

PAPER

## Sparse deconvolution methods for online energy estimation in calorimeters operating in high luminosity conditions

To cite this article: T. Teixeira *et al* 2021 *JINST* **16** P09008

View the [article online](#) for updates and enhancements.

### You may also like

- [Point-spread Function Deconvolution of the IFU Data and Restoration of Galaxy Stellar Kinematics](#)  
Haeun Chung, Changbom Park and Yong-Sun Park
- [Hyperspectral Image Classification Using Discriminative Dictionary Learning](#)  
Y Zongze, S Hao, J Kefeng et al.
- [Two sparsities are better than one: unlocking the performance benefits of sparse-sparse networks](#)  
Kevin Hunter, Lawrence Spracklen and Subutai Ahmad



The Electrochemical Society  
Advancing solid state & electrochemical science & technology



**249th  
ECS Meeting**  
May 24-28, 2026  
Seattle, WA, US  
*Washington State  
Convention Center*

# Spotlight Your Science

***Submission deadline:  
December 5, 2025***

**SUBMIT YOUR ABSTRACT**

RECEIVED: March 24, 2021

REVISED: July 16, 2021

ACCEPTED: August 19, 2021

PUBLISHED: September 9, 2021

# Sparse deconvolution methods for online energy estimation in calorimeters operating in high luminosity conditions

T. Teixeira,<sup>a,\*</sup> L. Andrade<sup>a</sup> and J.M. de Seixas<sup>b</sup>

<sup>a</sup>Electrical Engineering, Federal University of Juiz de Fora,  
Minas Gerais, Brazil

<sup>b</sup>Signal Processing Lab, COPPE/POLI — Federal University of Rio de Janeiro,  
Rio de Janeiro, Brazil

E-mail: [tiago.teixeira@engenharia.ufjf.br](mailto:tiago.teixeira@engenharia.ufjf.br)

**ABSTRACT:** Energy reconstruction in calorimeters operating in high luminosity particle colliders has become a remarkable challenge. In this scenario, pulses from a calorimeter front-end output overlap each other (pile-up effect), compromising the energy estimation procedure when no preprocessing for signal disentanglement is accomplished. Recently, methods based on signal deconvolution have been proposed for both online and offline reconstructions. For online processing, constraints concerning fast processing, memory requirements, and cost implementation limit the overall performance. Offline reconstruction allows the use of Sparse Representation theory to implement sophisticated Iterative Deconvolution methods. This paper presents Iterative Deconvolution methods based on Sparse Representation algorithms whose computational cost is effective for online implementation. Using simulated data, current techniques were compared to the proposed Sparse Representation ones for performance validation in the online environments. Analysis has shown that, despite the higher computational cost, when compared to standard methods, the performance improvement may justify the use of the proposed techniques, in particular for the Separable Surrogate Functional, which is shown to be feasible for implementation in modern FPGAs.

**KEYWORDS:** Data acquisition circuits; Data acquisition concepts; Digital electronic circuits; Digital signal processing (DSP)

ARXIV EPRINT: [2103.12467](https://arxiv.org/abs/2103.12467)

\*Corresponding author.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The pile-up effect	2
<b>2</b>	<b>Techniques</b>	<b>3</b>
2.1	FIR filter based energy estimation	4
2.2	The sparse representation	4
<b>3</b>	<b>Proposed sparse representation based on deconvolution methods</b>	<b>5</b>
3.1	Matching pursuit	6
3.1.1	Orthogonal matching pursuit	6
3.1.2	Least squares orthogonal matching pursuit	7
3.2	Iterative-shrinkage	7
3.2.1	Separable surrogate functional initialization	8
<b>4</b>	<b>Performance results</b>	<b>8</b>
4.1	Physical model	8
4.2	Parameters choice	9
4.2.1	FIR filter design	10
4.2.2	Matching pursuit parameters	10
4.2.3	Separable surrogate functional parameters	10
4.3	Performance	14
4.3.1	Detection efficiency	16
4.3.2	Analysis summary	16
<b>5</b>	<b>Hardware implementation and cost analysis</b>	<b>17</b>
5.1	Firmware implementation	19
5.2	Cost analysis	19
<b>6</b>	<b>Conclusion</b>	<b>20</b>

---

## 1 Introduction

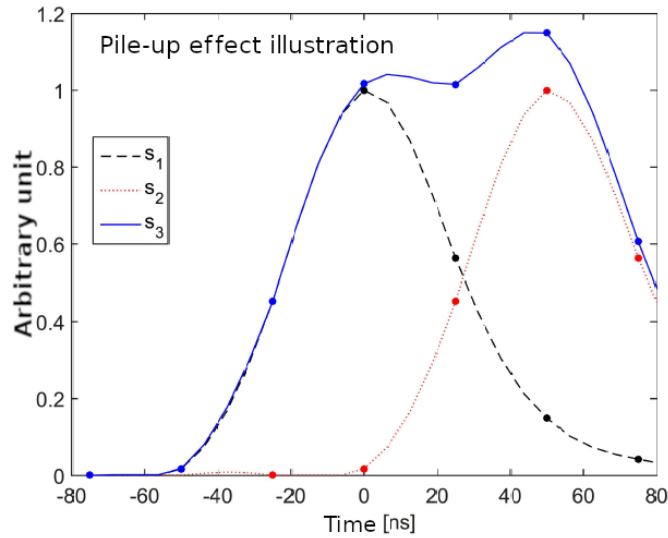
High-Energy Physics (HEP) experiments usually require sophisticated instrumentation systems, especially when particle colliders are involved [1]. In order to analyze increasingly rare physical phenomena, there is a tendency of increasing their collision rates and the number of particles per collision, thus increasing the luminosity achieved by the experiment [2]. This is the case of the most potent collider ever built, the LHC (Large Hadron Collider) [3] at CERN (*Organisation Européenne pour la Recherche Nucléaire*) [4]. The LHC collides bunches of protons every 25 nanoseconds

and has been establishing records in terms of both collider energy and luminosity [5]. Moreover, upgrades that are on the way for the next LHC operation periods (Run 3 and beyond) will increase even more the luminosity [6], towards a projected peak around  $5$  to  $7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  [7].

For new physics identification at the LHC, calorimeters [8] play an important role, as they measure the energy of the incoming particles and are essential for triggering purposes [9]. The two general-purpose LHC experiments (ATLAS [10] and CMS [11]) comprise hundred of thousands of calorimeter readout channels each and employ state-of-art detector technology. However, the LHC's enormous luminosity imposes severe restrictions on calorimetry performance as, despite being fast, the signal generated in the detector's front-end electronics requires a certain number of bunches crossings (BCs) to be fully developed. Therefore, signal pile-up effects may arise when the luminosity is increased.

The trigger system is often implemented through an online cascade signal processing chain consisting of two or more levels, where the rate reduction is proportional to the level of the algorithm complexity for final event selection. Due to the high event rate, the lower filtering levels are usually implemented in hardware and rely very much on online energy estimation, which is addressed in this paper for applications where signal pile-up becomes an issue.

### 1.1 The pile-up effect



**Figure 1.** Example of the pile-up effect. See text.

An example of out-of-time signal pile-up in collider experiments can be seen in figure 1, where signals were digitized by a 40 MHz clock (25 ns period, as in the LHC). The unipolar signal generated in a hypothetical calorimeter cell by a given collision is referred to as  $s_1$ , and 50 ns later a new signal ( $s_2$ ) from two bunch collisions after reaches that particular cell. Supposing a 150 ns readout acquisition length, the  $s_3$  signal is formed, which is actually the stacking of both  $s_1$  and  $s_2$  signals. This pile-up effect leads to a sub-optimal operation of the classical energy estimation algorithms, which are based on Optimal Filtering (OF) theory [12–15].

The calorimeter signal shape is a linear combination of a signal with neighboring pileup signals, where the signal amplitude is proportional to the target energy value [8]. Therefore, the sensor output can be modeled as the convolution between the target sequence (energy deposition per bunch crossing) and the calorimeter normalized (unitary amplitude) reference pulse shape. To recover the target sequence, a deconvolution algorithm may be applied [16].

Different deconvolution algorithms have been proposed for energy estimation in high-luminosity calorimeter operation. The work in [17] proposes an offline procedure in two steps, where a signal detection per bunch crossing is performed first, followed by a multi-amplitude estimation procedure. Still focusing on offline reconstruction, the work in [18] implements an approach based on Sparse Representation (SR) [19] of data, presenting remarkable results, over-performing standard deconvolution methods. Sparse Representation was originally developed and applied in signal compression and image denoising routines [20, 21]. The work in [18] inaugurated the Sparse Representation energy reconstruction applications. However, its cost implementation is presently forbidden for online data-stream.

Concerning online processing, a deconvolution method based on Finite Impulse Response (FIR) [22] filters was proposed in [23] and it has shown to be very attractive due to its lower cost implementation in hardware. However, recently Sparse Representation theory has developed families of algorithms targeting cost implementation reduction [24–26]. Although FIR filter structures are known to be the most economic approach in terms of hardware resources, the cost-effectiveness of such modern Sparse Representation techniques has paved the way for online energy reconstruction implementation. Those approaches have already been used in several online applications like machine learning [27] and big data [28].

Therefore, the main goal of this paper is to use modern Sparse Representation theory in a design that fulfills online requirements. The adaptation of some of those techniques is proposed envisaging energy estimation in calorimeter cells, opening a new research area in online energy reconstruction. Since Sparse Representation has shown to be the most accurate energy reconstruction procedure under pile-up, the methods proposed here can be used as an alternative for FIR filter implementation in applications where the hardware resource minimization is not the main constraint. A comparison among the aforementioned methods, in terms of cost implementation and energy reconstruction performance, is also provided, indicating the best situation for the usage of each approach.

In order to complete the discussion, a possible implementation for one of the proposed Sparse Representation methods (in this case, the Separable Surrogate Functional) is outlined at the end of the paper. The circuitry was developed in Hardware Description Language (HDL) [29] envisaging a straightforward implementation in Field-Programmable Gate Array (FPGA) technology [30].

