

# Implementação em FPGA de um Método Recursivo de Deconvolução Aplicado em Calorímetros Operando a Alta Taxa de Eventos

Tiago A. Teixeira, João Paulo B. S. Duarte, Luciano M. de Andrade Filho, José M. de Seixas

**Abstract**—The LHC particle accelerator has undergone an update process, producing the effect known as *pile-up* (overlapping signals) in the Experiment front-end electronics. As the algorithm currently used for the reconstruction of the energy generated in these collisions is not tolerant to this effect, this work proposes the hardware implementation of an iterative method based on a variant of Descending Gradient that recovers the information of the amplitude of the signals superimposed within an acquisition window. With this implementation dozens of channels can be implemented in parallel within a single FPGA, in addition to respecting the required operating latency to the ATLAS acquisition system.

**Keywords**—Hardware, FGPA, Implementation, Calorimeters

**Abstract**—O acelerador de partículas LHC vem passando por um processo de atualização, produzindo o efeito conhecido como *pile-up* (empilhamento de sinais) na eletrônica de leitura dos detectores. Como o algoritmo atualmente utilizado para a reconstrução da energia gerada nessas colisões não é tolerante a esse efeito, este trabalho propõe a implementação em hardware de um método iterativo baseado em uma variante do Gradiente Descendente que recupera a informação da amplitude dos sinais sobrepostos dentro de uma janela de aquisição. Tal implementação permite que dezenas de canais possam ser implementados em paralelo dentro de uma única FPGA, além de respeitar a latência de operação necessária ao sistema de aquisição do ATLAS.

**Keywords**—Hardware, FPGA, Implementação, Calorímetros

## I. INTRODUÇÃO

Para o estudo da Física de Altas Energias, faz-se necessária a utilização de um refinado sistema de instrumentação, principalmente quando a área de pesquisa envolve a colisão de partículas em altas velocidades, sendo, para tal, utilizados os colisores de partículas. Para analisar fenômenos físicos cada vez mais raros, há a tendência de aumentar suas taxas de colisão e também o número de partículas por colisão, aumentando assim a probabilidade de observação de tais eventos [1]. Seguindo tal objetivo,

T. A. Teixeira<sup>†</sup>, J. P. B. S. Duarte<sup>†</sup>, L. M. de Andrade Filho<sup>†</sup> e J. M. Seixas<sup>‡</sup>, Departamento de Engenharia Elétrica, Universidade Federal de Juiz de Fora, Juiz de Fora-MG<sup>†</sup>, Brasil, Laboratório de Processamento de Sinais COPPE/POLI, Universidade Federal do Rio de Janeiro, Rio de Janeiro-RJ<sup>‡</sup>, Brasil, Emails: tiago.teixeira@engenharia.ufjf.br, joao.duarte@engenharia.ufjf.br, luciano.andrade@engenharia.ufjf.br, seixas@lps.ufrj.br.

a Organização Europeia para a Pesquisa Nuclear - CERN (*Conseil Européen pour la Recherche Nucléaire*) vem promovendo estudos em física de altas energias desde a sua fundação em 1954 [2]. Para tal, utiliza-se de aceleradores de partículas, sendo o principal deles o LHC (*Large Hadron Collider*), que possui quatro detectores de partículas: o CMS, o LHCb, o ALICE e o ATLAS [3].

O experimento ATLAS (*A Toroidal LHC ApparatuS*) é um experimento de propósito geral utilizado na investigação das colisões próton-próton [4]. Como pode ser visualizado na Figura 1 ele é composto por subdetectores dispostos em camadas, sendo cada um responsável por medir propriedades específicas das partículas geradas pelas colisões.

