2012, P1.010.

[3] S. Coda, Nucl. Fusion 51 (9) (2011) 094017.

[4] J.B. Simões, C.M.B.A. Correia, Nucl. Instrum. Methods Phys. Res. Sect. A: Acceler. Spectrom. Detectors Assoc. Equip. 422 (1–3) (1999) 405 http://dx.doi.org/10.1016/S0168-9002(98)00992-9.

[5] J.M.R. Cardoso, J.B. Simões, C.M.B.A. Correia, Nucl. Instrum. Methods Phys. Res. Sect. A: Acceler. Spectrom. Detectors Assoc. Equip. 422 (1–3) (1999) 400 http://dx.doi.org/10.1016/S0168-9002(98)00991-7.

[6] M. Mutterer, W.H. Trzaska, G.P. Tyurin, A.V. Evsenin, J. von Kalben, J. Kemmer, M. Kapusta, V.G. Lyapin, S.V. Khlebnikov, IEEE Trans. Nucl. Sci. NS-47 (3) (2000) 756, http://dx.doi.org/10.1109/23.856510.

[7] D.M. Scates, J.K. Hartwell, Appl. Radiat. Isot. 63 (4) (2005) 465, http://dx.doi.org/10.1016/j.apradiso.2005.05.052.

[8] H. Seta, M.S. Tashiro, Y. Terada, Y. Shimoda, K. Onda, Y. Ishisaki, M. Tsujimoto, T. Hagihara, Y. Takei, K. Mitsuda, K.R. Boyce, A.E. Szymkowiak, AIP Conf. Proc. 1185 (1) (2009) 278, http://dx.doi.org/10.1063/1.3292332.

[9] S. Li, X. Xu, H. Cao, G. Yuan, Q. Yang, Z. Yin, A real-time n/γ digital pulse shape discriminator based on FPGA, Appl. Radiat. Isot. 72 (0) (2013) 30, http://dx.doi.org/10.1016/j.apradiso.2012.10.004.

[10] K. Vetter, A. Kuhn, M.A. Deleplanque, I.Y. Lee, F.S. Stephens, G.J. Schmid, D. Beckedahl, J.J. Blair, R.M. Clark, M. Cromaz, R.M. Diamond, P. Fallon, G.J. Lane, J.E. Kammeraad, A.O. Macchiavelli, C.E. Svensson, Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detectors Assoc. Equip. 452 (1–2) (2000) 223, http://dx.doi.org/10.1016/S0168-9002(00)00430-7.

[11] T. Tada, K. Hitomi, T. Tanaka, Y. Wu, S.-Y. Kim, H. Yamazaki, K. Ishii, Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detectors Assoc. Equip. 638 (1) (2011) 92, http://dx.doi.org/10.1016/j.nima.2011.02.070.

[12] S.H. Byun, H. Peng, Radiation & Radioisotope Methodology, 2012, ⟨http://www.science.mcmaster.ca/medphys/courses/166-med-phys-4r06.html⟩.

[13] M. Nakhostin, IEEE Trans. Nucl. Sci. NS-58 (5) (2011) 2378, http://dx.doi.org/10.1109/TNS.2011.2164556.

[14] V.T. Jordanov, Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detectors Assoc. Equip. 670 (0) (2012) 18, http://dx.doi.org/10.1016/j.nima.2011.12.042.

[15] A. Pullia, A. Geraci, G. Ripamonti, Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detectors Assoc. Equip. 439 (2–3) (2000) 378, http://dx.doi.org/10.1016/S0168-9002(99)00897-9.

[16] Y. Takenaka, A. Uritani, K. Inoue, H. Sakai, C. Mori, Appl. Radiat. Isot. 49 (9–11) (1998) 1219, http://dx.doi.org/10.1016/S0969-8043(97)10049-5.

[17] M. Faisal, R.T. Schiffer, M. Flaska, S.A. Pozzi, D.D. Wentzloff, Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detectors Assoc. Equip. 652 (1) (2011) 479, http://dx.doi.org/10.1016/j.nima.2010.10.072.

[18] K.R. Boyce, M.D. Audley, R.G. Baker, J.J. Dumonthier, R. Fujimoto, K.C. Gendreau, Y. Ishisaki, R.L. Kelley, C.K. Stahle, A.E. Szymkowiak, G.E. Winkert, Design and performance of the ASTRO-E/XRS signal processing system Proc. SPIE 3765 (1) (1999) 741. http://dx.doi.org/10.1117/12.366557.

[19] H. Sakai, Y. Takenaka, A. Uritani, K. Inoue, C. Mori, Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detectors Assoc. Equip. 400 (2–3) (1997) 401, http://dx.doi.org/10.1016/S0168-9002(97)01015-2.

[20] M. van Berkel, G. Witvoet, M.R. de Baar, P.W.J.M. Nuij, H.G. ter Morsche, M. Steinbuch, Fusion Eng. Des. 86 (12) (2011) 2908, http://dx.doi.org/10.1016/j.fusengdes.2011.07.002.

[21] J. Gomez, I. Melgar, J. Seijas, Upgrading Pulse Detection with Time Shift Properties Using Wavelets and Support Vector Machines, Proceedings of the World Automation Congress (WAC-04) 2004.

[22] C.J.C. Burges, Data Mining Knowl. Discovery 2 (2) (1998) 121, http://dx.doi.org/10.1023/A:1009715923555.

[23] S. Gnesin, Electron Cyclotron Heating and Suprathermal Electron Dynamics in the TCV Tokamak, Ph.D. thesis, EPFL, 2011.

[24] Y. Peysson, F. Imbeaux, Rev. Sci. Instrum. 70 (10) (1999) 3987, http://dx.doi.org/10.1063/1.1150025.

[25] V. Radeka, Nucl. Instrum. Methods 99 (3) (1972) 525, http://dx.doi.org/10.1016/0029-554X(72)90666-0.

[26] J. Canny, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6) (1986) 679, http://dx.doi.org/10.1109/TPAMI.1986.4767851.

[27] S. Riboldi, A. Geraci, R. Abbiati, E. Gatti, G. Ripamonti, A new method for LMS synthesis of optimum finite impulse response (FIR) filters with arbitrary time and frequency constraints and noises, in: Nuclear Science Symposium Conference Record, 2002 IEEE, vol. 1, 2002, pp. 198–202. doi:http://dx.doi.org/10.1109/NSSMIC.2002.1239298.

[28] B. Joly, G. Montarou, J. Lecoq, G. Bohner, M. Crouau, M. Brossard, P.-E. Vert, IEEE Trans. Nucl. Sci. NS-57 (1) (2010) 63, http://dx.doi.org/10.1109/TNS.2009.2031871.

[29] A. Pullia, S. Riboldi, IEEE Trans. Nucl. Sci. NS-51 (4) (2004) 1817, http://dx.doi.org/10.1109/TNS.2004.832564.