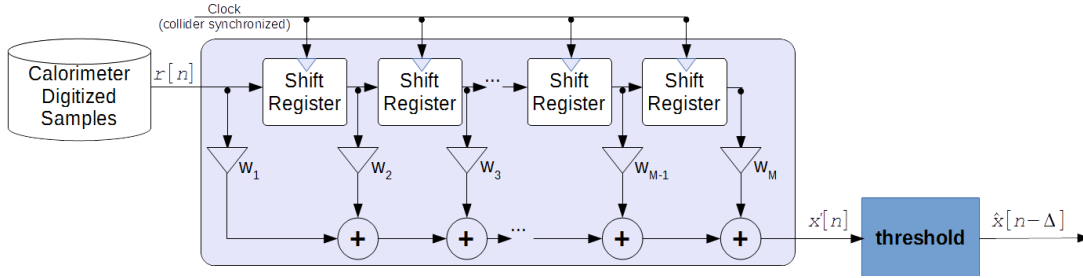
## 2 Techniques

In this section, firstly the FIR approach for high-luminosity online energy estimation is briefly reviewed, as it is being used here for baseline comparison. In the sequence, the basics of modern sparse signal processing are presented, as a way to exploit the nonlinear statistics that comes with the out-of-time signal pileup.

## 2.1 FIR filter based energy estimation

Figure 2 shows the basic structure for online energy estimation based on a FIR deconvolutional filter [23]. The signal  $r[n]$  represents the measured information from a single front-end calorimeter sensor, after applying the free-running (an uninterrupted) Analog-to-Digital Converter (ADC). A cascade of registers, synchronous with the ADC clock, implements a shift-register structure [30], which is responsible for storing the latest  $M$  digitized samples. The filter order ( $M - 1$ ) depends on both the calorimeter reference pulse-shape width and the signal pile-up intensity. For deconvolution, the filter order is often higher than the reference pulse width (two times on average), since the filter needs tail information of neighbor superimposed signals to perform the desired source separation. The equation employed in this deconvolution procedure is:

$$x'[n] = \sum_{l=0}^{M-1} (w[n-l]r[n]) \quad (2.1)$$



**Figure 2.** The FIR filter structure implementing signal deconvolution in online energy estimation, based on [23].

The weighting vector  $\mathbf{w}$  is obtained offline, based on deconvolution principles. Details on the filter design can be found in the original paper [23]. The samples are, then, linearly combined at the filter output, which delivers new information at each clock. A threshold is applied to the filter output to avoid small artifacts from the deconvolution process in adjacent bunch crossings, mainly small negative energy estimations.

The system output corresponds to the reconstructed target information sequence  $\hat{x}[n]$ , which is a straightforward estimation of the energy deposit per bunch crossing ( $x[n]$ ). The system starts delivering the output sequence  $\Delta$  clock cycles after the first sample goes through the processing chain. This delay is necessary because the system needs information from previous samples in order to perform the deconvolution procedure. Typical values of  $\Delta$  are around half of the filter order.

Despite of its very suitable hardware implementation, the purely linear operation of FIR filters does not allow access to the Higher Order Statistics (HOS) [31] present in the embedded pile-up noise, which may be exploited for further performance improvement. The proposed Sparse Representation based deconvolution methods access the Higher Order Statistics information indirectly through the usage of non-linear functions in the deconvolution process.

## 2.2 The sparse representation

The Sparse Representation theory has been applied for energy reconstruction with great success for offline environments in [18]. On that paper, the implementation was based on Linear Programming



(LP) [32], which is a standard procedure with high computational cost. Recently, modern Sparse Theory has raised with more cost-effective algorithms [19]. Modifications on standard Matching Pursuit (MP) methods [33] have presented an intrinsic relationship with energy estimation tasks. In those approaches, a signal detection iterative procedure is executed first, using a bank of linear Matching-Filter (MF) [34] based filters. After some iterations, all the superimposed signals within an acquisition window are detected. Their amplitudes are, then, estimated simultaneously using the Least Square (LS) algorithm [35].

Another Sparse Representation family of methods that is being employed due to its excellent cost-benefit properties is the Iterative-Shrinkage (IS) algorithms [36]. These methods combine standard linear Coordinate-Descent (CD) algorithms [37] with a linear-by-part Shrinkage function, in order to promote sparsity, and becoming one of the most cost-effective algorithms for Sparse Representation.

### 3 Proposed sparse representation based on deconvolution methods

The Sparse Representation based deconvolution methods are designed by using a windowed acquisition point of view. The method is guided by the matrix formulation from the convolution operation of finite size sequences, as proposed in [18]. Given a normalized calorimeter reference pulse-shape, represented by the vector  $\mathbf{h}$  of size  $V$  and a target sequence, represented by the vector  $\mathbf{x}$  (energy per bunch crossing) of size  $G$ , the convolution between these two signals produces the measured signal  $\mathbf{r}$ ; that is, the calorimeter readout channel response. Because there are  $G$  bunch crossings, and the fact that the calorimeter responses extends  $V$  samples beyond the last bunch crossing, the total samples considered for the vector  $\mathbf{r}$  are  $G + V - 1$  [22]. This convolution process can be written as:

$$\mathbf{r} = \mathbf{H}\mathbf{x} \quad (3.1)$$

where  $\mathbf{H}$  is the convolution matrix of size  $U \times G$  (columns of  $\mathbf{H}$  contain shifted versions of the vector  $\mathbf{h}$ , with zero padding). The windowed deconvolution procedure consists of recovering the sequence  $\mathbf{x}$  when  $\mathbf{r}$  is measured and  $\mathbf{H}$  is known. When the size of the vector  $\mathbf{r}$  is greater than the one of  $\mathbf{x}$ , there are infinite solutions for this system of equations. In [18], it was shown that, for deconvolution purposes, the sparsest representation of  $\mathbf{x}$  is the correct solution when the target sequence is characterized by impulsive signals.

According to the Sparse Representation theory [19], the sparsest solution for equation (3.1) can be found solving the optimization Problem  $P_\ell$  for  $0 \leq \ell \leq 1$

$$(P_\ell) : \min_{\mathbf{x}} \|\mathbf{x}\|_\ell^\ell \text{ subjected to } \mathbf{r} = \mathbf{H}\mathbf{x} \quad (3.2)$$

where

$$\|\mathbf{x}\|_\ell^\ell = \sum_b |x_b|^\ell \quad (3.3)$$

is the  $\ell$ -norm of the vector  $\mathbf{x}$ .

For  $\ell = 1$ , this results in a typical Linear Programming problem [38]. The use of Linear Programming for Sparse Representation aiming at energy estimation has been outlined in [18], where this method was shown to outperform other windowed deconvolution methods. However,

the focus of that paper was on offline energy reconstruction, since Linear Programming presents a very high computational cost, becoming usually prohibitive for online execution. In the sequence, two families of Sparse Representation algorithms are presented in order to solve the  $P_\ell$  problem with different perspectives, both focusing on computational cost-effectiveness procedures.

### 3.1 Matching pursuit

For  $\ell = 0$ , equation (3.3) results in the number of non-zero elements in  $\mathbf{x}$  and the Sparse Representation solution given by  $P_0$  becomes evident. However,  $P_0$  is not a mathematically tractable problem [19]. This is a combinatorial problem infeasible to be computationally implemented in practice.

To avoid the test of all combinatorial possibilities, Matching Pursuit seeks for similarities between columns of the matrix  $\mathbf{H}$  and the vector  $\mathbf{r}$ , selecting the smallest number of columns that, when linearly combined, produce the vector  $\mathbf{r}$  added to a small residue vector  $\mathbf{d}$ .

Given the support matrix  $\mathbf{H}_s$ , which comprises the selected columns of  $\mathbf{H}$ , the residue vector  $\mathbf{d}$  is

$$\mathbf{d} = \mathbf{H}_s \mathbf{x}_s - \mathbf{r} \quad (3.4)$$

where the vector  $\mathbf{x}_s$  has the amplitudes for each signal in  $\mathbf{H}_s$  (the columns of  $\mathbf{H}_s$ ). The components in  $\mathbf{x}_s$  are chosen in order to minimize the square module of the residue:

$$\epsilon^2 = \mathbf{d}^T \mathbf{d} = |\mathbf{H}_s \mathbf{x}_s - \mathbf{r}|^2 \quad (3.5)$$

where  $T$  indicates a vector transposition.

The vector  $\mathbf{x}_s$  that minimizes the equation (3.5) is obtained by the Least Square equation

$$\mathbf{x}_s = (\mathbf{H}_s^T \mathbf{H}_s)^{-1} \mathbf{H}_s^T \mathbf{r} \quad (3.6)$$

The Matching Pursuit algorithms select columns of the matrix  $\mathbf{H}$  with the highest potential of minimizing  $\epsilon^2$  [39]. Depending on the way this task is performed, a specific variation of the basic Matching Pursuit algorithm is applied. In this work, the Orthogonal Matching Pursuit (OMP) [40] and the Least Square Orthogonal Matching Pursuit (LS-OMP) [41] will be outlined.

#### 3.1.1 Orthogonal matching pursuit

This method selects the column  $\mathbf{h}_j$  of the matrix  $\mathbf{H}$  that best matches the current residue  $\mathbf{d}$ , minimizing the cost function

$$Q(x_j) = |x_j \mathbf{h}_j - \mathbf{d}|^2 \quad (3.7)$$

The value  $x_j$  that minimizes this function is the Matching Pursuit equation

$$x_j = \frac{\mathbf{h}_j^T \mathbf{d}}{\mathbf{h}_j^T \mathbf{h}_j} \quad (3.8)$$

This value is computed for all  $\mathbf{h}_j$  columns not yet inserted in the support matrix  $\mathbf{H}_s$ . The column with the smallest  $Q(x_j)$  value is selected, the new residue vector is computed and a new iteration is performed until  $\epsilon$  becomes lower than a given threshold  $\epsilon_0$ . It can be shown [41] that all the columns of  $\mathbf{H}_s$  are orthogonal to the remaining residue  $\mathbf{d}$ .