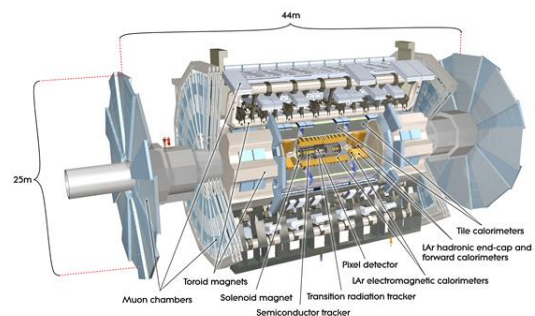


Fig. 1. Vista detalhada do detector ATLAS [4]

Dentre os subdetectores presentes no ATLAS podemos destacar: o Detector de Traços (*Inner Detector*) e seus Ímãs Solenoidais (*Solenoidal Magnets*), os calorímetros Hadrônico e Eletromagnético, além dos Ímãs Toroidais (*Toroid Magnets*) que funcionam em conjunto com o Espectrômetro de Múons (*Muon Spectrometer*) [5]. Os calorímetros são instrumentos construídos com o objetivo de medir a energia das partículas e fornecer boa contenção para as mesmas de modo a limitar o impacto no sistema de muons.

O detector ATLAS possui dois calorímetros. O mais interno é o eletromagnético, também conhecido como Calorímetro de Argônio Líquido - LAr (*Liquid Argon*), o qual mede a energia das partículas que interagem de forma eletromagnética. O segundo calorímetro do experimento ATLAS é o calorímetro hadrônico ou calorímetro de telhas TileCal (*Tile Calorimeter*) que funciona medindo a

energia de hádrons que atravessam o calorímetro eletromagnético, por ser menos denso.

As interações nos subdetectores do ATLAS [6] chegam a gerar um fluxo de dados de cerca de 60 Tb/s vindos da colisão de dois feixes de prótons com 2.808 grupos de  $10^{11}$  partículas cada um, viajando em direções opostas a uma velocidade de 99.9998 % da velocidade da luz, colidindo a uma taxa de 40 MHz [7]. Para processar estes dados, o ATLAS utiliza um sistema de *trigger* para selecionar quais eventos devem ser armazenados e quais podem ser ignorados. Esse sistema de *trigger* é de fato um sistema de filtragem online composto por dois níveis.

Sendo o primeiro nível da filtragem o foco deste trabalho, mais especificamente, no que diz respeito à informação de energia das colisões fornecida pelo TileCal. Neste primeiro nível de *trigger* a solução deve ser implementada em *hardware*, devido a alta taxa de eventos (40 MHz). É neste nível de *trigger*, que o sinal elétrico proveniente das células dos calorímetros (cerca de 10.000 células no caso do TileCal) é digitalizado a uma taxa de 40 MHz e sincronizado com as colisões no LHC. Como um pulso conformado tem uma duração temporal de 150 ns, enquanto a taxa de colisão do LHC tem uma duração temporal de 25 ns. Observa-se, assim, que essa diferença nos tempos de resposta do calorímetro, em relação a taxa de colisão do LHC, faz com que haja a possibilidade de ocorrer a sobreposição de amostras de eventos subsequentes, ou seja, causando empilhamento de sinais, *pile-up*. E também será nesse nível que haverá o maior impacto proveniente do aumento de probabilidade de colisões, previsto para ocorrer com as atualizações projetadas para os próximos anos.

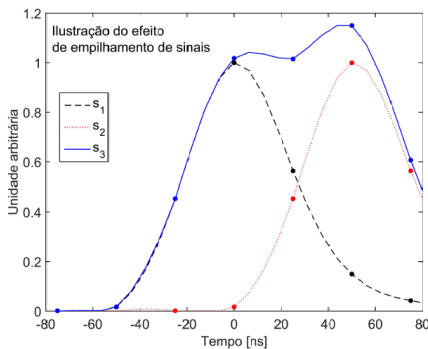


Fig. 2. Exemplo de empilhamento de sinais [8]

Um exemplo de *pile-up* pode ser visualizado na Figura 2, onde uma colisão gerou um sinal  $s_1$  sensibilizando uma determinada célula, e 50 ns depois, o equivalente a 2 colisões posteriores a uma taxa de 40 MHz, a mesma célula foi sensibilizada gerando um segundo sinal  $s_2$ . Por se tratarem de eventos temporalmente muito próximos, e inferiores ao tempo de 150 ns, necessário a identificação de um pulso conformado, forma-se algo parecido com o que seria um terceiro sinal  $s_3$ , que na verdade se trata

do empilhamento dos sinais  $s_1$  e  $s_2$ , fazendo com que o algoritmo atualmente em uso, baseado em um filtro casado [9], falhe na detecção do mesmo.