### 3.1.2 Least squares orthogonal matching pursuit

The LS-OMP method updates temporarily  $\mathbf{H}_s$  with the current column  $\mathbf{h}_j$  in test and computes  $\epsilon$  directly (equation (3.5)). The temporary matrix  $\mathbf{H}_s$  that gives the smallest  $\epsilon$  value is kept. The process stops when the current residue is lower than  $\epsilon_0$  as well. Different from OMP, which computes a simple inner product (Matching Pursuit equation) to each column not yet inserted in  $\mathbf{H}_s$ , LS-OMP computes the Least Square (equation (3.6)) instead. Therefore, the computational cost of LS-OMP is higher than OMP.

## 3.2 Iterative-shrinkage

Iterative-Shrinkage algorithms are based on the Problem  $P_1$ . However, a relaxation is employed in order to allow a small residual square error  $\epsilon_0$  on the constraint.

$$(P_{1,\epsilon_0}) : \min_{\mathbf{x}} \|\mathbf{x}\|_1^1 \text{ subjected to } |\mathbf{H}\mathbf{x} - \mathbf{r}|^2 < \epsilon_0 \quad (3.9)$$

The problem  $P_{1,\epsilon_0}$  is a typical Linear Programming problem, as directly performed in [18] for offline signal reconstruction. However, focusing on cost-effectiveness, firstly, this problem is transformed to an optimization routine without constraint by the use of a Lagrange multiplier  $\lambda$  [19]

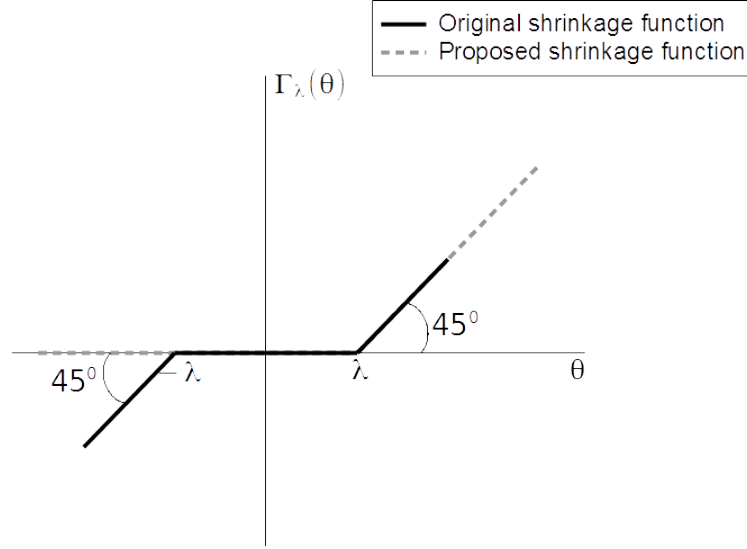
$$(P_{1,\lambda}) : \min_{\mathbf{x}} \|\mathbf{x}\|_1^1 + \lambda |\mathbf{H}\mathbf{x} - \mathbf{r}|^2 \quad (3.10)$$

where  $\epsilon_0$  is absorbed by  $\lambda$ . The Iterative-Shrinkage algorithms propose solving Problem  $P_{1,\lambda}$  by means of iterative Coordinate-Descent methods: given an initial guess vector  $\mathbf{x}^0$ , the optimal  $\mathbf{x}_{opt}$  value can be recursively inferred in a small number of steps. However, the use of standard Coordinate-Descent methods, like Newton-Raphson [42], becomes prohibitive for  $P_{1,\lambda}$  because of the discontinuity of the  $\ell$ -norm for  $\ell = 1$  [19].

In [43], the insertion of terms in  $P_{1,\lambda}$  is proposed, so that the minimum coordinate position  $\mathbf{x}_{opt}$  is not changed, but the multivariate problem is split into separated one-dimensional problems, which can be solved by parts. The resulting equation is named as Surrogate Function, and the respective Iterative-Shrinkage method is the Separable Surrogate Functional (SSF). After some algebraic manipulation, the iterative procedure can be compacted as follows

$$\mathbf{x}^{i+1} = \Gamma_{\lambda} \left( \mathbf{x}^i + \mu [\mathbf{H}^T (\mathbf{r} - \mathbf{H}\mathbf{x}^i)] \right) \quad (3.11)$$

where  $\mu$  is the step size in the direction of the minimum. The function  $\Gamma_{\lambda}(\theta)$  is the one-dimensional shrinkage function that should be applied to each component of the vector  $\mathbf{x}$ . In this paper, a modification on the shrinkage function is proposed in order to promote positive energy estimation, as shown in figure 3. This approach increases the performance of the method for further iterations, since negative (and wrong) estimations are discarded along the process. With this modification, the resulting function can be simply implemented through a subtraction followed by a threshold ( $\lambda$ ) operation. The argument in equation (3.11) is identified as a linear Gradient-Descent (GD) iteration [44] and comprises only addition and multiplication operations. The parameters  $\mu$  and  $\lambda$  are both obtained from simulation.



**Figure 3.** The original shrinkage function and the proposed modified version to promote positive energy estimation.

### 3.2.1 Separable surrogate functional initialization

One important prerequisite for online implementation of the Separable Surrogate Functional algorithm is to avoid a higher number of iterations. This can be accomplished by choosing a proper initial vector  $\mathbf{x}^0$ . In this work, a preprocessing routine is proposed, in order to initialize the solution close to the optimal one. The proposal is to implement the Least Square (equation (3.6)) for a full-support matrix  $\mathbf{H}$ :

$$\mathbf{x}^0 = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{r} = \mathbf{H}^+ \mathbf{r} \quad (3.12)$$

where  $\mathbf{H}^+ = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$  is the pseudo-inverse of matrix  $\mathbf{H}^T$  [45]. This is a constant matrix, which may be obtained offline.

## 4 Performance results

In order to validate the proposed algorithms, a toy Monte Carlo simulation was used for generating a set of consecutive samples, which represent the target energy values, per bunch crossing, for a single readout channel. Energy distributions and sensor's front-end were simulated following the characteristics found in modern calorimeters [46–48]. The pile-up effect was assessed through the calorimeter cell occupancy: null occupancy corresponds to the absence of hits in a given run and on the other extreme, 100% of occupancy means that an amount of energy is deposited at every particle collision that happens in a given run.

### 4.1 Physical model

Concerning hardware implementation for the target online operation, Sparse Representation and FIR filters comprise very different approaches. The latter produces its output signal in a sample-by-sample sequence, which is very suitable and straightforward for free-running environments. On the

other hand, Sparse Representation algorithms operate simultaneously on all the samples previously buffered within a window. Actually, this non-causal characteristics is one of the reasons of the increasing in performance, when compared to the FIR filter structures. Therefore, operating a windowed energy reconstruction algorithm in a free-running environment is one of the challenges addressed by the present work, as the online implementation of windowed methods demands extra digital circuitry for proper data-flow. In this paper, the proposed Sparse Representation methods are first demonstrated. Then, a hardware solution for online implementation is outlined.

Actually, the adaptation of such a non-causal method to online operation is possible thanks to some characteristics of modern colliders, as is the case of the LHC, which operates in a train-of-bunches [49, 50]. Due to particle injection stability issues, the accelerator needs small periods of empty-bunches intercalated with periods of bunch-trains [51]. At Run 2, for instance, typical physics runs were operated in a configuration of 48 filled bunches followed by 7 (or more) empty bunches [52]. It turns out that this issue encourages the use of windowed estimation methods as the ones proposed here, since such an operational approach eliminates two important design difficulties:

1. **Window size determination** — in adapting windowed methods to free-running environments, the window size is to be determined. In a train-of-bunches configuration, it is natural to choose the window size in order to fit an entire train within the processing window.
2. **Border effect** — a problem found in the windowing signal processing is that signals on the window border are split into neighbor windows. As a consequence, the estimation of bunch crossings closer to the window edge tends to reduce the processing efficiency. However, due to the train-of-bunches configuration, providing that the calorimeter reference signal width is smaller than the empty-bunches period, the border effect does not occur.

In this paper, bunch-trains of 48 bunches followed by 7 empty bunches were used. Thus, window lengths of 55 bunches (48 central plus the guard comprising three bunches on each side of the window and an extra bunch between windows) were employed to avoid the border effect.

Without loss in generality, the cell calorimeter shaped pulse was considered as unipolar [53–55], Gaussian-like, with an acquisition length of 150 ns. The collision rate was of 40 MHz (as in the LHC), resulting in a seven-sample pulse representation where the pulse peak corresponded to the central sample (4<sup>th</sup> sample). It is worth mentioning that the proposed method might be applied to any pulse shape, provided it represents the impulse response ( $h[n]$ ) to be deconvolved.

For optimal parameter determinations and for performance evaluations, the RMS of the error between the target ( $x[n]$ ) and the estimated ( $\hat{x}[n]$ ) sequences was employed (equation (4.1))

$$RMS\ Error = \sqrt{\frac{\sum_{n=1}^N (\hat{x}[n] - x[n])^2}{N}} \quad (4.1)$$

where  $N$  was the total number of samples in the test set (10,010).

## 4.2 Parameters choice

The simulation framework employed here was also used in recent related works [18, 23], where more details can be found. The signal characteristics were:

- The cell occupancy range: from 1% to 90%, in steps of approximately 10%.
- Energy deposition: from an exponential distribution of 360 MeV.
- Phase shift: an uniformly random pulse-phase in the range  $\pm 1$  ns is employed to each generated signal.
- Electronic noise: zero-mean white Gaussian noise with  $\sigma = 15$  MeV.

For general filter design and performance comparisons, signal data from 1,820 windows (with 55 bunches each) were considered, forming a data stream with 100,100 samples. The simulation data were split into ten subsets of ten thousand and ten events each, in order to evaluate the statistical fluctuations. The error bars were computed through a cross-validation procedure [56]. In this procedure, one subset was kept for testing and did not participate in the filter design, which used the remaining nine subsets. The testing subset was shifted for each fold, so that each subset could play once the role of the testing subset along the cross-validation procedure. The average performance and the Root Mean Square (RMS) value from the ten models developed along such a procedure were computed as the estimate of the expected performance and the corresponding error bar, respectively.