Para resolver este problema, na literatura, recentemente, encontram-se métodos baseados na deconvolução do sinal do calorímetro, de forma a obter novamente os sinais impulsivos [1] [8] [10] [11] representando diretamente o valor da energia em cada colisão. Tais métodos, em relação a implementação, podem ser classificados como os algoritmos de forma direta tais como Filtros de Resposta Finita ao Impulso (*Finite Impulse Response - FIR*), ou como métodos iterativos, onde um modelo linear de sobreposição de sinais é iterativamente alcançado, seguindo o critério de minimização do erro médio quadrático (*Root Mean Square - RMS*) entre os dados e o modelo.

Os métodos iterativos têm a vantagem de poder-se aplicar um limiar de corte no final de cada iteração, eliminando ruído e tornando a resposta mais esparsa, o que é mais condizente com o modelo dos dados. Isto faz com que o desempenho deste método seja superior ao das técnicas baseadas em filtros FIR, no que diz respeito a acurácia da informação de energia reconstruídas. Porém, estes métodos têm a desvantagem de possuir um custo computacional de implementação em FPGA muito alto, se comparados aos métodos baseados em filtros FIR, além de um severo problema de falha na reconstrução dos eventos que se encontram nas bordas da janela de aquisição (devido à presença de sinais incompletos pertencentes a janelas vizinhas).

Porém, desde que o LHC precisou lidar com o problema que ficou conhecido como 16L2 [12], uma nova estrutura de colisão passou a ser adotada. Essa nova estrutura é composta por trens de 8 colisões seguidas de um intervalo equivalente a 4 colisões vazias "8b4e"[13]. Com isto, no final de cada trem de colisões, haverá um período de 4 colisões sem geração de sinal. Do ponto de vista dos métodos iterativos, esta é a região ideal para início de uma janela, uma vez que não há o risco de haver um pedaço do sinal de janelas anteriores, resolvendo-se, assim, o problema de borda. Além disso, o tamanho da janela forçosamente tem que ser de 12 colisões. Resta, porém, o problema do custo computacional a ser resolvido, sendo esta a proposta deste trabalho: uma implementação em FPGA que viabiliza o uso de métodos de deconvolução baseados em janelas de 12 amostras, cujas bordas não apresentam partes incompletas de sinais de janelas vizinhas.

Na Seção II iremos apresentar as Técnicas de estimação de energia analisadas. Na Seção III apresentaremos o embasamento matemático da implementação. Na seção IV a proposta da implementação em si será detalhada. Na Seção V serão apresentados os resultados obtidos e, finalmente, na Seção VI é apresentada a conclusão deste trabalho.

## II. DECONVOLUÇÃO POR GRADIENTE DESCENDENTE

Podemos expressar a convolução como uma multiplicação de matrizes. Elucidando melhor esse contexto, consideremos uma janela amostral de tamanho 3  $[h_0 \ h_1 \ h_2]$  convoluindo-o com três amostras de um outro sinal  $[x_0 \ x_1 \ x_2]$ . O resultado dessa convolução gerará um terceiro sinal contendo agora 5 amostras  $[s_0 \ s_1 \ s_2 \ s_3 \ s_4]$ . Tal convolução pode ser expressa conforme o sistema:

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 \\ h_1 & h_0 & 0 \\ h_2 & h_1 & h_0 \\ 0 & h_2 & h_1 \\ 0 & 0 & h_2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \quad (1)$$

ou:

$$s = Hx \quad (2)$$

Interpretando o vetor  $s$  como sendo o resultado do sinal  $x$  convoluído com um sistema cuja matriz de convolução é conhecida e dada por  $H$ , o objetivo passa agora a ser a determinação do vetor  $x$  que melhor representa o modelo acima, para um dado vetor  $s$  conhecido a priori. O resultado obtido para  $x$  representa o sinal deconvoluido e será aqui interpretado como a informação da amplitude de cada sinal sobreposto. Construindo-se uma matriz  $H$  cujas colunas são dadas por deslocamento de pulsos característico da eletrônica de *front-end* do TileCal, o vetor  $x$  recuperado representa a energia depositada em cada colisão dentro de uma janela de aquisição.