#### 4.2.1 FIR filter design

The baseline implementation was proposed in [23] (see figure 2). For filter design, two parameters must be tuned: the filter order and the threshold value for reducing the estimation error.

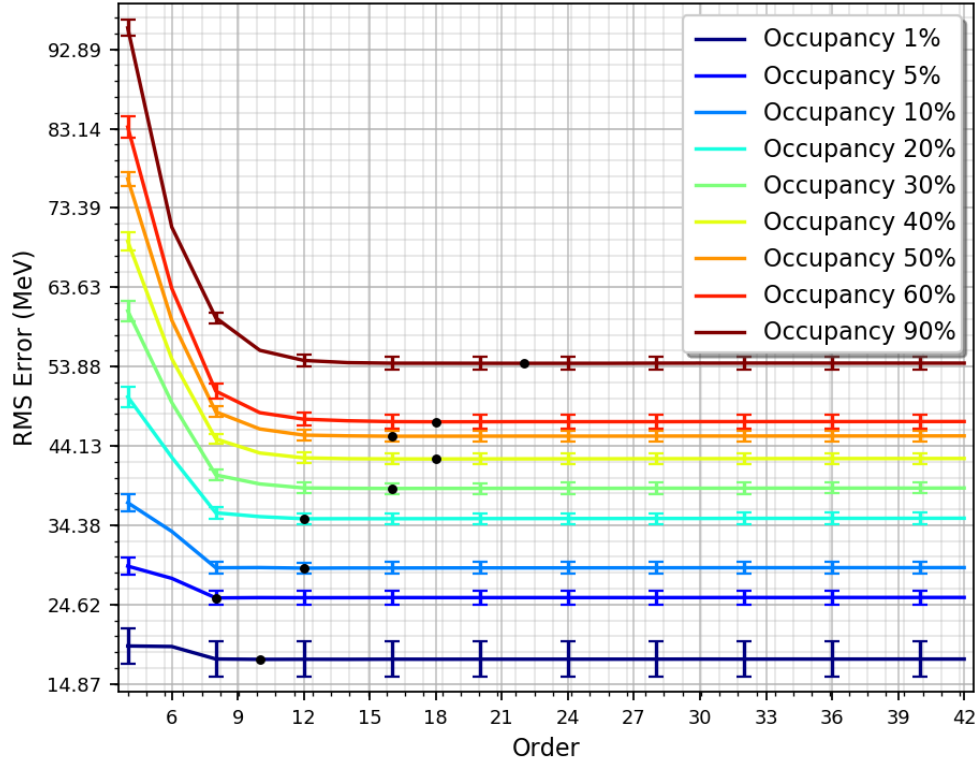
The energy estimation performance as a function of the filter order is shown in figure 4. In this figure, the output threshold was set to null aiming at avoiding negative energy estimations. Based on this figure, a filter order of 22 was kept fixed for further analysis (hence, the filter delay was  $\Delta = 11$  samples), regardless of the occupancy value, since the RMS error did not decrease for higher filter order values. For optimal threshold value determination, the estimation error as a function of the threshold value is plotted in figure 5 for each occupancy value. One can conclude that a threshold around 30 MeV might be used regardless of the occupancy level. As expected, the estimation error increased with the pile-up level.

#### 4.2.2 Matching pursuit parameters

The unique parameter to be determined for the Matching Pursuit methods is the squared value for the residue limit ( $\epsilon_0^2$ ). Figure 6 shows such error limits as a function of this parameter for both the OMP and the LS-OMP methods. Since this parameter is measured in  $\text{MeV}^2$ , its squared root value was employed for the sake of physical interpretation in terms of energy values in MeV. As expected, the optimal residue value decreased as the occupancy level was increased, requiring more signals in the support matrix  $\mathbf{H}_s$ . Although both methods achieved similar performance, it is worth mentioning that the amount of operations for LS-OMP implementation is much higher, as it executes a complete estimation with Least Square for each iteration whereas the OMP method applies the linear regression only at the final stage.

#### 4.2.3 Separable surrogate functional parameters

For Separable Surrogate Functional, two parameters require tuning: the step-size  $\mu$  for the Coordinate-Descent iterations and the  $\lambda$  value for the shrinkage function. For faster convergence, the  $\mu$  value may be determined dynamically, depending on the current residue  $\mathbf{d}$ . Since the argument



**Figure 4.** The estimation error as a function of the FIR filter order (threshold set as zero). The black dots point out the FIR Filter order values for which the minimum RMS errors were achieved.

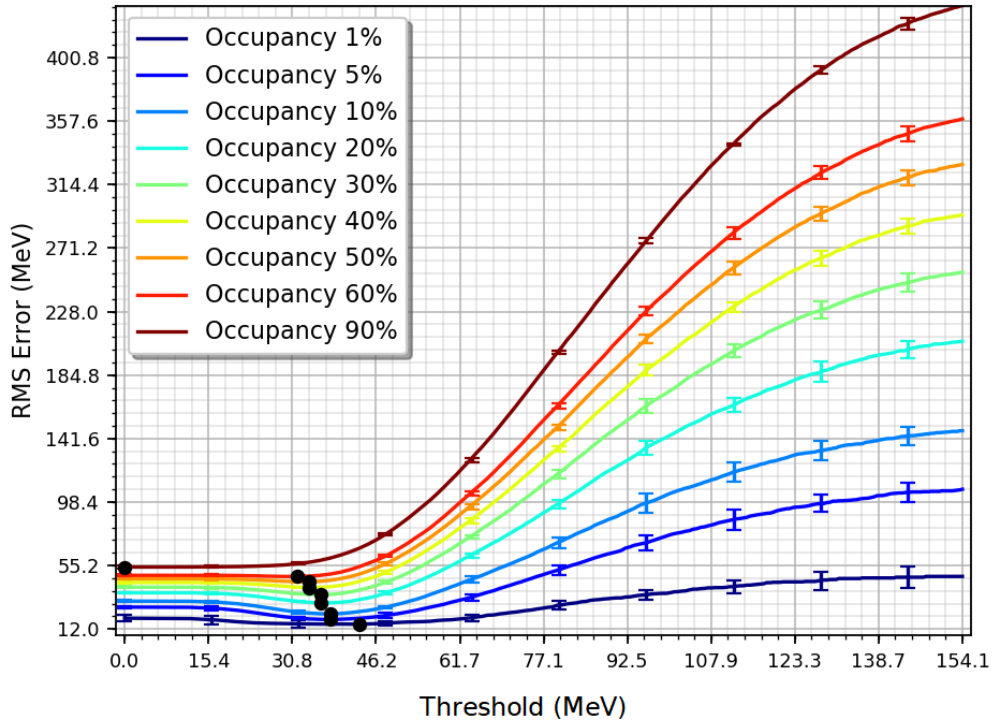
in equation (3.11) is a linear Gradient-Descent iteration step, the optimal step-size  $\mu$  for the  $i$ -th iteration is [57]

$$\mu = \frac{\mathbf{d}^{iT} \mathbf{d}^i}{\mathbf{d}^{iT} (\mathbf{H}^T \mathbf{H}) \mathbf{d}^i} \quad (4.2)$$

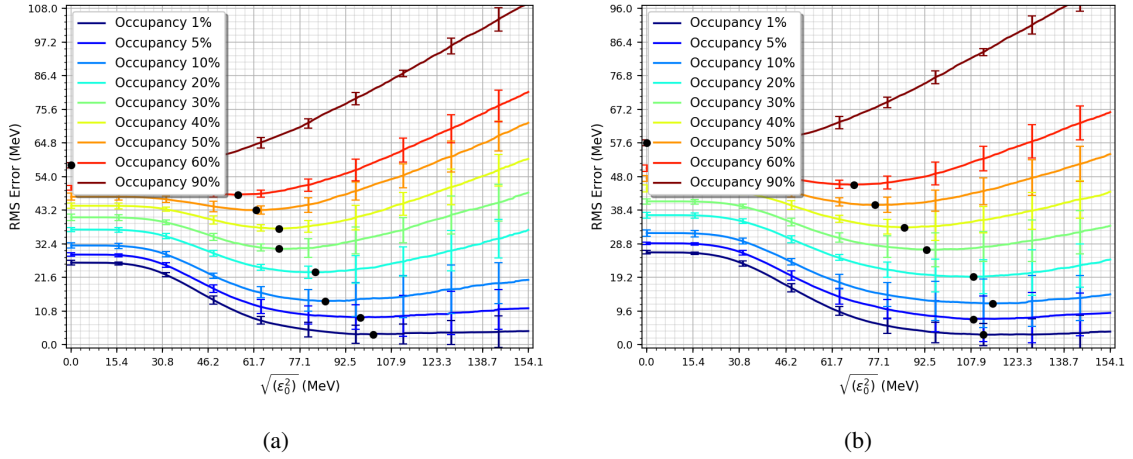
where  $\mathbf{d}^i = (\mathbf{H}\mathbf{x}^i - \mathbf{r})$  is the residue vector at the  $i$ -th iteration.

Figure 7 shows the evolution of the RMS error as a function of the iteration index, for four different  $\mu$  values, when the occupancy level was kept fixed at 30%. When evaluating the dependency with respect to the parameter  $\mu$ , the other parameter ( $\lambda$ ) was set to zero. For these plots, the entire set (100,100 samples) was used to increase the statistics and no error bar is shown.

The four evaluated values for  $\mu$  were: the dynamic one (according to equation (4.2)) and three fixed values, in order to avoid the computation of equation (4.2) in hardware. The three fixed values were chosen within the range of the values obtained through the dynamic approach, so that no multiplication circuitry would be necessary in a fixed-point numerical representation for any of these values in hardware (bit-shifting operation only) [58]. One may notice that  $\mu = 0.5$  was too high and the algorithm turned out to be unstable. On the other extreme, for  $\mu = 0.125$  the convergence was too slow. The best solution was found for  $\mu = 0.25$ , which produced an evaluation performance similar to the dynamical one (see figure 7). Figure 8 summarises the performance for the Separable Surrogate Functional method as a function of the number of iterations for different occupancy levels, when  $\mu = 0.25$  was kept fixed.