Em [14], é proposto um método que parte da minimização do erro médio quadrático entre  $s$  e  $Hx$ , resultando na estimação através da pseudoinversa da matriz  $H$  [14]. O autor segue propondo em seu artigo uma metodologia para a realização de maneira iterativa, usando Gradiente Descendente (GD), que resulta na equação:

$$\hat{x}^{iter+1} = \hat{x}^{iter} + \mu(H^T s - A\hat{x}^{iter}) \quad (3)$$

onde  $A = H^T H$ . [11] cita a existência na literatura de um limite superior para  $\mu$ , tal que seu valor não pode ultrapassar o maior autovalor da matriz obtida por  $H^T H$  dividido por dois. Neste trabalho foi utilizado um valor médio obtido por simulação e respeitando o critério de limite superior acima.

### A. Gradiente Descendente Positivo

O método do Gradiente Descendente Positivo (GDP), propõe uma variação no GD de forma a colocar em zero os componentes da amostra que estiverem abaixo de um limiar positivo, antes da próxima iteração. Com isso, a cada iteração as amostras estimadas negativas ou próximas de zero são consideradas zero.

Tal modificação no GD faz com que o GDP tenda para uma solução esparsa do vetor  $\hat{x}^{iter+1}$  concentrando assim a energia em poucas componentes positivas, em vez de distribuir esta informação erroneamente em colisões

adjacentes, estimando valores positivos e negativos de energia. Sendo assim, essa solução tende a reconstruir as energias originais de cada colisão com menor erro.

## III. EMBASAMENTO MATEMÁTICO DA IMPLEMENTAÇÃO

A implementação em hardware do algoritmo GDP partiu da implementação do Algoritmo 1 em Matlab®.

### Algoritmo 1

IMPLEMENTAÇÃO DO GDP EM MATLAB®

---

```

1 mu = 0.25;
2 A = H' * H;
3 Hs = (H' * s');
4 x = s';
5 for l = 1:iterac;
6     B = Hs - A * x;
7     x = x + mu * B;
8     x(find(x < 0)) = 0;
9 end;
```

---

Algumas considerações pertinentes quanto à implementação deste algoritmo em FPGA são:

- 1) O vetor  $s$  representa uma janela do sinal de entrada.
- 2) Um circuito combinacional dedicado foi implementado para calcular a multiplicação da matriz  $H$  com o sinal de entrada  $s$ , formando o vetor  $Hs$ .
- 3) O vetor  $x$  representa registradores internos que armazenam o resultado após cada iteração.
- 4)  $\mu$  é uma constante determinada por simulação cujo valor utilizado é de 0.25.
- 5) Um circuito síncrono com o *clock* de 40 MHz do LHC é utilizado para implementar o processo iterativo do algoritmo. Os circuitos presentes nesta cadeia realimentada são descritos nos itens que se seguem.
- 6) Um circuito combinacional dedicado foi implementado para computar a matriz  $B$  igual  $Hs - Ax$ .
- 7) O vetor  $B$  passa por um circuito de deslocamento de 2 bits a direita, de forma a implementar a sua multiplicação por  $\mu$ .
- 8) Um novo vetor é calculado a partir da soma do mesmo com os resultados temporariamente armazenados em  $x$ .
- 9) Os valores menores que zero são substituídos por zero e o resultado é armazenado no próprio banco de registradores  $x$  na borda de subida do próximo *clock*.

Vale ressaltar que, para o ambiente externo, tal circuito deve operar de forma sequencial, conforme a operação de um filtro FIR, ou seja, para cada *clock* do LHC, um novo dado é inserido e retirado do sistema de forma ininterrupta. Isto requer circuitos adicionais de controle de fluxo, serialização e deserialização de dados. A próxima seção apresenta detalhes da implementação.

#### IV. IMPLEMENTAÇÃO

A Figura 3 apresenta um diagrama em blocos do circuito implementado. Os cálculos de uma iteração do algoritmo são computados de forma combinacional pelo bloco GDP. Portanto, uma cascata de  $n$  blocos GDP em sequência computa  $n$  iterações do algoritmo em um único ciclo de *clock*.

Os dados de entrada, vindos de um conversor analógico-digital de um canal de leitura do TileCal, apresentam-se de forma serial a uma taxa de 40 MHz. Inicialmente, estes dados são paralelizados em uma janela de 12 amostras pelo bloco conversor Serial-Paralelo (*Serial Input / Paralel Output* - SIPO). Os dados, agora em paralelo, alimentam o bloco *Hs* que computa o produto entre o vetor de entrada e a matriz *H* de forma combinacional. Na primeira iteração do algoritmo, estes mesmos dados de entrada devem alimentar o banco de blocos GDP. Isto equivale a inicializar o algoritmo (valor do vetor  $x$ ) com os dados de entrada. Este controle é feito por um multiplexador que, durante o primeiro *clock*, alimenta o banco de iterações com os dados de entrada. Para os outros *clocks*, o banco é realimentado através dos registradores de saída, permitindo a realização de  $n$  iterações por *clock*. Por fim, um circuito conversor Paralelo-Serial (*Paralel Input / Serial Output* - PISO) serializa os dados de saída de forma que os dados são apresentados também em sequência na saída do circuito.

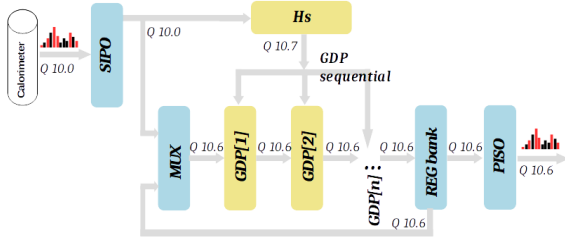


Fig. 3. Circuito completo implementado

##### A. Conversor SIPO

Este bloco possui dois bancos de 12 registradores, como mostrado na Figura 4. O primeiro banco é configurado como registradores de deslocamento, de modo a permitir a paralelização dos dados. Completado um ciclo de deslocamento de 12 amostras, as mesmas são copiadas para um novo banco de registradores que mantém os dados registrados por 12 ciclos de *clock*. Este registro é controlado por sinais de *enable*, através de um bloco contador, chamado de (*Enable Generator*).

##### B. Módulo GDP

Esse módulo pode ser visualizado em detalhes na Figura 5. Podemos identificar que as duas entradas do módulo correspondem a: 1) o vetor formado pelo sinal vindo do módulo *Hs*, 2) o vetor formado pela janela de sinais vindos diretamente do TileCal, ou pelo vetor de sinais previamente processados na iteração anterior. A seleção

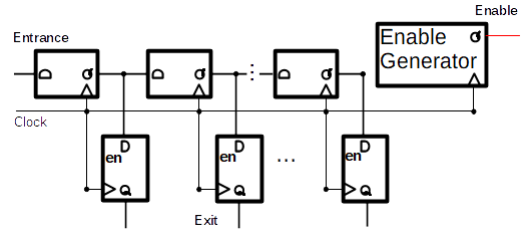


Fig. 4. Esquema do Conversor SIPO utilizado na entrada dos dados

de qual será a fonte do sinal é feita pelo Multiplexador sincronizado pelo *Enable Generator* do conversor SIPO.

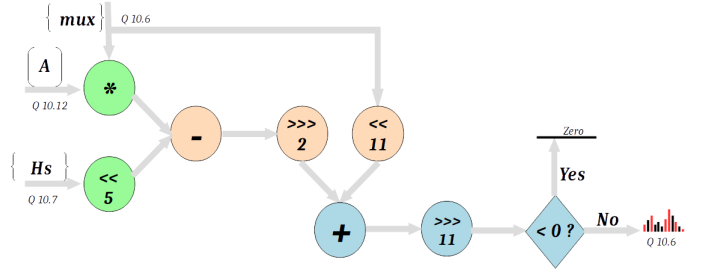


Fig. 5. Core do Circuito GDP completo implementado

##### C. Conversor PISO

Este bloco compreende uma sequência de registradores que, de forma equivalente ao Conversor SIPO, são configurados como registradores de deslocamento. A principal diferença é que, no caso do PISO, os registradores têm um controle adicional de *load*, permitindo que os dados de saída sejam carregados paralelamente e simultaneamente a cada 12 ciclos de *clock*. A Figura 6 mostra um diagrama em blocos deste circuito, onde a função de *load* é implementada através de multiplexadores.