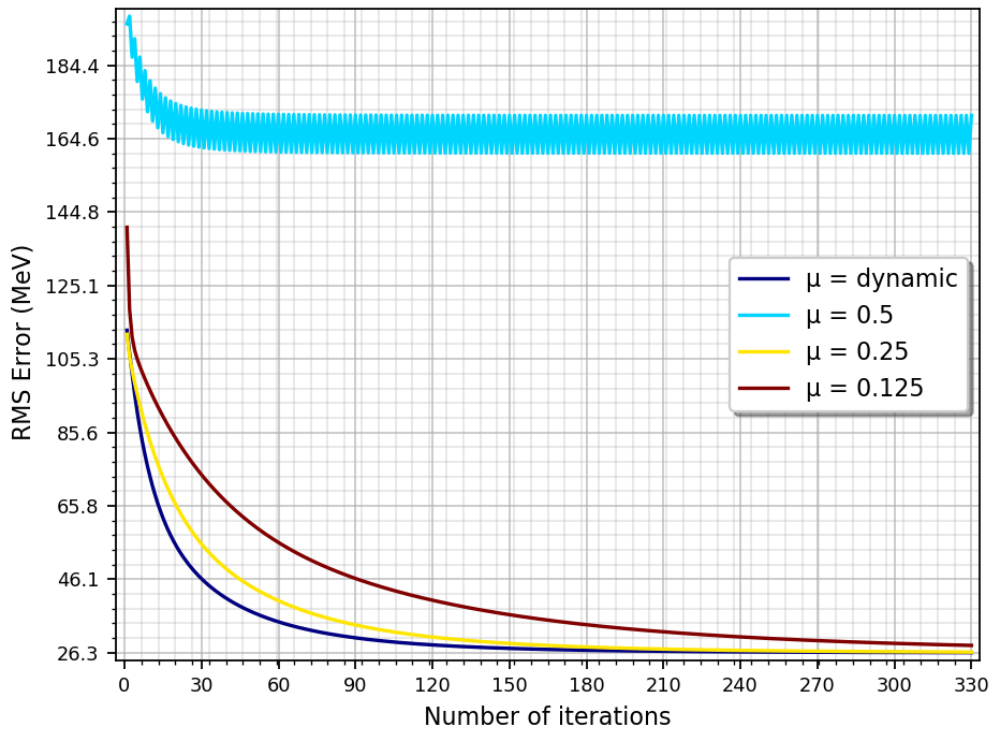


**Figure 5.** The estimation error as a function of the threshold value determined in the output of the FIR filter. The black dots point out the threshold values for which the minimum RMS errors were achieved.

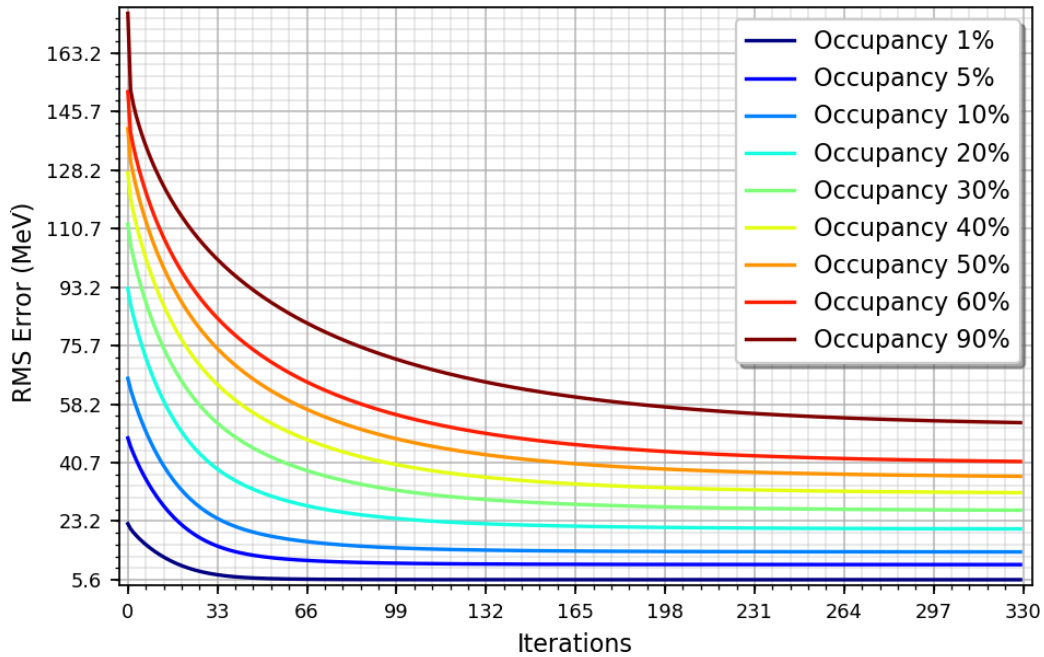


**Figure 6.** The RMS error as a function of the residue energy for both OMP (a) and LS-OMP (b) methods. The black dots point out the threshold values for which the minimum RMS errors were achieved.

Concerning the tuning for the parameter  $\lambda$ , figure 9 shows how the estimation performance behaved around the optimal value for several occupancy levels. It can be seen that the value of the parameter  $\lambda$  tended to zero while the pile-up level increased. Therefore, for sake of hardware simplicity, the parameter  $\lambda$  was set to zero, regardless of the occupancy level.

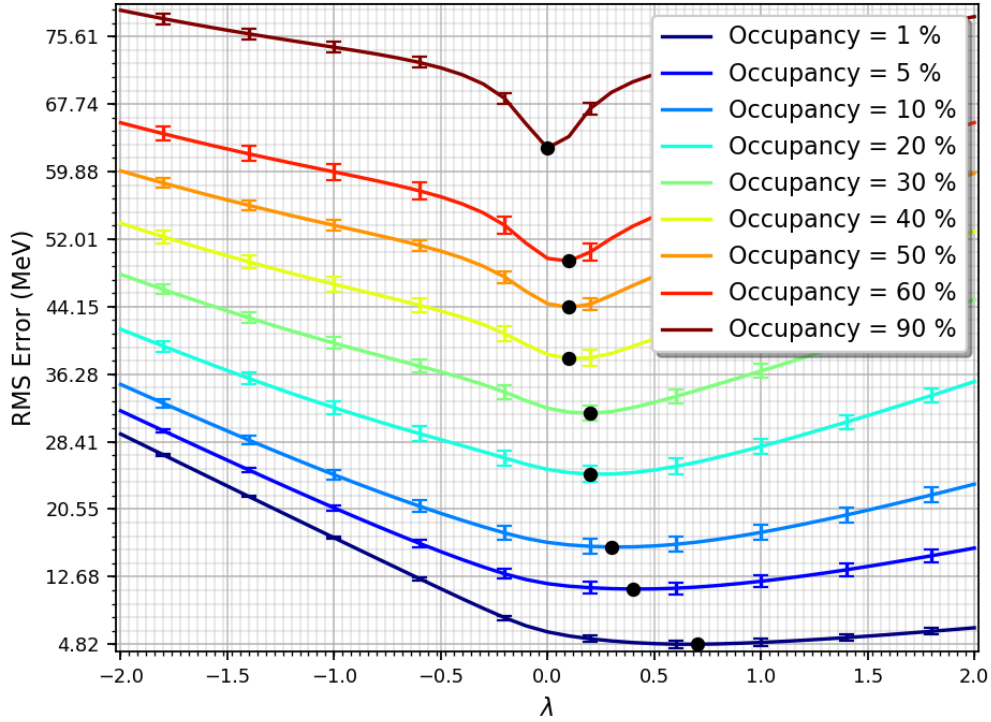


**Figure 7.** The RMS error as a function of the number of iterations. A cell with 30% of occupancy level was employed.



**Figure 8.** The RMS error as a function of the number of Separable Surrogate Functional iterations for several occupancy levels and  $\mu = 0.25$ .





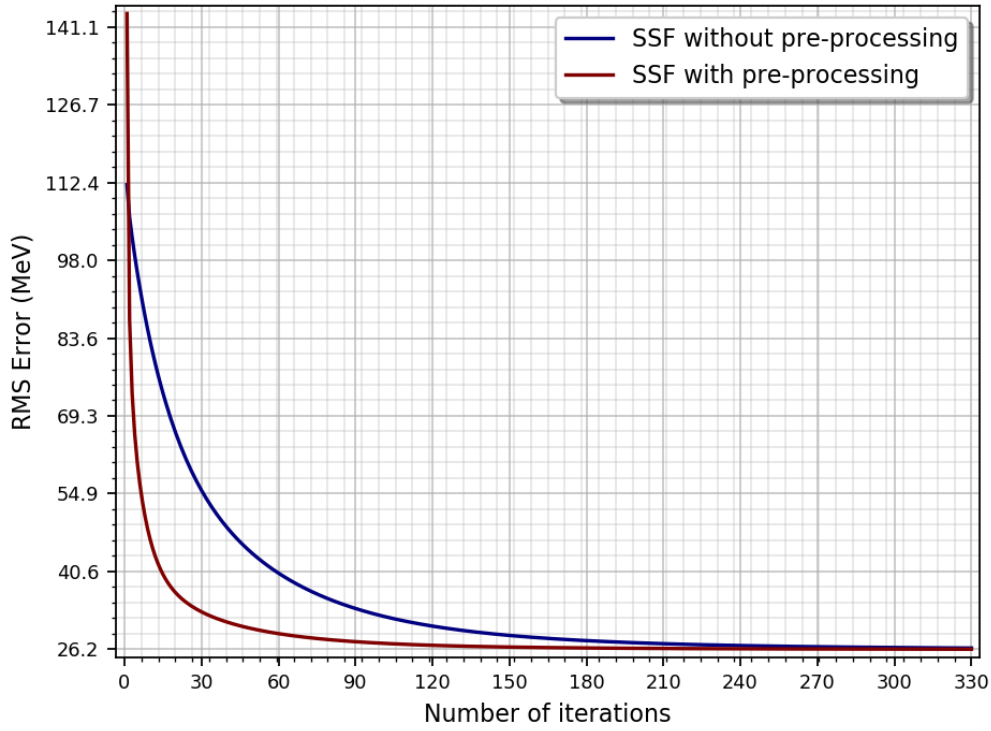
**Figure 9.** The RMS error as a function of the parameter  $\lambda$  in the Separable Surrogate Functional method, considering several occupancy levels and  $\mu = 0.25$ . A total of 330 iterations were applied in order to guarantee full convergence of the algorithm. The black dots refer to the optimal values.