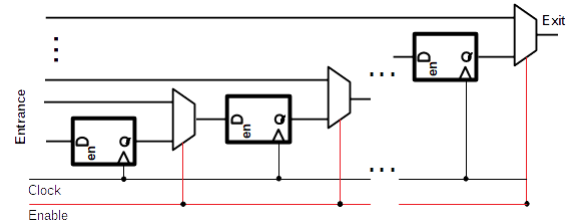


Fig. 6. Esquema do Circuito Paralelo-Serial utilizado na saída dos dados

#### V. RESULTADOS

Para a geração dos resultados apresentados nesta seção foram utilizadas as configurações de quantização de bits presentes na Figura 3. Esta configuração serviu de base para a quantização e deslocamentos utilizados na Figura 5. As constantes da Matriz *A*, agora no formato 10.12, sendo 10 bits da parte inteira e 12 bits representando a parte fracionária, são multiplicadas pelos valores vindos do

Multiplexador, no formato 10.6. Tal resultado é diminuído do valor vindo do módulo  $Hs$  deslocado à direita em 5 bits. O resultado é deslocado à esquerda em 2 bits, fazendo assim o equivalente a multiplicação pela constante  $\mu$ . O resultado obtido é somado ao valor vindo do Multiplexador deslocado à esquerda 11 bits, resultando em um valor que é deslocado à direita em 11 bits. Este último valor é avaliado, se for maior que zero, segue no circuito, se for menor, é colocado em zero antes de seguir.

O tempo disponível para a operação do algoritmo, em uma mesma janela, é de 12 ciclos de  $clock$  do LHC. Depois deste tempo, o bloco SIPO muda o vetor de entrada  $s$  para os dados da próxima janela e o bloco PISO passa a serializar os dados computados da janela anterior. Com isso, pode-se calcular a latência do algoritmo como sendo de 24 ciclos de  $clock$  do LHC, ou de 600  $ns$ , estando dentro da especificação de latência permitida pelo primeiro nível de  $trigger$  do ATLAS, que é de um microssegundo.

Como todo o circuito é sincronizado pelo mesmo  $clock$ , o número efetivo de iterações dependerá, portanto, do número  $n$  de blocos GDP implementados em cascata. Desta forma, o número total de iterações será de  $12n$ . A medida que aumentamos o número de blocos do GDP, aumentamos também o número de iterações equivalentes e, conseqüentemente, o número de elementos lógicos necessários para a construção do circuito final também aumenta. De forma inversa, o erro na reconstrução diminui a medida que o número de iterações equivalentes aumenta.

A Tabela I traz a relação entre tais componentes. A eficiência da reconstrução é dada pelo valor RMS do erro entre o valor real e o valor obtido da amplitude dos pulsos simulados. A título de comparação, um filtro FIR projetado para mesma função utiliza cerca de 3.000 elementos lógicos e como visto em trabalhos anteriores [11] [8], os valores em RMS mínimos alcançados se estabilizam em um patamar de 3,9 RMS, este valor também foi utilizado como referência no processo de quantização. Este patamar só é equivalente a utilização do método aqui proposto com apenas 2 blocos de GDP em cascata. Porém, a medida que aumentamos o número de blocos GDP, a eficiência do método aumenta consideravelmente. No entanto, o número de elementos lógicos utilizados também aumenta, para este trabalho foi utilizada uma placa da família Cyclone 10 LP a 10CL120, que comporta pouco menos de 120.000 elementos lógicos, possibilitando assim a implementação em paralelo de no mínimo dois canais. Em uma FPGA moderna, como a Stratix 10 GX 5500, a qual comporta mais de 5.500.000 de elementos lógicos, dezenas de canais operando em paralelo podem ser implementados.