The results in figure 8 were obtained by initializing the vector  $\mathbf{x}^0$  with the 48 central elements of the measured vector  $\mathbf{r}$ . This was shown to be faster than initializing  $\mathbf{x}^0$  as a null vector. However, as can be seen from this figure, more than 200 iterations were still necessary for algorithm convergence for most of the occupancy levels. Thus, in this work, the initialization of  $\mathbf{x}$  is proposed to follow equation (3.12) and figure 10 compares the evaluation of the estimation error as the number of iterations increases, with and without this preprocessing scheme, for a cell with 30% of occupancy level. One can notice that the convergence proved to be faster when the proposed initialization was applied.

### 4.3 Performance

Once the parameters for the aforementioned methods were all set to their design tuned values, a performance comparison was carried out. Table 1 summarizes the final performance. In this table, Linear Programming (as described in [18]) corresponds to a direct implementation of the problem  $P_{1,\epsilon_0}$  (equation (3.9)), and FIR corresponds to the method proposed in [23]. The last three columns refer to the methods proposed in this paper.

For lower occupancy levels, the target sequence ( $x[n]$ ) presented high sparsity and the Sparse Representation methods outperformed the FIR filter considerably. As the occupancy level was increased, the sparsity assumption gradually lost its strength. Therefore, the performance of the Sparse Representation methods got closer to the one from FIR filters for occupancy values above 30%. Among the Matching Pursuit methods, LS-OMP was slightly better. However, the



**Figure 10.** The RMS error as a function of the number of Separable Surrogate Functional iterations with and without the preprocessing step for  $\mathbf{x}^0$  determination. A cell with 30% of occupancy level was employed.

**Table 1.** Performance estimation in terms of the RMS error values (in MeV). Best performance for each occupancy level is shown in boldface.

Occupancy level	Linear Programming	FIR Filter	OMP	LS-OMP	SSF*
1 %	4.44±1.07	15.22±2.33	3.27±0.75	<b>2.86±0.49</b>	3.72±0.33
5 %	10.13±0.67	18.48±0.91	8.77±0.56	<b>7.29±0.35</b>	9.21±0.62
10 %	14.9±0.93	22.3±0.74	13.95±0.9	<b>11.8±0.91</b>	13.28±0.84
20 %	22.12±0.83	29.76±0.68	23.27±0.64	<b>19.36±0.79</b>	20.51±0.92
30 %	28.47±0.9	35.33±0.69	30.79±1.03	27.12±0.73	<b>26.26±0.82</b>
40 %	34.32±0.92	40.27±0.65	37.26±0.86	33.45±0.93	<b>31.67±0.9</b>
50 %	39.03±0.91	44.12±0.7	43.25±1.09	39.91±0.83	<b>36.57±0.68</b>
60 %	45.6±0.97	47.72±0.8	48.17±0.77	45.67±0.97	<b>41.01±0.98</b>
90 %	59.21±0.87	54.15±0.76	57.68±0.88	57.68±0.88	<b>52.63±0.65</b>

\*Separable Surrogate Functional (SSF) parameters set as  $\mu = 0.25$  and  $\lambda = 0$ .

Matching Pursuit methods were less efficient with respect to Separable Surrogate Functional for higher occupancy levels. The first step of the Matching Pursuit algorithms, the vector support determination, presented lower efficiency when the sparsity of the target sequence was not so high. Concerning the offline method used for comparison (Linear Programming), the achieved results were similar to the ones from LS-OMP.

In general, the occupancy levels shown in table 1 may be considered practical for different designs in modern collider's experiments like ATLAS and CMS, in LHC, and for technologies which are being considered for future colliders. Apart from the nominal collider luminosity, the occupancy level is also strongly correlated to the detector geometry parameters, such as pseudo-rapidity and radial layer. For forward calorimeters (with very high pseudo-rapidity), occupancy values at 90% level are already found in the LHC detectors, while in low pseudo-rapidity regions, the occupancy near 1% is the one mostly found. However, as the luminosity of the colliders is foreseen to increase continuously, the necessity of using methods more resilient to pile-up effects, such the one proposed here, becomes more evident. In the Run 3, for example, the luminosity is expected to increase considerably.

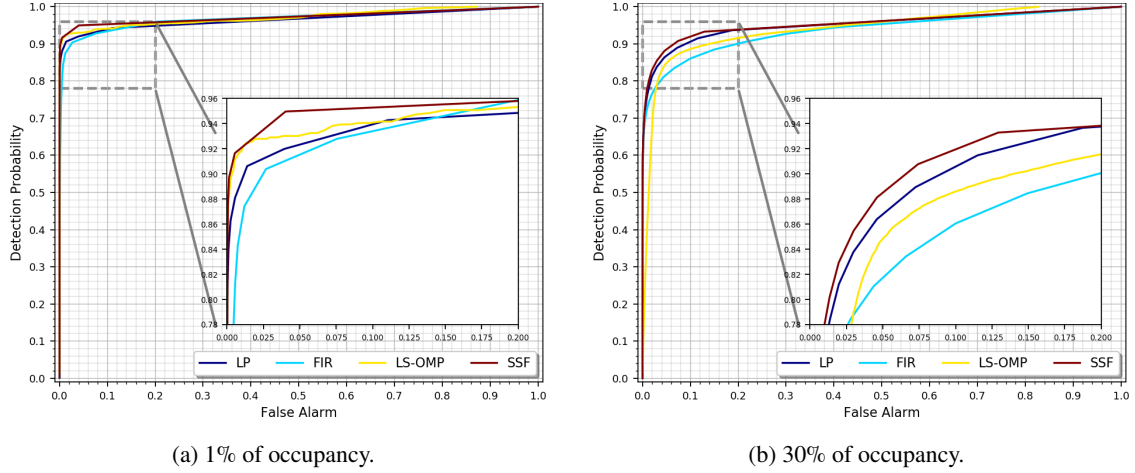
#### 4.3.1 Detection efficiency

In this section, performance evaluation in terms of detection efficiency is outlined. The analysis was performed through the Receiver Operating Characteristic (ROC) curve [59], which shows how the Probability of Detection (PD) for a given signal varies when a detection threshold is applied in the energy estimation value and the corresponding impact in the False Alarm (FA) probability. Here, PD comprised the percentage of bunches crossings with hits that were correctly detected (estimated energy above the threshold) and FA was the percentage of bunches crossings with no hits that have been wrongly selected. Figure 11 presents the ROC curves for two different occupancy levels (1% and 30%). The two tested Matching Pursuit methods (OMP and LS-OMP) presented similar results. Thus, only LS-OMP is shown in this analysis. For 1% of occupancy, both FIR filter and Separable Surrogate Functional methods presented similar detection efficiency. As the occupancy was increased, the advantage of the Sparse Representation methods, when compared to the FIR filter, became more evident. The Separable Surrogate Functional method remained as one with the highest detection efficiency, regardless of the occupancy level (1% or 30%).

#### 4.3.2 Analysis summary

From the above analysis, one can conclude that, for lower occupancy conditions, if only the detection efficiency is taken into account (triggering purpose only), the FIR filter structure is a competitive choice, as it can be seen from the ROC curves. On the other hand, if the estimation accuracy is also an important issue, Matching Pursuit methods may be employed. For high pile-up levels, the Separable Surrogate Functional method may be chosen when both estimation accuracy and detection efficiency are of main concern. Besides, even for lower occupancy values, the Separable Surrogate Functional method was competitive with respect to general energy estimation performance.

The Separable Surrogate Functional method has also the lowest computational cost among the Sparse Representation one. Therefore, the next section evaluates an online implementation of the Separable Surrogate Functional method in FPGA technology and does a comparison with the FIR filter implementation in terms of technology resource requirements.



**Figure 11.** The ROC curves for low (a) and high (b) occupancy values.

## 5 Hardware implementation and cost analysis

Aiming at optimizing the hardware resource utilization, the argument in equation (3.11) is rewritten according to:

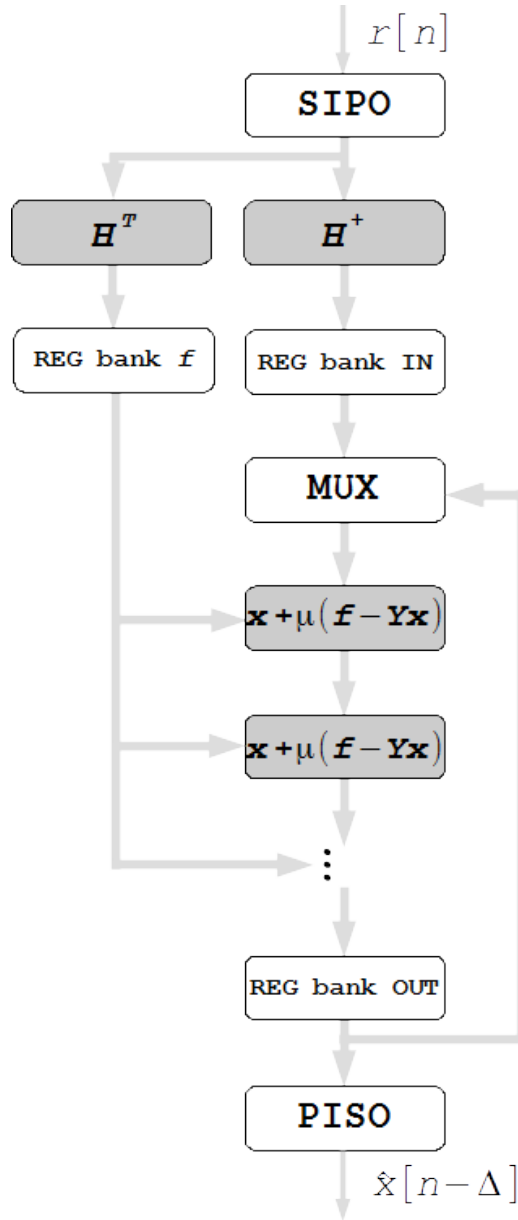
$$\mathbf{x}^i + \mu(\mathbf{H}^T \mathbf{r} - \mathbf{H}^T \mathbf{H} \mathbf{x}^i) \quad (5.1)$$

where  $\mathbf{H}^T \mathbf{H}$  is a constant Toeplitz matrix [60], which is computed offline. Concerning logic circuitry allocation, the Toeplitz matrix presents a certain symmetry that allows to reduce the cost through parallel implementation. In the text and in the figures that follow, this matrix is identified as  $\mathbf{Y}$  for sake of simplicity. For the same reason, the vector  $\mathbf{H}^T \mathbf{r}$  is also previously computed in hardware, just before the first iteration, and it will be indicated as a constant vector  $\mathbf{f}$ .