A latência do circuito também ficou equivalente a implementação do filtro FIR, cujo valor é dado diretamente pela ordem do filtro, que é de 26.

## VI. CONCLUSÕES

Neste trabalho desenvolveu-se um circuito digital a ser implementado em FPGA capaz de operar métodos itera-

TABELA I  
RELAÇÃO ENTRE O RMS DAS IMPLEMENTAÇÕES DO ALGORITMO E O NÚMERO DE ELEMENTOS LÓGICOS GASTOS

Blocos	Iterações	Elementos Lógicos	RMS do erro	
			Verilog	Matlab
9	108	41.063	1,9767	1,9534
8	96	34.509	2,0508	2,0266
7	84	27.294	2,1354	2,1085
6	72	20.664	2,2764	2,2476
5	60	13.959	2,4543	2,4226
4	48	8.786	2,7000	2,6686
3	36	6.998	3,1213	3,0922
2	24	5.210	3,8227	3,7981

tivos de deconvolução com uma latência de duas vezes a janela de aquisição. A quantidade de recursos lógicos aumenta com o número de iterações requeridos. Para a implementação em janelas de 12 amostras, ideal para operação em calorímetros do LHC, é possível implementar dezenas de canais em paralelo e ainda operar com mais de 100 iterações. Embora a quantidade de recursos seja superior à de um filtro FIR de deconvolução, projetado com o mesmo propósito, a eficiência na reconstrução das amplitudes dos sinais empilhados é bem superior.

## REFERÊNCIAS

- [1] 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 Transactions on Nuclear Science*, vol. 62, pp. 3265–3273, dec 2015.
- [2] K. Anthony, "Celebrating the first of a kind," sep 2014.
- [3] CERN, "The large hadron collider," 2014.
- [4] J. Pequeno, "Computer generated image of the whole atlas detector." CERN-GE-0803012, mar 2008.
- [5] M. H. M. Faria, L. M. A. Filho, E. S. Filho, and J. M. Duarte, J. P. B. S. and Seixas, "Detecção de sinal em calorimetria de altas energias utilizando redes neurais artificiais," in *XXI Congresso Brasileiro de Automática*, (Vitória - ES), 2016.
- [6] C. Pralavorio, "The new annual report is available," jun 2017.
- [7] S. Klous, "Event streaming in the online system," Tech. Rep. ATL-DAQ-PROC-2010-017, CERN, Geneva, jul 2010.
- [8] J. P. B. S. Duarte, L. M. Andrade Filho, and J. M. Seixas, "Representação esparsa aplicada à estimação online de energia para um calorímetro submetido a uma alta taxa de eventos," in *XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, (São Pedro - SP), SBRT, sep 2017.
- [9] B. S. Peralva, "The tilecal energy reconstruction for collision data using the matched filter," in *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2013 IEEE, pp. 1–6, IEEE, 2013.
- [10] J. a. P. B. d. S. Duarte, "Estudo de técnicas de deconvolução para reconstrução de energia online no calorímetro hadrônico do atlas," Master's thesis, UFJF, 2015.
- [11] J. P. B. S. Duarte, L. M. Andrade Filho, E. F. Simas Filho, and J. M. Seixas, "Técnicas de deconvolução aplicadas à estimação de energia online em calorimetria de altas energias em condições de alta taxa de eventos," in *XXI Congresso Brasileiro de Automática*, (Vitória - ES), 2016.
- [12] K. Kahle, R. Steerenberg, and S. Pandolfi, "Lhc report: Something in the nothing," *CERN*, 2017.
- [13] J. Wenninger and S. Pandolfi, "Lhc report: operation with holes," *CERN*, 2017.
- [14] M. V. Teixeira, L. M. A. Filho, and B. S. Peralva, "Reconstrução online para calorímetros operando em condições de altas luminosidades," in *XX Congresso Brasileiro de Automática*, (Belo Horizonte - MG), 2014.