Other important simplification concerns the non-linearity imposed by the shrinkage function (figure 3). For  $\lambda = 0$ , this function is implemented by a simple logic comparison: if the value is lower than zero, the component of the vector  $\mathbf{x}$  is replaced by zero before the next iteration.

Figure 12 shows the block diagram for Separable Surrogate Functional algorithm implementation. The blocks in gray color execute matrix computations altogether, in parallel (distributed combinational circuitry) and in fixed-point arithmetic [61]. In this figure, it is possible to identify a system with three pipeline stages [62], split into input and output register banks, where:

1. The first pipeline stage comprises the SIPO (Serial-Input Parallel-Output) circuitry, the block which computes the  $\mathbf{f} = \mathbf{H}^T \mathbf{r}$  operation and the preprocessing for  $\mathbf{x}^0$  computation. The SIPO assembles a window of 55 samples (the vector  $\mathbf{r}$ ) buffering the most recent consecutive  $r[n]$  samples. When the SIPO internal registers are fully occupied, both vectors  $\mathbf{f}$  and  $\mathbf{x}^0$  are computed. In the next clock rising edge, the 48 resulting components, for both vectors, are stored in input register banks. This process is repeated to each window (55 clock cycles) uninterruptedly.
2. The second pipeline stage comprises the blocks between the input and the output register banks. The ones in gray color are responsible to execute one Separable Surrogate Functional iteration. This stage has a feedback path, synchronous to the data clock (40 MHz, if LHC



**Figure 12.** The block diagram of the proposed circuitry. It operates synchronously with the  $r[n]$  input signal.

would be the target application), in order to compute several iterations before the next window is delivered by the first pipeline stage. The resulting reconstructed vector ( $\mathbf{x}'$ ) is updated in the output register bank to each clock.

3. The third and last stage is executed by the PISO (Parallel-Input Serial-Output) block. This circuitry serialises the reconstructed vector present in the output register bank, delivering the free-running  $\hat{x}[n - \Delta]$  information, synchronous with the ADC clock (40 MHz). In this way, the windowed processing of the proposed system becomes transparent.

## 5.1 Firmware implementation

As described before, the system operates in a three stage pipeline architecture. Thus, three consecutive windows are evaluated simultaneously, one at each stage. The first stage buffers the most recent window and sends the data, in parallel, to the next stage.

The Separable Surrogate Functional iterations are implemented by the second pipeline stage. In the first clock, the multiplexer is switched to the input register. In this way, the initial value of the algorithm is  $\mathbf{x}^0$ , as expected. This signal propagates through the blocks that compute one Separable Surrogate Functional iteration. For the consecutive 55 clock cycles, the multiplexer points to the output register, which updates the reconstructed vector, in a feedback data-flow, at each clock cycle. In this stage, if only one Separable Surrogate Functional iteration block is implemented, it is possible to compute 55 iterations before the first stage sends a new fresh window. Therefore, a cascade of  $z$  blocks allows the execution of  $55 \times z$  iterations.

The last stage simply serializes the most recent reconstructed window delivered by the second stage. As a consequence of this proposed architecture, the response delay  $\Delta$  is kept fixed in 2 times the window size ( $2 \times 55 = 110$ ), regardless the number of iterations (which is a multiple of 55).

## 5.2 Cost analysis

The proposed circuitry was implemented in FPGA. The Software Vivado [63] was used in the design, and Verilog [64] was the Hardware Description Language (HDL) [29]. For performance tests, the chip XCZU7EV-FFVC1156-2-E from the Zynq UltraScale+ family [65] was chosen, since its evaluation is free and can be crosschecked without costs. For representation, 17 bits (7 bits for range and 10 bits for precision) were selected for fixed-point arithmetic, for all the parts of the circuitry. This configuration was chosen empirically after exhaustive simulation with several different numerical representations. Table 2 summarizes the implementation performance. As for comparison, the results for a FIR filter structure are also outlined, and the details about the hardware implementation of the FIR filter may be found in [23]. In this table, it is shown the performance results for up to three cascaded iteration blocks. For more than this, the operation frequency is below 40 MHz. Among the Separable Surrogate Functional implementations, the one with only one iteration block together with the preprocessing circuit is an attractive solution in terms of cost-efficiency.

As it can be seen from previous analysis, the Separable Surrogate Functional implementation requires  $15\times$  more resources, on average, than FIR filter structure. However, the proposed method presents a higher estimation performance, which may justify its implementation, depending on the available resources. It is worth to mention that the amount of resources for Separable Surrogate Functional implementation is proportional to the square of the window length. The value of 55 used here is just one example of direct implementation. Ongoing analyses are investigating whether splitting the processing window in several smaller windows may improve the implementation requirements. For practical considerations, table 3 presents the resources used to fit the Separable Surrogate Functional method in three different FPGA technologies, where the feasibility of implementation becomes evident.

**Table 2.** Implementation resources in terms of different hardware elements and frequency operation.

Metrics	FIR	SSF 1 Block		SSF 2 Blocks		SSF 3 Blocks	
		<b>Pp</b>	<b>nPp</b>	<b>Pp</b>	<b>nPp</b>	<b>Pp</b>	<b>nPp</b>
LUT**	1,447	31,378	33,132	59,927	63,966	87,160	94,870
FF**	731	2,097	2,625	2,098	2,625	2,097	2,625
DSP**	125	435	308	503	376	571	444
Freq (MHz)	56.76	71.20	79.17	50.34	57.39	41.47	42.22
RMS (MeV)	40	31	43	29	32	28	29

\*Pp = with preprocessing nPp = without preprocessing.

\*\*Number of: LUT = Look-up-tables, FF = Flip-Flops, DSP = Digital Signal Processor.

**Table 3.** Resources spent in commercial FPGA devices.

Metrics	SSF 3 Blocks <sup>1</sup>	XCZU7EV <sup>2</sup>	XC7VX1140T <sup>3</sup>	XCVU440 <sup>4</sup>
	<b>nPp</b>	<b>low cost</b>	<b>middle cost</b>	<b>high cost</b>
LUT	94,870	41.17% (230,400)	13,32% (712,000)	3,74% (2,532,960)
FF	2,625	0.56% (460,800)	0,18% (1,424,000)	0,05% (5,065,920)
DSP	444	25.69% (1,728)	13,21% (3,360)	15,41% (2,880)

<sup>1</sup>SSF 3 Blocks without preprocessing.

<sup>2</sup>XCZU7EV-FFVC1156-2-E.

<sup>3</sup>XC7VX1140T-1FLG1930C-ND.

<sup>4</sup>XCVU440-2FLGA2892E.

## 6 Conclusion

This paper presented an iterative method for a matrix based deconvolution process, which can be implemented for online operation. The proposed method is based on established sparse theory, over-performing the previously proposed FIR filter based deconvolution method. The method based on Iterative Shrinkage has shown to be a good choice for implementation and a FPGA circuitry was proposed. Compared with a simpler FIR filter design, despite the increase in cost implementation, it is feasible when modern FPGA technology is considered. The gain in estimation accuracy is evident, with improvement between 3% and 300% (depending on the occupancy level), when compared with FIR filter implementation. Therefore, it may justify its usage in experiments with intense pile-up levels.

## Acknowledgments

The authors would like to thank CNPq, CAPES, RENAFAP-MCTI, FAPERJ, FAPEMIG (Brazil) and CERN (Switzerland) for their support to this work.



## References

- [1] W. Barletta et al., *Future hadron colliders: from physics perspectives to technology R&D*, *Nucl. Instrum. Meth. A* **764** (2014) 352.
- [2] O. Brüning and L. Rossi, eds., *The High Luminosity Large Hadron Collider: the new machine for illuminating the mysteries of universe*, *World Scientific*, Geneva, Switzerland (2015).
- [3] L. Evans and P. Bryant eds., *LHC machine*, *2008 JINST* **3** S08001.
- [4] K. Anthony, *Celebrating the first of a kind*, *CERN document server*, (2014).
- [5] C. Pralavorio, *The new annual report is available*, *CERN document server*, (2017).
- [6] M. Cepeda, S. Gori, P. Ilten, M. Kado and F. Riva, *Report from working group 2: Higgs physics at the HL-LHC and HE-LHC*, *CERN Yellow Rep. Monogr.* **7** (2018) 221.
- [7] ATLAS collaboration, *ATLAS plans for the High-Luminosity LHC*, Tech. Rep. *ATL-PHYS-PROC-2018-048*, CERN, Geneva, Switzerland (2018).
- [8] R. Wigmans, *Calorimetry*, 2<sup>nd</sup> edition, *Oxford University Press*, U.K. (2018).
- [9] M. Jeitler, *Trigger systems of LHC experiments*, *2017 JINST* **12** C05012.
- [10] ATLAS collaboration, *The ATLAS experiment at the CERN Large Hadron Collider*, *2008 JINST* **3** S08003.
- [11] CMS collaboration, *The CMS experiment at the CERN LHC*, *2008 JINST* **3** S08004.
- [12] E. Fullana et al., *Digital signal reconstruction in the ATLAS hadronic tile calorimeter*, *IEEE Trans. Nucl. Sci.* **53** (2006) 2139.
- [13] V. Radeka and S. Rescia, *Speed and noise limits in ionization chamber calorimeters*, *Nucl. Instrum. Meth. A* **265** (1988) 228.
- [14] G. Bertuccio, E. Gatti, M. Sampietro, P. Rehak and S. Rescia, *Sampling and optimum data processing of detector signals*, *Nucl. Instrum. Meth. A* **322** (1992) 271.
- [15] W.E. Cleland and E.G. Stern, *Signal processing considerations for liquid ionization calorimeters in a high rate environment*, *Nucl. Instrum. Meth. A* **338** (1994) 467.
- [16] S. Haykin, *Adaptive filter theory*, 5<sup>th</sup> edition, Prentice Hall, Hoboken, NJ, U.S.A. (2013).
- [17] L.M. de A. Filho, B.S. Peralva, J.M. de Seixas and A.S. Cerqueira, *Calorimeter response deconvolution for energy estimation in high-luminosity conditions*, *IEEE Trans. Nucl. Sci.* **62** (2015) 3265.
- [18] D.P. Barbosa, L.M. de A. Filho, B.S. Peralva, A.S. Cerqueira and J.M. de Seixas, *Sparse representation for signal reconstruction in calorimeters operating in high luminosity*, *IEEE Trans. Nucl. Sci.* **64** (2017) 1942.
- [19] M. Elad, *Sparse and redundant representations*, *Springer*, New York, NY, U.S.A. (2010).
- [20] L.R.M. Silva, L.M. de Andrade Filho and C.A. Duque, *Sparse representation algorithm applied to power systems signal compression*, *Int. Trans. Elect. Energy Syst.* **29** (2018) e2693.
- [21] J.-L. Starck, F. Murtagh and J.M. Fadili, *Sparse image and signal processing: wavelets, curvelets, morphological diversity*, Cambridge University Press, New York, NY, U.S.A. (2010).
- [22] S.K. Mitra, *Digital signal processing: a computer-based approach*, 4<sup>th</sup> edition, McGraw-Hill Higher Education, New York, NY, U.S.A. (2010).
- [23] J.P.B.S. Duarte, L.M.d.A. Filho, E.F.d.S. Filho, P.C.M.A. Farias and J.M. de Seixas, *Online energy reconstruction for calorimeters under high pile-up conditions using deconvolutional techniques*, *2019 JINST* **14** P12017.

- [24] S.H. Lim, J. Won Choi and B. Shim, *Greedy sparse channel estimation for millimeter wave communications*, in *TENCON 2018 — 2018 IEEE region 10 conference*, [IEEE](#), October 2018, pg. 1628.
- [25] A. Zaki, A. Venkitaraman, S. Chatterjee and L.K. Rasmussen, *Greedy sparse learning over network*, [IEEE Trans. Signal Informat. Processing Networks](#) **4** (2018) 424.
- [26] M. Niu, W. Li, Z. Liu, Y. Zhang and J. Yang, *An accelerated iterative shrinkage-thresholding algorithm for real-beam scanning radar super-resolution*, in *2019 IEEE radar conference (RadarConf)*, [IEEE](#), April 2019, pg. 1.
- [27] Y. Ma and T. Zheng, *Stabilized sparse online learning for sparse data*, [J. Machine Learn. Res.](#) **18** (2017) 1.
- [28] P. Zhao, D. Wang, P. Wu and S.C.H. Hoi, *A unified framework for sparse online learning*, [ACM Trans. Knowledge Discovery Data](#) **14** (2020) 1.
- [29] N. Botros, *HDL with digital design*, Mercury Learning & Information, (2015).
- [30] U. Meyer-Baese, *Digital signal processing with field programmable gate arrays*, [Springer](#), Berlin, Heidelberg, Germany (2007).
- [31] G. Arce, *Nonlinear signal processing: a statistical approach*, Wiley-Interscience, Hoboken, NJ, U.S.A. (2005).
- [32] D.G. Luenberger and Y. Ye, *Linear and nonlinear programming*, [Springer](#), New York, NY, U.S.A. (2008).
- [33] S. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, [IEEE Trans. Signal Processing](#) **41** (1993) 3397.
- [34] D.R. Hadley, *Digital filtering performance in the ATLAS level-1 calorimeter trigger*, in *2010 17<sup>th</sup> IEEE-NPSS real time conference*, [IEEE](#), May 2010.
- [35] S.M. Kay, *Fundamentals of statistical signal processing*, volume I, Prentice Hall, NJ, U.S.A. (1993).
- [36] S. Chen and M. Yang, *An adaptive fast iterative shrinkage threshold algorithm*, in *2017 29<sup>th</sup> Chinese Control and Decision Conference (CCDC)*, [IEEE](#), May 2017.
- [37] M. Takac, *Randomized coordinate descent methods for big data optimization*, University of Edinburgh, Edinburgh, U.K. (2014).
- [38] E. Candes and T. Tao, *The Dantzig selector: statistical estimation when  $p$  is much larger than  $n$* , [Ann. Statist.](#) **35** (2007) 2313.
- [39] H. Wu and S. Wang, *Adaptive sparsity matching pursuit algorithm for sparse reconstruction*, [IEEE Signal Processing Lett.](#) **19** (2012) 471.
- [40] T.T. Cai and L. Wang, *Orthogonal matching pursuit for sparse signal recovery with noise*, [IEEE Trans. Inform. Theory](#) **57** (2011) 4680.
- [41] B. Dumitrescu, *Sparse total least squares: analysis and greedy algorithms*, [Linear Alg. Appl.](#) **438** (2013) 2661.
- [42] C.T. Kelley, *Solving nonlinear equations with Newton's method*, [Society for Industrial and Applied Mathematics](#), Philadelphia, PA, U.S.A. (2003).
- [43] I. Daubechies, M. Defrise and C.D. Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, [Commun. Pure Appl. Math.](#) **57** (2004) 1413.
- [44] S. Yun, *A coordinate gradient descent method for structured nonsmooth optimization: theory and applications*, VDM Verlag Dr. Müller, (2010).

- [45] S. Roman, *Advanced linear algebra*, Springer, New York, NY, U.S.A. (2008).
- [46] I. Hrivnacova et al., *The virtual Monte Carlo*, [CERN document server](#), (2003).
- [47] J. Chapman, *ATLAS simulation computing performance and pile-up simulation in ATLAS*, [LPCC](#), (2011).
- [48] S. Banerjee, *CMS simulation software*, [J. Phys. Conf. Ser.](#) **396** (2012) 022003.
- [49] TEVATRON BEAM-BEAM TEAM collaboration, *Overview of beam-beam effects in the Tevatron*, in *ICFA mini-workshop on beam-beam effects in hadron colliders*, [FERMILAB-CONF-13-107-APC](#), (2014), pg. 11 [[arXiv:1410.3326](#)].
- [50] W. Herr, *Dynamic behaviour of nominal and PACMAN bunches for different LHC crossing schemes*, Tech. Rep. [LHC-Project-Report-856](#), CERN, Geneva, Switzerland (2005) [[CERN-LHC-Project-Report-856](#)].
- [51] J. Fox et al., *Control of intra-bunch vertical motion in the SPS with GHz bandwidth feedback*, [J. Phys. Conf. Ser.](#) **1067** (2018) 072024.
- [52] M. Hostettler, K. Fuchsberger, G. Papotti, Y. Papaphilippou and T. Pieloni, *Luminosity scans for beam diagnostics*, [Phys. Rev. Accel. Beams](#) **21** (2018) 102801 [[arXiv:1804.10099](#)].
- [53] B.S.M. Peralva, L.M.A. Filho, A.S. Cerqueira and J.M. Seixas, *The TileCal energy reconstruction for collision data using the matched filter*, [IEEE Nucl. Sci. Symp. Med. Imag. Conf. Rec.](#) (2013) 1.
- [54] Y. Sugiyama et al., *The data acquisition system for the KOTO experiment*, [IEEE Trans. Nucl. Sci.](#) **62** (2015) 1115.
- [55] J. Keil, *Operational experience with HERA*, in *2007 IEEE Particle Accelerator Conference (PAC)*, [IEEE](#), (2007).
- [56] R. Duda, P. Hart and D.G. Stork, *Pattern classification*, volume XX, John Wiley and Sons Inc., U.S.A. (2001).
- [57] D.P. Bertsekas, *Nonlinear programming*, Athena Scientific, (1999).
- [58] W.E. Milne, *Numerical calculus*, Princeton University Press, Princeton, NJ, U.S.A. (2016).
- [59] T. Fawcett, *An introduction to ROC analysis*, [Pattern Recog. Lett.](#) **27** (2006) 861.
- [60] A. Böttcher and S.M. Grudsky, *Toeplitz matrices, asymptotic linear algebra and functional analysis*, [Hindustan Book Agency](#), (2000).
- [61] W.T. Padgett and D.V. Anderson, *Fixed-point signal processing*, [Synthesis Lect. Signal Proc.](#) **4** (2009) 1.
- [62] Y. Oda and T. Oyama, *Fast calculation using parallel processing and pipeline processing in power system analysis*, [Elect. Eng. Jpn.](#) **116** (1996) 85.
- [63] S. Churiwala ed., *Designing with Xilinx® FPGAs: using Vivado*, [Springer](#), Cham, Switzerland (2017).
- [64] S. Monk, *Programming FPGAs: getting started with Verilog*, McGraw Hill Education, New York, NY, U.S.A. (2017).
- [65] Xilinx, *Zynq UltraScale+ device packaging and pinouts*, [https://www.xilinx.com/support/documentation/user\\_guides/ug1075-zynq-ultrascale-pkg-pinout.pdf](https://www.xilinx.com/support/documentation/user_guides/ug1075-zynq-ultrascale-pkg-pinout.pdf), (2020).